Remote Instrumentation Services on the e-Infrastructure

Franco Davoli · Norbert Meyer · Roberto Pugliese · Sandro Zappatore
Editors

Remote Instrumentation Services on the e-Infrastructure

Applications and Tools

Springer

Editors
Franco Davoli
Department of Communications, Computer and Systems Science (DIST)
University of Genoa
Via Opera Pia 13
16145 Genova, Italy
franco@dist.unige.it

Norbert Meyer
Poznań Supercomputing and Networking Center (PSNC)
ul. Noskowskiego 10
61-704 Poznań, Poland
meyer@man.poznan.pl

Roberto Pugliese
Sincrotrone Trieste S.C.p.A.
Strada Statale 14 - km 163.5
in Area Science Park
34012 Basovizza
Trieste, Italy
roberto.pugliese@elettra.trieste.it

Sandro Zappatore
Department of Communications, Computer and Systems Science (DIST)
University of Genoa
Via Opera Pia 13
16145 Genova, Italy
sandro.zappatore@cnit.it

ISBN 978-1-4419-5573-9 e-ISBN 978-1-4419-5574-6
DOI 10.1007/978-1-4419-5574-6
Springer New York Dordrecht Heidelberg London

Preface

Accessing remote instrumentation worldwide is one of the goals of e-Science. The task of enabling the execution of complex experiments that involve the use of distributed scientific instruments must be supported by a number of different architectural domains, which inter-work in a coordinated fashion to provide the necessary functionality. These domains embrace the physical instruments, the communication networks interconnecting the distributed systems, the service oriented abstractions and their middleware. Indeed, high-speed networking allows supporting sophisticated, bandwidth-demanding applications to an unprecedented level. However, the transport and access networks are not the only components that enable such applications. An equally important role is played by the distributed system middleware enabling Grids and cloud computing for data intensive applications.

Physical instrumentation lies at the bottom of these environments, but in many cases it represents the primary source of data that may need to be moved across networks and processed by distributed systems. It would be very helpful to deal with instruments that appear just as manageable resources like storage and computing systems. There have been and there are many attempts and progresses in this sense. However, given the large amount of different instruments and their application domains, understanding the common requirements, the user needs, the adaptation and convergence layers (among other aspects), is not a straightforward task. This is the objective of Remote Instrumentation Services (RIS), and this book, along with its predecessors in the same collection, tries to address some of the most relevant related aspects.

Involving user communities in this process is very important, as the diffusion and adoption of a specific service ultimately depends on the favor of the users it is addressed to. Quite a few software developments have failed to reach widespread diffusion among scientific users (just to cite a category), because of the lack of friendliness and easiness of use in dealing with the specific problems of a particular application domain. This aspect has been recognized in many ongoing projects and development efforts. In the European scenario, the DORII (Deployment of Remote Instrumentation Infrastructure) project, within which many contributors of this book operate, has focused its activity around the needs of the different user communities, directly involved in the project.

The chapters in the book are grouped into five areas, each addressing a specific aspect of remote instrumentation.

The first group, *Remote Instrumentation Services*, includes contributions dealing with the two main middleware components that continue to be developed in relation with the tasks of exposing instrumentation to the distributed computing environment and with offering a unified and multifunctional user interface. These are centered on the concepts of the Instrument Element (IE) and the Virtual Control Room (VCR), respectively. The contributions by F. Lelli and C. Pautasso and by K. Bylec et al. concern aspects of the IE, whereas that of R. Pugliese et al. discusses the implications of the Software as a Service paradigm in the context of a synchrotron radiation facility.

In the second group, *Support of Grid Functionalities*, we have included six chapters representing different features of Grid resource management an operations that are relevant in the context of RIS. The topics addressed comprise: data streaming optimization in interactive Grids (L. Caviglione et al.), interconnection of service and desktop Grids (P. Kacsuk et al.), automation of Service Level Agreements (C. Kotsokalis and P. Wieder), storage and analysis infrastructure for high data rate acquisition systems (M. Sutter et al.), visualization tools in support of resource discovery (A. Merlo et al.), and scheduling in a multi-broker Grid environment (A. Di Stefano and G. Morana).

Contributions in the third group are devoted to *Networking*, one of the key supporting technologies that enable the interconnection of data sources and the transport of data. The first three chapters concern higher-layer aspects of networking, namely: analyzing the design of overlay network topologies (D. Adami et al.); the use of peer-to-peer paradigms for file transfers in a Grid filesystem (N. Kasioumis et al.); the context-aware management of heterogeneous autonomic environments (A. Zafeiropoulos and A. Liakopoulos). The last two chapters in this group describe the status, evolution and research aspects of two National Research and Education Networks (NRENs) in Italy (M. Reale and U. Monaco) and Poland (A. Binczewski et al.), respectively.

The fourth group of chapters touches application environments in various user communities. These include eVLBI (electronic Very Large Baseline Interferometry) and its exploitation of high-speed networks, by M. Leeuwinga, oceanographic applications (D. R. Edgington et al., and A. Cheptsov et al.), and road traffic data acquisition and modeling (L. Berruti et al.).

Finally, the last two chapters belong to the category of learning environments, where Remote Instrumentation plays a key role of increasing importance. S. Jeschke et al. describe the main features and demonstrator scenarios of BW-eLabs (networked virtual and remote laboratories in the Baden-Württemberg region of Germany), whereas M. J. Csorba et al. report on a distributed educational laboratory that is part of the "Wireless Trondheim" initiative in Norway.

All contributions in this book come from the selection and extension of papers presented at the 4th International Workshop on Distributed Cooperative Laboratories – "Instrumenting" the Grid (INGRID 2009), held in Alghero, Italy, in April 2009, which focused on the theme of RIS and their supporting eInfrastructure. We

wish to thank all that contributed to the success of the Workshop, as participants and organizers. Special thanks go to the two keynote speakers, Prof. Tatsuya Suda, from the University of California at Irvine (UCI), and Dr. Monika Kącik, from the European Commission, Information Society and Media DG, Unit F3 'GÉANT & e-Infrastructure'.

Genova, Italy — Franco Davoli
Poznań, Poland — Norbert Meyer
Trieste, Italy — Roberto Pugliese
Genova, Italy — Sandro Zappatore

Contents

Contributors

D. Adami CNIT Research Unit Department of Information Engineering, University of Pisa, Pisa, Italy, davide.adami@cnit.it

Steinar H. Andresen Department of Telematics, Norwegian University of Science and Technology, N-7491 Trondheim, Norway, steinara@item.ntnu.no

Z. Balaton MTA SZTAKI, Budapest, Hungary, balaton@sztaki.hu

Luca Berruti CNIT – University of Genoa Research Unit, Genoa, Italy, luca.berruti@cnit.it

Artur Binczewski Poznań Supercomputing and Networking Center, Poznań, Poland, artur@man.poznan.pl

Randal Butler National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA, rbutler@ncsa.illinois.edu

K. Bylec Poznań Supercomputing and Networking Center, Poznań, Poland, katarzyna.bylec@man.poznan.pl

Carlo Caligaris DIST – Department of Communications, Computer and Systems Science, University of Genoa, Genoa, Italy, carlo.caligaris@dist.unige.it

C. Callegari Department of Information Engineering, University of Pisa, Pisa, Italy, christian.callegari@iet.unipi.it

L. Caviglione Institute of Intelligent Systems for Automation (ISSIA), Genova, Italy, luca.caviglione@ge.issia.cnr.it

C. Cervellera Institute of Intelligent Systems for Automation (ISSIA), Genova, Italy, cristiano.cervellera@ge.issia.cnr.it

A. Cheptsov High-Performance Computing Center, University of Stuttgart, Stuttgart, Germany, cheptsov@hlrs.de

Andrea Clematis IMATI-CNR, Genova, Italy, clematis@ge.imati.cnr.it

Angelo Corana IEIIT-CNR, Genova, Italy, corana@ieiit.cnr.it

Máté J. Csorba Department of Telematics, Norwegian University of Science and Technology, N-7491 Trondheim, Norway, csorba@item.ntnu.no

Alessio Curri Scientific Computing Group, Information Technology Department, ELETTRA Sincrotrone Trieste, Trieste, Italy, alessio.curri@elettra.trieste.it

Daniele D'Agostino IMATI-CNR, Genova, Italy, dago@ge.imati.cnr.it

Livio Denegri DIST – Department of Communications, Computer and Systems Science, University of Genoa, Genoa, Italy, denegri@dist.unige.it

Antonella Di Stefano Department of Information and Telecommunication Engineering, Catania Univerisity, Catania, Italy, antonella.distefano@diit.unict.it

Duane R. Edgington Monterey Bay Aquarium Research Institute, Moss Landing, CA 95039, USA, duane@mbari.org

Z. Farkas MTA SZTAKI, Budapest, Hungary, zfarkas@sztaki.hu

Terry Fleury National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA

Paolo Gamba Department of Electronics, University of Pavia, Pavia, Italy, paolo.gamba@unipv.it

Vittoria Gianuzzi DISI - Università degli Studi di Genova, Genova, Italy, gianuzzi@disi.unige.it

S. Giordano Department of Information Engineering, University of Pisa, Pisa, Italy, stefano.giordano@iet.unipi.it

Kevin Gomes Monterey Bay Aquarium Research Institute, Moss Landing, CA 95039, USA, kgomes@mbari.org

J. Gracia High-Performance Computing Center, University of Stuttgart, Stuttgart, Germany, gracia@hlrs.de

John Graybeal Monterey Bay Aquarium Research Institute, Moss Landing, CA 95039, USA, graybeal@mbari.org

M. Hardt Karlsruhe Institute of Technology, Steinbuch Centre for Computing, Eggenstein-Leopoldshafen, Germany, hardt@kit.edu

V. Hartmann Karlsruhe Institute of Technology, Institute for Data Processing and Electronics, Eggenstein-Leopoldshafen, Germany

Eckart Hauck RWTH Aachen University, Aachen, Germany, hauck@zlw-ima.rwth-aachen.de

Robert Herlien Monterey Bay Aquarium Research Institute, Moss Landing, CA 95039, USA, bobh@mbari.org

T. Jejkal Karlsruhe Institute of Technology, Institute for Data Processing and Electronics, Eggenstein-Leopoldshafen, Germany, thomas.jejkal@ipe.fzk.de

Sabina Jeschke RWTH Aachen University, Aachen, Germany, sabina.jeschke@zlw-ima.rwth-aachen.de

P. Kacsuk MTA SZTAKI, Budapest, Hungary, kacsuk@sztaki.hu

Prajwalan Karanjit Department of Telematics, Norwegian University of Science and Technology, N-7491 Trondheim, Norway, karanjit@stud.ntnu.no

Nikolas Kasioumis National Technical University of Athens, Athens, Greece, nikolas@ece.ntua.gr

B. Koller High-Performance Computing Center, University of Stuttgart, Stuttgart, Germany, koller@hlrs.de

Constantinos Kotsokalis Dortmund University of Technology, Dortmund, Germany, constantinos.kotsokalis@udo.edu

George Kourousias Scientific Computing Group, Information Technology Department, ELETTRA Sincrotrone Trieste, Trieste, Italy, george.kourousias@elettra.trieste.it

Pavlos Kranas National Technical University of Athens, Athens, Greece, pkran@ece.ntua.gr

Michael Krüger University of Freiburg, Freiburg, Germany, michael.krueger@fmf.uni-freiburg.de

Matteo Lanati Eucentre, Pavia, Italy, matteo.lanati@eucentre.it

P. Lazzari Department of Oceanography, Istituto Nazionale di Oceanografia e di Geofisica Sperimentale (OGS), Trieste, Italy, plazzari@ogs.trieste.it

M. Leeuwinga Joint Institute for VLBI in Europe (JIVE), 7990 AA Dwingeloo, The Netherlands, leeuwinga@jive.nl

Francesco Lelli Faculty of Informatics, University of Lugano, Lugano, Switzerland, firstname.lastname@usi.ch

A. Liakopoulos Greek Research and Technology Network, 11527 Athens, Greece, aliako@grnet.gr

Andrea Del Linz Scientific Computing Group, Information Technology Department, ELETTRA Sincrotrone Trieste, Trieste, Italy, andrea.dellinz@elettra.trieste.it

R. Marcialis Institute of Intelligent Systems for Automation (ISSIA), Genova, Italy, roberto.marcialis@ge.issia.cnr.it

Alessio Merlo IEIIT-CNR, Genova, Italy; DISI - Università degli Studi di Genova, Genova, Italy, alessio@ieiit.cnr.it

Ugo Monaco Consortium GARR – The Italian Research and Academic Network, Roma, Italy, ugo.monaco@garr.it

Giovanni Morana Department of Information and Telecommunication Engineering, Catania Univerisity, Catania, Italy, gmorana@diit.unict.it

S. Mueller Poznań Supercomputing and Networking Center, Poznań, Poland, szymon.mueller@man.poznan.pl

G. Nencioni Department of Information Engineering, University of Pisa, Pisa, Italy, gianfranco.nencioni@iet.unipi.it

Wolfgang Osten University of Stuttgart, Stuttgart, Germany, osten@ito.uni-stuttgart.de

M. Pabiś Poznań Supercomputing and Networking Center, Poznań, Poland, uranium@man.poznan.pl

M. Pagano Department of Information Engineering, University of Pisa, Pisa, Italy, michele.pagano@iet.unipi.it

Cesare Pautasso Faculty of Informatics, University of Lugano, Lugano, Switzerland, firstname.lastname@usi.ch

Marco Perrando DIST – Department of Communications, Computer and Systems Science, University of Genoa, Genoa, Italy, marco.perrando@gmail.com

Olivier Pfeiffer Technische Universität Berlin, Berlin, Germany, pfeiffer@math.tu-berlin.de

Milan Prica Scientific Computing Group, Information Technology Department, ELETTRA Sincrotrone Trieste, Trieste, Italy, milan.prica@elettra.trieste.it

Roberto Pugliese Scientific Computing Group, Information Technology Department, ELETTRA Sincrotrone Trieste, Trieste, Italy, roberto.pugliese@elettra.trieste.it

Mario Reale Consortium GARR – The Italian Research and Academic Network, Roma, Italy, mario.reale@garr.it

Thomas Richter University of Stuttgart, Stuttgart, Germany, thomas.richter@rus.uni-stuttgart.de

S. Salon Department of Oceanography, Istituto Nazionale di Oceanografia e di Geofisica Sperimentale (OGS), Trieste, Italy, ssalon@ogs.trieste.it

Stanisław Starzak Technical University of Łodz, Łodz, Poland, starzak@man.lodz.pl

R. Stotzka Karlsruhe Institute of Technology, Institute for Data Processing and Electronics, Eggenstein-Leopoldshafen, Germany, rainer.stotzka@kit.edu

Maciej Stroiński Poznan Supercomputing and Networking Center, Poznan, Poland, stroins@man.poznan.pl

M. Sutter Karlsruhe Institute of Technology, Institute for Data Processing and Electronics, Eggenstein-Leopoldshafen, Germany, michael.sutter@kit.edu

Panayiotis Tsanakas National Technical University of Athens, Athens, Greece, panag@cs.ntua.gr

Von Welch National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA, vwelch@ncsa.uiuc.edu

Philipp Wieder Dortmund University of Technology, Dortmund, Germany, philipp.wieder@udo.edu

M. Wojtysiak Poznań Supercomputing and Networking Center, Poznań, Poland, mariusz.wojtysiak@man.poznan.pl

P. Wolniewicz Poznań Supercomputing Networking Center, Poznań, Poland, pawelw@man.poznan.pl

A. Zafeiropoulos Greek Research and Technology Network, 11527, Athens, Greece, tzafeir@grnet.gr

Sandro Zappatore CNIT – University of Genoa Research Unit, Genoa, Italy, sandro.zappatore@cnit.it

Part I
Remote Instrumentation Services

Design and Evaluation of a RESTful API for Controlling and Monitoring Heterogeneous Devices

Francesco Lelli and Cesare Pautasso

Abstract In this paper we apply the REST principles to the problem of defining an extensible and lightweight interface for controlling and monitoring the operations of instruments and devices shared on the Grid. We integrated a REST Service in the Tiny Instrument Element (IE) that has been used for an empirical evaluation of the approach demonstrating that this implementation can coexist with a Web Service back-end and be used in parallel where is needed. Finally we present a preliminary performance comparison with the WS-$*$ compliant implementation.

1 Introduction

The remote control and monitoring of devices and experiments requires many interactions between the instruments and the computational Grid. Scientific equipment must be accessed by running jobs that need to interact with the instrument while performing some computation. These jobs are also interactive, as the users need to be able to use them to monitor and steer the instrument operations. In addition, in most demanding use cases, such as instruments for high energy physics experiments, achieving the required performance and quality of service guarantees represents an important challenge [1].

In the past few years modern web based companies offered multiple ways for accessing the services that they provide. The most popular approaches are based on the Web Service technology stack or on Representational State Transfer (REST) [2]. From a technical point of view both approaches have strengths and weaknesses and the adoption of a particular solution is really use case dependent [3].

The REST architectural style has been introduced to give a systematic explanation to the scalability of the HTTP protocol and the rapid growth of the World Wide Web. Its design principles have been recently adopted to guide the design of the next generation of Web services called RESTful Web services [4]. The benefits of REST

F. Lelli (✉)
Faculty of Informatics, University of Lugano, via Buffi 13, 6900 Lugano, Switzerland
e-mail: firstname.lastname@usi.ch

F. Davoli et al. (eds.), *Remote Instrumentation Services on the e-Infrastructure*,
DOI 10.1007/978-1-4419-5574-6_1,

lie in the simplicity of the technology stack required to build a Web service and in the recognition that most Web services are indeed stateful entities.

In this paper we investigate how to apply the REST design principles to give a lightweight solution to the problem of monitoring and controlling scientific instruments. We build a set of APIs for controlling and monitoring devices consisting of an oculate selection of the resource URIs and a precise description of their representation, as it is exchanged in the request and response of each method applied to a URI published by the device. We integrated these APIs in the Tiny Instrument Element (IE) [5, 6] that has been used for an empirical evaluation of the approach. The Original Tiny IE Web Service interface exposes methods that are similar to the ones developed in international cooperations like GridCC [7], RINGrid [8] and DORII [9]. Since no standards have been produced by the RISGE [10] Research Group yet we decided to demonstrate the feasibility of a REST design by implementing all the methods that are exposed by the original WS-∗ compliant interface used to monitor and control instruments. Our tests and measurements indicate better performance than the classical Web Service implementation. Finally it is worth noting that a WS-∗ and a REST based implementation can coexist and be used in parallel where is needed. The rest of this paper is structured as follows: Section 2 presents a selection of works that are relevant for the purpose of this paper while Section 3 introduces our proposed REST APIs. Finally in Section 4 we presents some performance comparison between Web Service (WS) and REST and in Section 5 we summarize our experience and we draw our conclusions.

2 Related Work and Background

Traditional efforts in providing remote access to equipment propose a common instrument middleware based on Web Services using SOAP over HTTP as a communication layer and specific WSDL interfaces [1, 11]. However modern software products propose different approaches for consuming stateful resources.

In [12] Foster et al. present and compare the four most common techniques for defining interactions among Web Services to support stateful resources (REST, WS-RF [13], WS-Transfer and “non standard”). REST is emerging as a lightweight technique for the remote consumption of services in modern e-business applications. Therefore we decided to investigate if this approach may be adopted in the context of accessing remote instruments. So far not much effort has been spent in this task; however, the following contributions are relevant for this objective.

In [4] techniques on how to build RESTful Web Services are underlined and a detailed comparison between REST and WS-∗ can be found in [3]. REST has also started to make inroads in different application domains. In [14], for example, a set of REST APIs for accessing mobile phones information such as photos, tags and manipulating device properties is presented.

In this paper we present a definition of a RESTful service for accessing, controlling, and monitoring remote instruments. Our proposed REST APIs (described in Section 3) maintain all the functionality that was previously exposed via a Web

Service interface thus showing that REST can be a good example of an alternative technology platform for instrument management. The rest of this section continues with a description of the Resource Oriented Instrument Model that we are considering (Section 2.1) and with a brief background description about REST where we outline its most relevant features (Section 2.2).

2.1 Resource Oriented Instrument Model

Considering the heterogeneous nature of instruments, one current shortcoming is that the applications that use them must have a complete operational model of the instruments and sensors they work with.

We can consider a generic model for a device consisting of a collection of parameters, attributes and a control model, plus an optional description language [1]:

- *Parameters*: are variables on which the instrument configuration depends, like range or precision values of a measure;
- *Attributes*: refer to the properties of the actual object that the instrument is measuring, such as the values that are being measured;
- *Finite State Machine*: this defines a control model, which represents the list of commands that the instrument can support. Commands are usually related using a Finite State Machine but in principle different formalisms (such as Petri Nets, Rule-based systems, etc.) may be adopted.
- *XML-based description language* that provide information about the semantic of the particular instrument such as SensorML [15] or OWL [16] etc.

The main difference between parameters and attributes concerns the access patterns that should be supported to read their data values. While parameters are typically read by polling, attributes should additionally support also an event-based or stream-based publish/subscribe approach. Therefore, both push and pull access patterns must be supported for some kinds of attributes.

This model is used for the representation of generic instruments. Our goal is also to provide support for more complex systems, where devices are logically (or physically) grouped into hierarchies for aggregating data and/ or distributing commands in more convenient ways. Therefore a way to retrieve the topology of the devices must be provided as well.

The code developed for controlling and monitoring devices is usually difficult to develop and expensive to maintain especially when the underlying instrument hardware is changed and/or improved. A primary design goal of this model is to externalize the instrument description so that applications can build an operational model “on the fly”. This approach makes it possible to preserve investments in codes as instrument hardware evolves and to allow the same code to be used with several similar types of instruments. Different representations of this model can be provided in order to let user decide the most convenient way for accessing and controlling the physical instrument.

2.2 REST Background

In the following we give a quick overview over the design principles and constraints of the REST architectural style. For more information we refer the interested reader to [2, 4, 17].

A RESTful service exposes its state, data, and functionality through the *resource* abstraction. Each resource is identified by a Uniform Resource Identifier (URI [18]). Its state can have one or more representations and can be handled with a limited and predefined set of methods. A resource can negotiate with clients which representation format should be used to exchange data and also can inform its clients about the supported methods. The set of methods that can be used to manipulate and interact with a resource is the following.

– *GET*: retrieve the current state of a resource.
– *PUT*: update the state of a resource.[1]
– *POST*: create a resource within the current one.
– *DELETE*: delete a resource.
– *OPTIONS*: reflect upon which methods are allowed on a given resource.

These methods are commonly found in the HTTP protocol and carry additional semantics which helps to support stateless interactions (where each request is self-contained) and idempotency (where requests can be repeated without side-effects). The only method which is unsafe, i.e., cannot be retried without side-effects, is *POST*.

Whereas – according to the *uniform interface principle* – the set of methods provided by a resource is fixed to the previous ones, REST does not constrain the set of resources that are published by a service. Thus, the expressiveness of the interface lies in the selection of a URI naming scheme and in the definition of the resource representations that are exchanged for each request method, as opposed to the freedom of defining a specific set of methods for each service interface, like in traditional Web services.

3 REST APIs for Remote Controlling and Monitoring Instruments

In this section we apply the REST design guidelines to the definition of an API to control and monitor instruments. We first define the set of resource URIs (Section 3.1) and specify which of the GET, POST, PUT, and DELETE methods can be applied. Then in Section 3.2 we define the resource representation formats used to exchange data with an instrument.

[1] If a resource does not exist, create it.

This REST API has been directly implemented using the HTTP protocol, which also provides security, access control, accounting, and exception signaling where needed.

3.1 Resource URI Design

In defining the URI for addressing instrument resources we follow this structure:

/Context/ <id>/Instrument/ <id>/ <instrument-resource>

where */Context/<id>* represents a configuration of the instrument itself, while */Instrument/<id>* represents a unique identifier of the instrument within the instrument element. A different naming convention for representation of the same concept may be adopted without changing the final result, however this approach to the design of "nice" URIs is one of the most used [4]. Note that this URI structure maintains the same structure and granularity of the addressing information of the original Web Service interface. Therefore this interface does not change the number of requests needed by the clients to perform the same operations.

Finally *<instrument-resource>* represents a resource published by the instrument such as a Parameter, an Attribute or the State machine. As presented in Section 2.1, an instrument must also allow introspection. To do so, for each instrument we define the following URIs that clients can use to discover more information about the instrument capabilities:

/Context/<id>/Instrument/<id>/Description
/Context/<id>/Instrument/<id>/Status
/Context/<id>/Instrument/<id>/FSM
/Context/<id>/Instrument/<id>/Parameters
/Context/<id>/Instrument/<id>/Attributes

where:

- *Description*: represents the description of the instrument in a given Language (SensorML [15] or OWL [16] , plain text, etc)
- *Status*: get the current status of the instrument.
- *Finite State Machine (FSM)*: inspect the finite state machine representation that is mapped as a set of transitions plus an initial state
- *Parameters*: retrieve the list of parameters exported by the instrument
- *Attributes*: retrieve the list of attributes exported by the instrument

Few of these resources (like the FSM) may have a non trivial representation, which will be defined in Section 3.2.

Table 1 summarizes the resources used for representing an IE and gives a detailed specification of which methods are allowed for each resource.

Table 1 REST model for controlling and monitoring instruments

Method	URI	Description
GET	/Context	return the list of possible instrument configurations or instruments topologies
GET	*/Context/<id>/*	return the list of intruments that are accessible in a given configuration
GET	*/Context/<id>/Instrument/<id> /Description*	Read the description of the device(s)
GET	*/Context/<id>/Instrument/<id> /Status*	Read the current instrument status
GET	*/Context/<id>/Instrument/<id> /Parameters*	Retrieve a list of parameters of the given instrument
GET/PUT/POST/ DELETE	*/Context/<id>/Instrument/<id> /Parameter/<id>*	Access the description of individual parameters
GET	*/Context/<id>/Instrument/<id> /Attributes*	Retrieve a list of attributes of the given instrument
GET/PUT/POST/ DELETE	*/Context/id/Instrument/<id> /Attributes/<id>*	Access the description of individual attributes
GET	*/Context/<id>/Instrument/ <id>/FSM*	Read the finite state machine description of the instrument
GET	*/Context/<id>/Instrument/ <id>/FSM/Transition/<id>*	Read the description of a transition
GET	*/Context/<id>/Instrument/ <id>/Commands*	Read the description of a command
POST	*/Context/<id>/Instrument/ <id>/Command/<id>*	Execute a command
PUT	*/Context/<id>/ Instrument/<id>/Transition/<id>*	Execute a Transition

More in detail, the URI */Context/<id>/Instrument/<id>* has the following semantics when used in conjunction with each method:

- GET */Context/<id>/Instrument/<id>* : Retrieve the list of instruments controlled/ supervised by the given instrument-id
- PUT */Context/<id>/Instrument/<id>* : If it is not already present, it creates an instance of the given instrument-id by instantiating the proxy for the real instrument. Otherwise it simply configures an existing proxy.
- DELETE /Context/<id>/Instrument/<id> : De-instantiate the proxy.

This approach to the modeling of instruments with resources has the following implications:

- Clients can browse the possible set of configurations and the list of instruments using the */Context* and *Context/<id>* .

– Clients can get information about the instruments using /Description, /Status, /Parameters, /Attributes, /Commands and /FiniteStateMachine.
– Clients can use the URI */Context/<id>/Instrument/<id>* for introspection and for instantiating the instrument proxy control structure.
– Clients can execute commands and trigger FSM transitions using the */Command/<id>* and */Transition/<id>* URIs.
– We decided to allow a POST and DELETE commands on Parameters and attributes. However few instruments may not allow such operations. In this case the error 405 (Method Not Allowed) can be used.
– Parameters, Attributes, Commands and FiniteStateMachine may return empty values because not all instruments may implement all these functionalities.
– The URI structure maps the model presented in Section 2.1 trying to minimize the number of service requests needed in order to retrieve a conceptual information.
– Few complex structures have been used for representing information related to the instrument.

3.2 Resource Representation Format

Concerning the data, the API supports the exchange of data for different applications using formats such as ATOM, JSON, or binary JSON. ATOM is a popular XML format that is commonly adopted in Web data feeds while JSON is, compared to XML, a lightweight format for exchanging structured data [19].

In this paper we concentrate our attention in the XML representation of the information but similar considerations could be repeated for different serialization formats.

What follows is an XML representation of a Parameter:

```
<Parameter>
.       <Name>
.       <Value>
.       <Unit>
.       <Min>
.       <Max>
.       <Description>
.       <Parameter>
.       ....
.       </Parameter>
.       <Parameter>
.       ....
.       </Parameter>
</Parameter>
```

We choose a complex XML representation of the information because parameters have the possibility of containing other parameters. Therefore a representation

/Parameter/<id>/<Subparameter/<id> may cause an excessive fragmentation of the interface thus increasing the number of requests needed in order to retrieve the complete parameter value.

Additional resources are represented as follows.

- *Attributes*: They are represented in a structure that is quite similar to the one used by parameters
- *Finite State Machine*: It is represented as a set of Transitions plus an initial state. (<initialState> <Transition> <Transiton> ...)
- *Transitions*: are represented as a command, an initial state and an arriving state. (<fromState> <toState> <Command>)
- *Commands*: are represented as a Name plus a set of Parameters. (<command-Name> <Parameter>)

3.3 Usage Scenarios

In this section we present few usage scenarios of the proposed REST APIs. Service requests related to instrument introspection may require 1 or 2 calls while the majority of the requests for controlling and monitoring the devices can be handled in a single call. We can also note that in terms of the supported use cases there are not many difference between the REST APIs and the original WS-∗ APIs [1].

Controlling the List of available Instruments
An Operator can control the list of available Instruments sending a GET request to the IE (URI: /Context) in order to retrieve the list of possible configurations and use this information in order to access each context (URI: */Context/<id>*).

Accessing the current setting of an instrument
The Parameters of an Instrument represents the current setting of an instrument. The list can be retrieved using a GET request to */Context/<id>/Instrument/<id>/Parameters* where */Instrument/<id>* identifies a device with configuration */Context/<id>/*.

Retrieve the list of Instruments controlled by a given instrument
Assuming that the URI */Context/<id>/Instrument/<id>* represents the instrument supervisor, the list of controlled instruments can be retrieved using a GET request to */Context/<id>/Instrument/<id>/* that returns the list of URIs identifying the instruments controlled by the supervisor.

Retrieve the list of available commands of a given instrument
The URI */Context/<id>/Instrument/<id>/* addresses the instrument that we want to use. We can retrieve the list of available commands sending a GET request to the URI */Context/<id>/Instrument/<id>/Command/*

Execute a Measure
We can execute a measurement command sending a POST request to the URI */Context/<id>/Instrument/<id>/Command/measure*; the results

of the measure can be fetched sending a GET request to the URI */Context/<id>/Instrument/<id>/Attributes/<id>*.

4 Performance Results

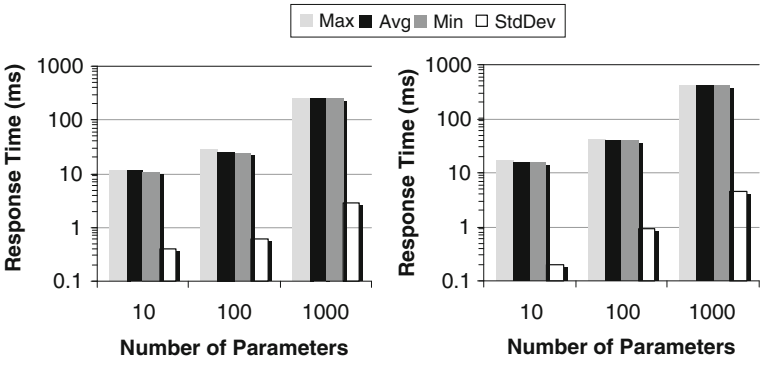
In this Section we present some preliminary benchmark tests. The goal is to understand the scalability, the overhead and the flexibility of our proposed REST APIs compared with the existing Web Service (WS) implementation. We run two different tests:

- *Throughput*: we measure the maximum number of requests per second that can be handled by the server as it is saturated with requests from clients.
- *Service Request Time*: we measure the time needed by a single client in order to perform a service invocation.

The experimental setup consists of a Tiny Instrument Element (IE) [5] containing only one Instrument Manager controlling a dummy instrument, which can be configured using a variable number of parameters (10, 100, 1000). The values of these parameters were automatically generated and fixed. One or more clients were retrieving the parameters from the instrument using the Web Service and the REST APIs.

The service was running on an Intel dual core 2.1 GHz with 2 GB of RAM with java 1.5. Apache Axis 1.4 [20] was used as WS provider while REST APIs were implemented using the RESTlet framework [21]. Both alternatives used an XML encoding of the parameters.

In Fig. 1 the service response time for both WS and REST is presented, while in Fig. 2 the maximum number of request per second is shown. Each measure has been taken 1000 times and in the plots Average, Minimum, Maximum and Standard



(a) REST (b) Web Service

Fig. 1 Response time comparison. (**a**) REST (**b**) Web Service

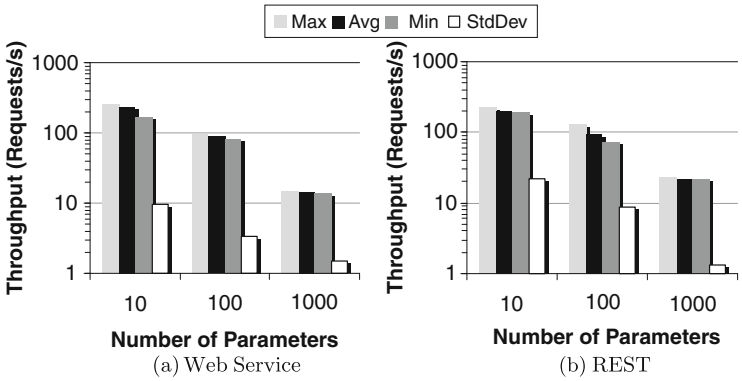


Fig. 2 Throughput comparison. (**a**) Web Service (**b**) REST

Deviations are reported. We note that in each experienced case (10, 100, 1000 Parameters) an IE based on REST offers a shorter response time. In terms of the throughput, the WS performed better for small messages (10 Parameters) while for big messages (1000 Parameters) REST achieved a higher throughput. This is a counter intuitive behavior as the overhead of SOAP should be higher fractionally for small messages. Moreover the experienced standard deviation of REST is larger than WS-∗. This may be due to the RESTlet framework [21] which begins to be stable but has not yet reached its maturity.

In order to provide a fair comparison, both WS and REST implementations were configured to exchange XML data. However with REST it would have been possible to use different resource representations, and – for example – choose a more lightweight data format such as JSON [19], for which we expect an additional reduction of the overhead.

5 Conclusion

In this chapter we investigate the possibility of applying the REST design principles to the problem of monitoring and controlling scientific instruments in order to provide a lightweight solution. We present a set of APIs for controlling and monitoring devices that follows the REST design principles. Our prototype demonstrates the feasibility of implementing all the methods that are exposed by the original WS-∗ compliant interface used to monitor and control instruments. Both implementations can coexist and be used in parallel. Tests and measurements of our prototype using XML payloads indicate better performance of the RESTlet container as opposed to the Axis-based Web service implementation. We plan to further leverage the flexibility of REST by introducing more lightweight data formats with the potential to further reduce the overhead of the instrument manager interface. Also, the possibility of directly using HTTP streams to perform measurement data streaming looks promising.

References

1. F. Lelli, E. Frizziero, M. Gulmini, G. Maron, S. Orlando, A. Petrucci, and S. Squizzato. The many faces of the integration of instruments and the grid. International Journal of Web Grid Services 3(3), 239–266, (2007)
2. R. Fielding, R.N. Taylor. Principled design of the modern web architecture. ACM Transactions on Internet Technology, 2(2), 115–150, (2002)
3. C. Pautasso, O. Zimmermann, F. Leymann. RESTful web services vs. big web services: Making the right architectural decision. 17th International World Wide Web Conference (WWW2008), Beijing, China, 805–814, (April 2008)
4. L. Richardson, S. Ruby. RESTful web services: web service for real world. O'Reilly, May 2007
5. Tiny Instrument Element Project: http://instrumentelem.sourceforge.net/, last accessed October 2010
6. F. Lelli, C. Pautasso. The tiny instrument element project. In proceedings of 4th International Conference on Grid and Pervasive Computing (GPC 2009), Geneva, Switzerland, May 2009
7. GridCC Project Web Site: http://www.gridcc.org/, last accessed May 2009, (2004)
8. RINGrid Project web site: http://www.ringrid.eu/, last accessed October 2010, (2008)
9. DORII Project web site: http://www.dorii.eu/, last accessed October 2010, (2010, in press)
10. Remote Instrumentation Services In Grid Environment: http://forge.gridforum.org/sf/projects/risge-rg, last accessed October 2010, (2010, in press)
11. D. McMullen, T. Devadithya, K. Chiu. integrating instruments and sensors into the grid with CIMA web services. Proceedings of the 3rd APAC Conference on Advanced Computing, Grid Applications and e-Research (APAC05), Gold Coast, Australia, September 2005
12. I.T. Foster, S. Parastatidis, P. Watson, M. Mckeown. How do I model state?: Let me count the ways. Communication of the ACM, 51(9), 34–41, (2008)
13. OASIS: Web Services Resources Framework (WSRF 1.2). http://www.oasis-open.org/committees/wsrf/. (April 2006), last accessed October 2010
14. C. Riva, M. Laitkorpi. Designing web-based mobile services with REST. Proceedings of Standard Performance Evaluation Corporation (SPEC) Benchmark Workshop, Vienna, Austria, January 2009
15. OGC Sensor Web Enablement:: www.opengeospatial.org/functional/?page=swe, last accessed May 2009, (2009)
16. M.K. Smith, C. Welty, D.L. McGuinness. OWL Web Ontology Language Guide,W3C. also available at http://www.w3.org/TR/owl-guide/, last accessed October 2010, (2010, in press)
17. R. Fielding. A little REST and relaxation. The International Conference on Java Technology (JAZOON07), Zurich, Switzerland. http://www.parleys.com/display/PARLEYS/A%20little%20REST%20and%20Re%laxation, last accessed October 2010. (June 2007)
18. T. Berners-Lee, R. Fielding, L. Masinter. Uniform Resource Identifier (URI): generic syntax. IETF RFC 3986, January 2005
19. D. Crockford. JSON: The fat-free alternative to XML. Proceedings of XML 2006, Boston, MA USA http://www.json.org/fatfree.html, last accessed October 2010. (December 2006)
20. S. Graham, S. Simeonov, T. Boubez, G. Daniels, D. Davis, Y. Nakamura. Building, web services with Java: Making sense of XML, SOAP, WSDL, and UDDI. Sams, December 2001
21. RESTlet Framework: http://www.restlet.org/, last accessed October 2010, (2010, in press)

Parametric Jobs – Facilitation of Instrument Elements Usage In Grid Applications

K. Bylec, S. Mueller, M. Pabiś, M. Wojtysiak, and P. Wolniewicz

Abstract The chapter discusses how to simplify integration of Grid applications in the Remote Instrumentation Infrastructure with usage of parametric extensions for JSDL. Capabilities of Parametric Sweep are presented spotlighting features most valuable in the frame of applying to Instrument Element enabled applications. Then a brief analysis of JSDL support in most common middlewares is followed by a real case study for one of the DORII project's applications. Starting from this application's use case two solutions in the field of parametric jobs are presented – g-Eclipse JSDL-Param library and Workflow Management System in DORII.

1 Introduction

Grid technologies are in constant development. Not only new tools emerge, but also the infrastructure elements are changing and new standards appearing. In the area of integrating new ideas into existing solutions the awareness of tools in the research field is significant. Combining together implementations from different levels can result in daily work improvements. This is a case for two relatively new Grid-related releases – Instrument Elements [1, 2], at the level of Grid infrastructure, and the Parameter Sweep [3] standard proposal, in the area of resources description languages.

With Instrument Elements the classical Grid infrastructure of computer and storage elements was enriched by the abstraction of real scientific devices. Providing Grid users with interfaces to instruments known from their daily work introduced new possibilities, but also the challenges. One of those challenges is to seamlessly integrate the Instrument Element into existing and newly created services in e-Infrastructures.

At the same time JSDL's [4] extension of Parameter Sweep was proposed to meet users' expectations for simplifying the description of Grid jobs that tend to become more complicated. Frequently their complexity emerges from a variety of

K. Bylec (✉)
Poznań Supercomputing and Networking Center, Poznań, Poland
e-mail: katarzyna.bylec@man.poznan.pl

F. Davoli et al. (eds.), *Remote Instrumentation Services on the e-Infrastructure*,
DOI 10.1007/978-1-4419-5574-6_2, © Springer Science+Business Media, LLC 2011

possible input data sets as well as program execution options. Though the main logic of the application stays the same – the description of Grid jobs may vary significantly depending on different execution switches. Parameter Sweep allows to seize the abstract description of application's work in one part of the document, while spreading its execution to differently parameterized job instances with special extension, e.g. varying with input data.

In fact this is the data where Instrument Elements and Parameter Sweep meet. Instrument Elements seen as a virtualization of application's data sources may produce a vast amount of information to be processed by applications. Existing applications may not be prepared to handle different sets of data at the same time, especially when each set requires special processing. To make Instrument Elements of use without re-developing existing software or providing complex user interface solutions, we propose the adaptation of the Parameter Sweep in the field of Grid tools.

An introduction to the Instrument Element is briefly covered in Section 2. Then, starting from the above problem, the JSDL Parameter Sweep extension's possibilities are described in Section 3. Moving from user perspective to technical issues this work discusses the status of support for JSDL and its extensions in some Grid environments. The current state of middlewares' capability of handling the JSDL influenced the above problem's solution implementations – presented in Sections 4 and 5, while describing the g-Eclipse [5] platform and DORII's [6] Workflow Management System. The description of problems encountered by developers when introducing parametric jobs into their applications is followed by a presentation of the Java library for JSDL Parameter Sweep and a real case study for one of the DORII project applications.

2 Instrument Elements

Instrument Elements extend the classical Grid infrastructure of Computing and Storage Elements with the concept of virtualized and distributed scientific devices. They were introduced to the Grid with GRIDCC [7] and RINGrid [8] projects and since then – as the unified interface to program devices was provided – the tendency in Grid tools development started to change. The focus is moving from providing access to distributed computing and storage resources to enabling Grid applications with full access to scientific instruments – including the acquisition of experimental data, monitoring, managing, maintenance and troubleshooting.

Providing scientific communities with remote instrumentation results in improvement of collaborative work in distributed teams and increase in devices usage efficiency, as with remote access they are made available to a wider audience, which decreases the idle time of the experimental tools.

Nowadays an increasing interest in Remote Instrumentation Infrastructure is noticed worldwide – to mention only CIMA [9] in the USA (The Common Instrument Middleware Architecture, applying SOA to integrate sensors, seen as real time data sources, into the Grid), SpectoGrid [10] in Canada (exposing Nuclear Magnetic Resonance for remote access) and DORII [6] in Europe. DORII is

described in more detail in Section 5. of this work; here, we will mention only the devices this projects aims to enable for the e-Infrastructure:

- **Sensors of earthquake early warning system** – constantly register seismic movements to warn in case of increased activity (European Centre for Training and Research in Earthquake Engineering, Italy);
- **Floats and programmable Gliders** – oceanographic and coastal observation instruments to measure drifts with the currents at specific depths, temperature, salinity, oxygen, chlorophyll and turbidity profiles (Istituto Nazionale di Oceanografia e di Geofisica Sperimentale, Italy);
- **Digital cameras, pressure and temperature sensors** – registering images and conditions of coastal line and beaches for measuring users density and calculate the weather line (IH Cantabria – Universidad de Cantabria, Spain);
- **X-Ray Diffraction detector** – data acquisition detectors for Diffraction Beamline supporting scientists in controlling if the data acquisition process is meeting quality requirements and for on-line data analysis (Sincrotrone Trieste ScpA, Italy);
- **Optical sensors for monitoring inland waters** – provide information in near real time about the water quality status (Ecohydros SL, Spain).

The above small subset from the category of Instrument Elements illustrates the main problem in the field of creating a homogenous interface of Remote Instrumentation Infrastructure – this is the heterogeneity of the domain. Instrument Elements range from all-time-up sensors to devices run on demand, which may require advanced reservation and accounting algorithms. Aside of the access scenarios, devices vary with network traffic they rise – from constant and low to irregular and bulk data transfers. This variety creates one of the reasons for bringing to life the OGF Remote Instrumentation Services in Grid Environment Research Group [11] (RISGE-RG), which gathers representatives from different fields and projects to collect existing solutions in accessing remote devices and aims at seizing the abstract approach for Remote Instrumentation. Its work will result in defining the foundation of an OGF Working Group, which will manage standardization in the field of Instrument Elements.

Though being relatively new in the Grid science, Instrument Elements acquire increasing interest. The scientific community seems to notice the fact that with Remote Instrumentation Infrastructure they may enable their work with devices for high performance and gain access to remote and distributed tools. Having this in mind we should focus on tools and technologies that would facilitate the usage of Instrument Elements in users' daily work. Such tools would be especially those with standards behind them – e.g. JDSL.

3 JSDL

Grid accessibility can be expressed with the power and capability of basic tools to accomplish standard Grid operations including job submission and data operations. Among them a well designed job description language is one of the most important

Grid features. In this section an OGF standard within this domain will be presented: the Job Submission Description Language (JSDL). A brief description of the general JSDL capabilities (3.1) is followed by an attempt to characterize its popularity within Grid middleware and middleware access tools (3.2). The parameter sweep feature is presented in more detail – discussing its power to express different parametric use cases (3.3). The section is completed with a description of the middleware capacity to handle parameter sweep (3.4).

3.1 JSDL Non-Functional Features

The Job Submission Description Language (JSDL) [4] is an XML-based computational[1] resource description language developed by OGF JSDL-WG [12]. The project was launched in Sept. 2003, and now it is publicly available in version 1.0 of the specification.

It is worth noticing that in JSDL-WG's goal, which is to specify an abstract standard of job description language independent of underlying middleware, JSDL is thought as a superset of other popular job description languages such as JDL or RSL, but also those e.g. based on Web Services invocations.

Because of this concept its structure is composed of the general purpose description part and domain or technology specific extensions.[2] Among those extensions are [13]:

- **POSIX Application** – defines which executable host destinations are POSIX standard compatible machines;
- **HPC Profile Application** – describes executables to be run as operating system processes;
- **SPMD Application** – Single Program Multiple Data extension for describing parallel applications with single executable;
- **Parameter Sweep** [3] – abstract description of jobs collection varying with some parameters as a single job, the *Proposed Recommendation* by OGF.[3]

3.2 JSDL Popularity

Because of its general and abstract way of describing jobs, JSDL is a very powerful language. Even if some of the available extensions are lacking desired functionalities JSDL's users may provide their own extensions – not only at the level

[1] Though it is named *computational* on the JSDL-WG web page it can be also used for data transfer operations alone.

[2] The extensible architecture is becoming a popular solution in resource description; it is also introduced in the Job Description (JD) Document by Globus Toolkit version 4.2.1. The difference is that Globus' GRAM4 was designed to support a single extension point (for Local Resource Manager) which is one of the reasons this middleware is lacking support for JSDL. See Table 1. for details.

[3] Announced 12.05.2009

Table 1 Support for JSDL in Grid middlewares

Middleware	Default JOB Description Language	JSDL Support	Details of JSDL support
gLite 3.2	Job Description Language (JDL)	Yes	As a patch to WMS UI 3.3, simple jobs only, plan to add bulk submission [14]. In older version it is possible to convert JSDL to JDL [15] with XSLT provided by OMII-Europe [16].
Globus Toolkit 2	Resource Specification Language (RSL)	No	N/A
Globus Toolkit 4	Job Description Document (JDD aka RSL 4)	Yes	Since version 4.0.5 GT is provided with GridWay Metascheduler that takes care of JSDL submission [17]. It also has support for JSDL HPC profile.
UNICORE 6	JSDL in server services (JSON in the UCC)		Job descriptions in JSON format are handled by the server side flexible execution backend (XNJS).

of job type but also in every existing XML element and attribute. Another important fact about JSDL is it has OGF support and right now should be considered a standard. Knowing all this one would expect JSDL being popular in Grid application domains, which is not the case. The only reasonable explanation for this state are historical reasons. Many of the Grid middlewares and tools developed for them have started before JSDL has confirmed its position. Some examples of introducing JSDL to Grid projects are presented in Table 1.

It is remarkable that none the most popular Grid middlewares made JSDL their default job description language. Support for it is provided in form of patches or additional tools. This solution is justified in cases when backward compatibility is a priority, but some of the projects (e.g. Globus Toolkit 4) introduced completely changed job descriptions instead of making a standard one their main format for describing Grid jobs. Therefore it is understandable that scientists tend to use old tools, even if usage of them means that the efficiency of work will not increase. It is not in the scope of this chapter to analyse the popularity of the JSDL language and reasons for it, nevertheless a quick conclusion here would be that even within the Grid domain (not so popular itself among other scientific fields) ideas and their releases lack good dissemination. This leads to sticking to old implementations and inhibition of spreading of new technologies. We are facing a vicious circle in case of JSDL and that is the reason why all efforts to support new ideas should be even more promoted, mainly by pointing fields in which their application may result in increase of efficiency. This is the case for introducing a Parametric Sweep extension to Remote Instrumentation Infrastructure.

3.3 Parameter Sweep Extension

The intuitive understanding of parameterizing a job is correct when it comes to Parameter Sweep [3]. It would be defined as *describing as a single job a set of many jobs varying among them with some detail*. This detail may be, e.g., the option of an executable or the name of an input file.

The above idea is not new in the Grid environment. gLite's JDL also supports parameters [18] (to be more precise – one parameter), but the difference here is significant; that is why we will provide a short description of Parameter Sweep capabilities, which are not limited to number of parameters, functions describing the values to be substituted or the ordering of parameters sweeping. All those aspects are covered in the following subsections.

3.3.1 Target of Substitution

The parameter element specifies which JSDL element should be substituted when resolving JSDL with parameters. The user can decide to sweep:

- **Any XML element or attribute of JSDL** – a value of XML element or attribute can be referenced using the XPath language [19]. Moreover XPath's substring function can be used to point only to part of the referenced string value. This functionality is covered with DocumentNode element.
- **A token inside a file external to JSDL** – in the same way as values inside a JSDL element are substituted, external text files can be modified. This is achieved through the FileSweep element.

The difference in behavior of FileSweep and DocumentNode is that the latter accesses values of XML entities pointed by name whereas FileSweep's input is a token, which will be substituted. In other words – after applying sweep to DocumentNode the user will still have the names of referenced XML entities with changed values and in case of FileSweep the token will no longer be present in the file, as it was replaced with the value given.

3.3.2 Values to be Substituted

Values to be sweeped for parameters are defined with the Function element. This is an abstract element and therefore can be seen as an extension point of the Parameter Sweep specification – the user can define a structure to meet his substitution requirements. By default Parameter Sweep provides three kinds of Functions (which don't have to be "function" in mathematical sense). Those are:

- **Array of values** – ordered set of unrelated data of the same type;
- **Loop** – loops are defined with starting, terminating and step values; it is also possible to exclude some of the values from the loop's range. The specification provides two default loop functions: LoopInteger and LoopDouble for integer and double values, respectively.

3.3.3 Order of Substitution

In case of many parameters definitions it is possible to steer the order of values substitution. This functionality is captured by the Assignment element, which couples the parameter with values to be sweeped for it. Because it is possible to nest Assignment elements as a side effect the user gains the ability of defining the dependencies between different sweeps as follows:

- **Sweep at the same time** – at each sweep of values one value is substituted for each parameter defined within JSDL. Parameters may be assigned to different functions and therefore may take different values at the same time. It is also possible to assign one function to more than one parameter. The number of resulting jobs is equal to the values set cardinality.
- **Independent sweep** – whole substitution for one parameter is done at one time while the rest of parameters take their default value. The number of resulting jobs is equal to the sum of all values sets cardinalities.
- **Nested sweep** – for one sweep of value of external parameter the whole substitution (of all values) is done for the internal one. The number of resulting jobs is equal to the multiplicity of values sets cardinalities for the nested elements.

3.4 Parametric Jobs at Middleware Level

We must stress the difference between job description and submission of job. The fact that the user is able to specify a set of jobs with one description is not equal to submitting only one job. It is up to the middleware implementation if after submission the parametric job will be seen as a single collection of jobs or as many independent Grid jobs. The difference concerns e.g. the execution environment settings or sharing of input/output files. The concept of a collection of jobs [18] and shared sandbox is known from gLite and JDL language. In this case JDL provides the ability to decide on a shared sandbox and gLite middleaware is able to fulfill this requirement. In case of Parameter Sweep for now there's no support for submission of a parametric job as a collection of jobs. At the level of JSDL it would be possible with a simple attribute extension, but we are still missing support from middlewares.

At the time of writing this chapter none of the existing Grid middlewares could handle the Parameter Sweep extension. Nevertheless in the following sections we will try to show that regardless of this fact using parametric jobs is reasonable. Introducing Parameter Sweep even only at the level of Grid clients results in simplification of user interfaces. It leads to abstraction of the job to be done and therefore allows to encapsulate the whole work delegated to the Grid as a workflow.

4 G-Eclipse Implementation

g-Eclipse [5] is project founded by the European Commission under the 6th Framework Programme (it lasted from July 2006 to December 2008) as well as the Eclipse Foundation [20] project. It builds on top of the Eclipse platform and

Table 2 g-Eclipse's middlewares support

Middleware	Available functionalities
GLite	Authentication and authorization – VOMS proxy extension; job management – editing, submission, monitoring (JDL/JSDL); data management (gsiFTP, LFC, RSM); GLUE information system; remote application debugging and deployment; workflow submission.
GRIA	Job management – editing, submission, monitoring (JSDL); data management (GRIA file store).
Globus Toolkit 2	Authentication and authorization – Globus proxy; job management – editing, submission, monitoring (RSL); data management (gsiFTP); gLogin.
Globus Toolkit 4	Support for GT 4.2 (not compatible with GT 4.0) Job management – editing, submission, monitoring (JDD aka RSL 4); information system (MDS)

provides a middleware independent architecture of workbench tools for accessing Grid resources. It is targeted at three users groups: Grid Users, Grid Application Developers and Grid Operators. g-Eclipse's exemplary middleware target was gLite, but support for other e-Infrastructure was also implemented – details are provided in Table 2.

Though designed and implemented as a graphical user tool, g-Eclipse can be also utilised as a Grid access library, especially taking into account work done within the DORII project (see Section 5.1 *WfMS and Common Library*). This approach is discussed in the following section, but first the original g-Eclipse solution is presented.

4.1 g-Eclipse JSDL Plug-Ins

g-Eclipse extends the Eclipse Platform's plug-ins architecture. It is composed out of layered plug-ins: at the bottom there are Eclipse's standard UI and resources management as well as extensions mechanism, on top of them the abstraction layer for Grid functionalities is build which is then extended – if needed – with middleware specific [21].

JSDL is introduced at the middle level – as the standard job description language for middleware abstraction. Still it is possible to submit the job description in middleware specific language but the recommended and – to some point – forced by architecture solution is to operate on the JSDL file and then transform it to specific language just before submission to the middleware.

g-Eclipse simplifies this recommended job management flow. It provides the JSDL Java model for developers and multipage graphical editor which hides the JSDL XML's complexity from the user. Supported JSDL extensions are POSIX and Parameter Sweep.

Because none of the supported middlewares handles JSDL Parameter Sweep, the parameterized JSDL files (compare Section 3) have to be preprocessed before submission. Preprocessing involves:

1. **resolving of parameters** – sweeping all values and generating resulting JSDL files without Parameter Sweep extension's elements;
2. **submission of each JSDL file** generated in the first step as an independent Grid job.

In the GUI – in Project view as well as in Jobs monitoring view – the parametric is represented as a single job with resulting sweeped files as children. Before the submission user is given the possibility to preview the values' array that will be used for sweeped job descriptions. This can be done in JSDL editor and won't affect the parametric JSDL file.

4.2 Java Library for Parameter Sweep

g-Eclipse is build on top of the Eclipse Platform and therefore depends on its architecture as well as on some low level model code e.g. for resources management. This is the reason why, though g-Eclipse's JSDL functionality is encapsulated within 3 plug-ins, it cannot be used outside of the Eclipse Platform. It would be a shame if this code – providing wide coverage of POSIX extension and one of the few existing implementations of Parameter Sweep – could not be reused by other projects. JSDL plug-ins can be easily integrated into RPC applications [22], but it is not always the case that an external project can be converted into Eclipse application. That is why we extracted parts responsible for Parameter Sweep handling. They are available in form of Eclipse independent Java library called Param-JSDL [23] under Eclipse Public Licence [24].

This library is independent of the g-Eclipse JSDL model – it operates on string elements. The entry point to param-jsdl.jar is the IParametricJsdlGenerator interface and its single method generator, which in turn takes IParametricJsdlHandler instance as an argument. IParametricJsdlHandler is a listener implementation for events that occur during processing of parametrized JSDL. Those are: start of generation, generation of new JSDL, cancel and finish, which developers can customize to suit their requirements.

The implementation in the background, to which instances of IParametricJsdlHandler register for generation events information, uses DOM XML representation and XPath language for traversing and modifying the JSDL structure. XML processing is done in memory – no I/O operations are involved – which results in efficiency increase (Fig. 1).

The speed of generation depends on the structure of JSDL and parameters number. It doesn't depend on parameters values. Results of tests performed on a middle class developer's PC computer (Intel ® Core ™ 2 CPU, T5500 @ 1.6 GHz, 2 GB RAM, Windows Vista ™ Home Basic 32-bits, Java ™ SE Runtime Environment, build 1.6.0_13-b03)) are visible in Figs. 2, 3, and 4. As a basic reference data set a JSDL with executable definition and one data staging was used, starting from one parameter with 3 values, whose number was increasing up to 15. For each JSDL 1500 substitutions were performed.

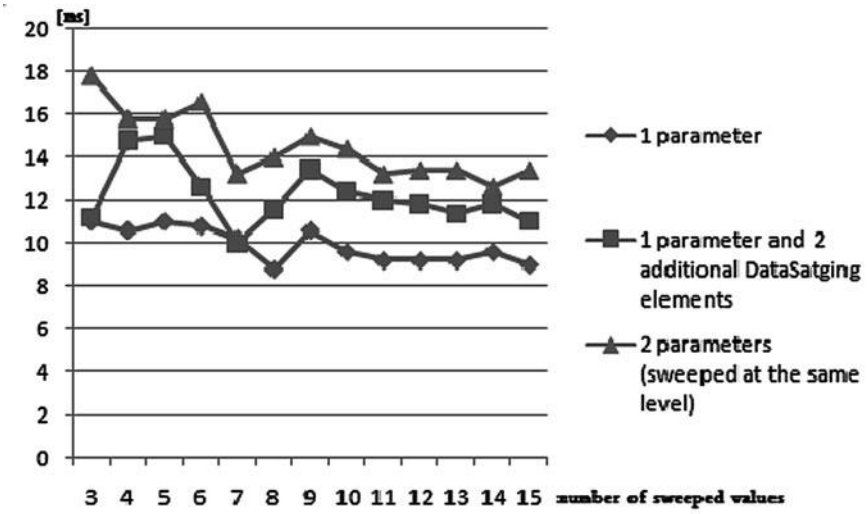


Fig. 1 Performance test results of JSDL-Param (own elaboration)

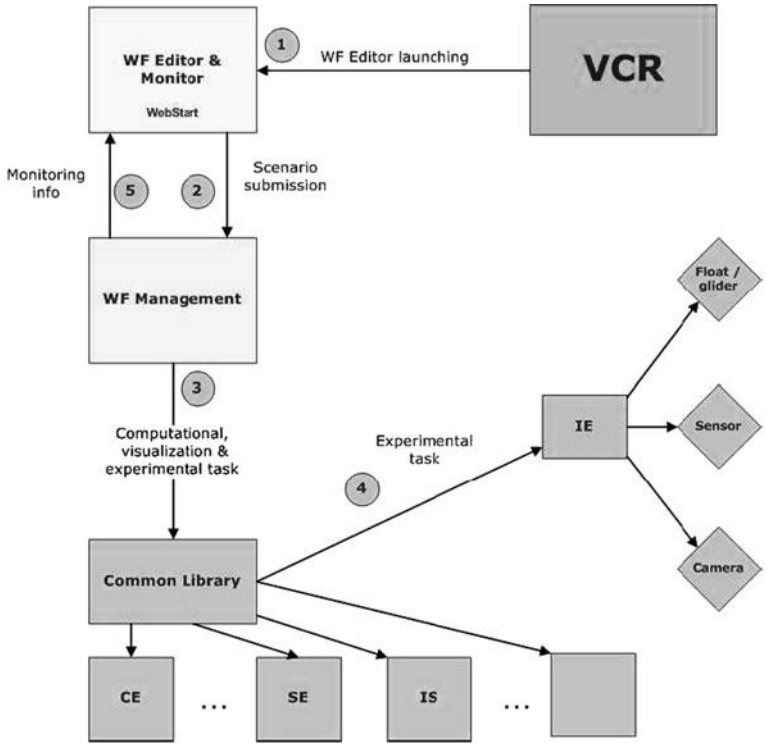


Fig. 2 Scheme of WfMS in DORII

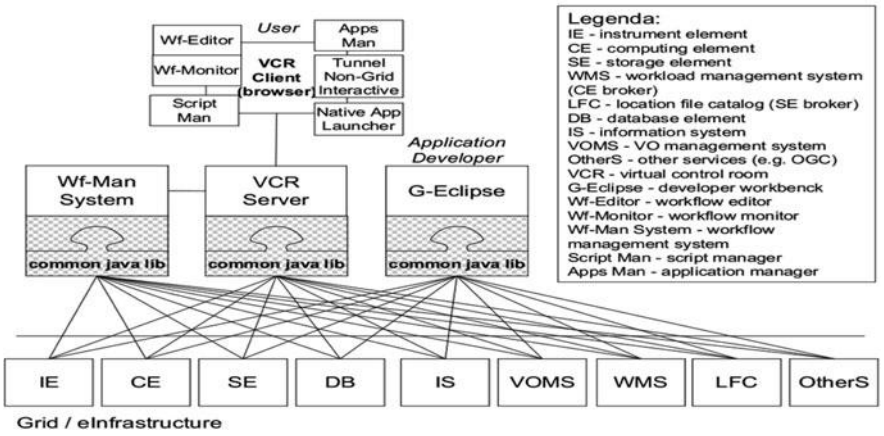


Fig. 3 DORII architecture

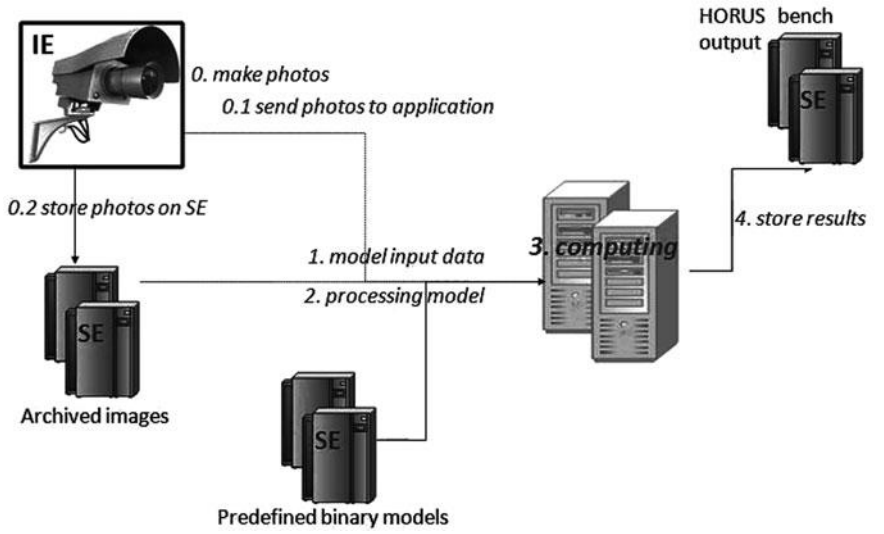


Fig. 4 HORUS_bench workflow

5 DORII Implementation

Deployment Of Remote Instrumentation Infrastructure (DORII) [6, 25] aims to introduce the Grid extended by Instrument Elements to scientific communities such as Earthquake community (with various sensor networks), Environmental science community, Experimental science community (with synchrotron and free electron lasers). The project goal is to integrate existing Instrument Elements with the Grid

infrastructure and develop applications handling them as well as provide high-end tools with a GUI to simplify maintenance and access to the deployed structure.

When first versions of Instrument Elements had been successfully installed users were able to start using high-level tools, among them – the VCR developed initially in GridCC [7] and the Workflow Management System, first introduced by the VLab project [26], whose idea was deployed with success e.g. in the EXPReS project [27, 28].

5.1 WfMS and Common Library

The Workflow Management System (WfMS) was designed to support users in running, managing and monitoring Grid applications instances, defined as a sequence of Grid jobs, i.e. workflows. In the DORII project workflows involve Instrument Elements, that may take place in the every part of the chain – being source of data for other workflows component as well as the data destination.

WfMS is composed out of two parts – GUI Workflow Editor and server side Workflow Manager. The Editor may be launched independently (as a Java Web Start application) or from within a VCR instance. It servers a workbench with predefined workflow instances that the user customizes before its submission to the Grid – specifying data and steering the flow between components, describing the experiment parameters and conditions. The workflow's definition may contain components representing operations on Instrument Elements, Grid job execution, storage access or prompting for user input if required. Grid components behind each of the workflow's block are restricted by the Virtual Organization (VO) of the user.

The workflow is started by the user from the Editor and then the steering is passed to the server side component. This is the Workflow Manager that is responsible for determining the order of the workflow's parts to be executed as well as for transforming them to concrete Grid operations (e.g. generating JSDLs, executing transitions or operations on Instrument Elements), for monitoring execution and managing data transfers. The server component of WfMS acts as a central management point of workflow execution and as a Grid access interface. For the latter it uses the Common Library component also developed within the DORII project. Details of the design are covered by Fig. 2.

The Common Library is a Java library designed to be a common access and functionality layer between integrated e-Infrastructure components. The idea that resulted in the Common Library creation was the observation that libraries for accessing the Grid infrastructure are distributed over many different projects with different goals and objectives, which means they may vary with technology (e.g. client or server side, web services based, etc.), programming language (Java or C) or may not be complete. The Common Library component exposes Grid access interfaces for services such as MyProxy and Instrument Element taken from EGEE and DORII projects, respectively, as well as Information System (BDII), WMS, and data management (gsiFTP, LFC, SRM) – extracted from g-Eclipse.

The relation between components described above is captured in Fig. 3 representing the DORII architecture.

5.2 HORUS_bench Application

HORUS_bench is one of the applications defined by the DORII project to be integrated into the Remote Instrumentation Infrastructure. It is developed by Instituto de Hidráulica Ambiental, Universidad de Cantabria, Spain and concerns the environmental data analysis.

HORUS_bench involves digital cameras, temperature and pressure sensors as Instrument Elements. Data gathered by them is used to monitor and simulate conditions of the beaches and coastal line of sea and rivers – especially calculating the water line, user density on the beaches, etc. Because of the distributed geographical nature of those devices as well as amounts of data they produce, scientists are lively interested into introducing HORUS_bench into the e-Infrastructure.

The use case is (see also Fig. 4):

- The gather input data from Instrument Elements – e.g. cameras taking pictures of beaches – they will come as an input stream to the application or be stored on SE and read from it;
- From the list of available image processing algorithms the user chooses the set that will be applied to input data. The chosen set of algorithms will constitute a binary file of the application;
- Application's input data and algorithms (in form of binary file) are taken by the HORUS_bench script run on the CE – the set of algorithms is applied to each of the images from input data gathered by Instrument Elements;
- Each file produced as a result of HORUS_bench processing is transferred as an output to a remote SE location.

5.3 Applying Parametric Jobs for HORUS_bench

Assuming that the HORUS_bench script takes one input file at once and this file has to be available on the script's path at the execution time, we propose the approach of submitting many HORUS_bench jobs, one for each file from the input set. Those jobs will be run in parallel, writing result files into the same remote location. This means that each job will have the same set of processing algorithms and the same executable. The only changing parameter will be the input file name (as the location stays the same).

Starting from the point where the user launches the HORUS_bench workflow from the Editor – the Workflow Manager transforms the information provided (input images location and processing algorithms) into a Parametric JSDL file. The simplified job description is presented in Fig. 5.

The solution described at the beginning of this subsection is reflected in JSDL design – parameterization is applied to job's data stagings (described in XPath language as /∗//jsdl:DataStaging[3]/jsdl:Source/jsdl:URI), but only to the file name part of JSDL's element (using XPath substring(/∗//jsdl:DataStaging [3]/jsdl:Source/jsdl:URI, 54, 1) function).

```
<?xml version="1.0" encoding="UTF-8"?>
<jsdl:JobDefinition>
  <jsdl:JobDescription>
    <!--  ... -->
    <jsdl:Application>
      <jsdl:ApplicationName>HORUS_bench</jsdl:ApplicationName>
      <jsdl-posix:POSIXApplication>
        <jsdl-posix:Executable>horus.sh</jsdl-posix:Executable>
      </jsdl-posix:POSIXApplication>
    </jsdl:Application>
    <jsdl:DataStaging>
      <jsdl:FileName>horus.sh</jsdl:FileName>
      <!-- ... -->
    </jsdl:DataStaging>
    <jsdl:DataStaging>
      <jsdl:FileName>algorithm_model_bin</jsdl:FileName>
      <!-- ... -->
    </jsdl:DataStaging>
    <jsdl:DataStaging>
      <jsdl:FileName>file1.png</jsdl:FileName>
      <jsdl:Source>
        <jsdl:URI>gsiftp://path/on/SE/to/Instrument/Element/output/file1.png</jsdl:URI>
      </jsdl:Source>
    </jsdl:DataStaging>
  </jsdl:JobDescription>
  <sweep:Sweep>
    <sweep:Assignment>
      <sweep:Match>substring(/*//jsdl:DataStaging[3]/jsdl:Source/jsdl:URI, 54, 1)</sweep:Match>
      <sweep:Mathc>substring(/*//jsdl:DataStaging[3]/jsdl:FileName, 5, 1)</sweep:Mathc>
      <sweepfunc:LoopInteger sweepfunc:end="99" sweepfunc:start="0" sweepfunc:step="1"/>
    </sweep:Assignment>
  </sweep:Sweep>
</jsdl:JobDefinition>
```

Fig. 5 Simplified parametric JSDL for HORUS_bench

It is worth noticing that in case of applying parameterization to data stagings elements two sweeps have to be defined. One for the resource path (as stated in the previous paragraph) and another for the name of the staged file. This may be seen as a disadvantage for users of Parameter Sweep, nevertheless before turning this into an entry point for discussion on user friendliness of the specification, we should be aware that Parameter Sweep is a low level language. It should be the users' tools responsibility to handle and hide such a situation from the user. Moreover coupling between data staging's path and its name is not always the case. The JSDL specification states that the FileName element defines a name used for file in sandbox, while the path points to a resource. In other words – during transfer of data staging to a sandbox its name may be changed.

The latter use case would be of use if we proposed a different approach to HORUS_bench parameterization. In the solution described above (sweep of input file's path and name) the application's script just looks for an image file on its execution path. We may imagine that this script was developed to read always a file of the same name (the input file name is hardcoded in the script). In that case two solutions are possible:

- To parameterize only the input file path in JSDL (not the FileName element which will take the hardcoded value from the script);
- To parameterize the input file path and name in JSDL and to apply the FileToken sweep to the HORUS_bench script).

The former is not much different from the solution chosen in WfMS, so we will skip its description. The latter would be only possible if the script is a local file, as it should be available at the moment of Parameter Sweep processing of JSDL. Having the script as a Grid storage location means that it is accessible only at the moment of execution of the job in the Grid sandbox, where neither the Parameter Sweep reference within JSDL nor the token definition inside the script would be properly interpreted.

When the WfMS has finished generation of JSDL from Fig. 4 the job description is submitted to the Grid using the Common Library's WMS interfaces. The way of handling JSDL by Common Library is based on g-Eclipse's model implying that the JSDL with Parameter Sweep is validated and then parsed by JSDL-Param library, which results in many non-parameterized JSDL descriptions, separately maintained by Common-Library in the process of Grid submission and monitoring.

5.4 Applying Parameterization at Different Levels

Details of the solution described above are distributed into two levels – implementation involves functionalities of the WfMS and Common Library components. In fact the lowest level is missing for the solution to be considered complete – the middleware support.

In gLite middleware while submitting a single job before the to a WMS the job is given its own sandbox. Data staging files are transferred to sandbox before the application is launched and outputs transferred to target destination after it has finished. Running HORUS_bench for a single input file out of GB of input results in heavy traffic load, not to mention the overload of WMS and job queues. That is why the presented solution is only a partial one for the described problem. It reduces the complexity of workflow at the level of user interface – it allows to hide details of the undelaying implementation and present the user only the main application's idea which, in this case, is bulk, not individual file processing. The server side workflow maintenance is also simplified – in the Computing Element block the steering float does not have to fork for many single jobs, as this is handled by the underlying Common Library's JSDL model and Param-JSDL library described in previous sections. Unfortunately, the lack middleware capability for handling Parameter Sweep does not allow for completing the process of facilitation of to the Instrument Element adaptation into HORUS_bench. Nevertheless – in comparison to the non-parametric description of HORUS_bench the gain is significant, measured mainly with the possibility of Grid enabling of the application.

6 Conclusions

Enabling existing and newly created services for a Remote Instrumentation Interface gains increasing interest among scientific fields. The expectation of efficiency improvement can be met by combining Instrument Elements usage with JSDL's

extension for parametric jobs – Parameter Sweep. First attempts to achieve this goal were made within the DORII project and its Workflow Management System solution for a sequence of Grid operations (including remote devices access as well as jobs submission). This was possible with the outcome of another project – g-Eclipse, whose early experience in Parameter Sweep could be ported to DORII.

Though the Parameter Sweep is of much help for enabling applications for Instrument Elements still the solution is not complete, as the middlewares lack the capability of JSDL and Sweep Parameter handling.

References

1. E. Frizzierol, M. Gulminil, F. Lelli, G. Maron, A. Oh, S. Orlando, A. Petrucci, S. Squizzato, and S. Traldil. Instrument Element: a new Grid component that enables the control of remote instrumentation. Workshop on Scientific Instruments and Sensors on the Grid, Trieste-Italy, 23–28 April, 2007
2. M. Płóciennik. RISGE-RG Collection of use cases, OGF Document Series web site, document GFD.168. http://ogf.org/documents/GFD.168.pdf, last accessed September 2010
3. JSDL Parameter Sweep Job Extension; http://www.ogf.org/documents/GFD.149.pdf, last accessed August 2009
4. Job Submission Description Language (JSDL) Specification, Version 1.0; http://www.ogf.org/documents/GDF.136/pdf, last accessed August 2009
5. g-Eclipse, project website, http://www.geclipse.eu, last accessed August 2009
6. Deployment of Remote Instrumentation Infrastructure (DORII), project website, http://www.dorii.eu, last accessed August 2009
7. GRIDCC, project website, http://www.gridcc.org, last accessed August 2009
8. RINGrid, project website, http://www.ringrid.eu, last accessed August 2009
9. T. Devadithya, K. Chiu, K. Huffman, D.F. McMullen. The common instrument middleware architecture: Overview of goals and implementation, http://cs.indiana.edu/~tdevadit/pubs/cima_isog05.pdf, last accessed August 2009
10. SpectroGrid, project webpage https://spectrogrid2.nrc.ca, last accessed August 2009
11. Remote Instrumentation Services In Grid Environment – RG (RISGE-RG), OGF Research Group webpage, https://forge.gridforum.org/sf/projects/risge-rg, last accessed August 2009
12. Job Submission Description Language WG, OGF Working Group webpage, https://forge.gridforum.org/sf/sfmain/do/viewProject/projects.jsdl-wg, last accessed August 2009
13. JSDL-WG Published Documents: JSDL HPC Profile Application Extension Version 1.0 and JSDL SPMD Application Extension Version 1.0, https://forge.gridforum.org/sf/docman/do/listDocuments/projects.jsdl-wg/docman.root.published_docs, last accessed August 2009
14. Gruppo GRID Datamat EGEE Wiki Pages, Patch #3040, http://trinity.datamat.it/projects/EGEE/wiki/wiki.php, last accessed August 2009
15. OMII-EU JRA1/Job Submission Task Wiki Pages, JSDL to JDL converter http://grid.pd.infn.it/omii/jsdl2jdl, last accessed August 2009
16. OMII-Europe, project webpage, http://www.omii-europe.org/, last accessed August 2009
17. Globus Aliance Community Webpage, GT 4.0.5 Incremental Release Notes, http://www.globus.org/toolkit/releasenotes/4.0.5/, last accessed August 2009
18. Gruppo GRID Datamat EGEE Wiki Pages, JDL Types, http://trinity.datamat.it/projects/EGEE/wiki/wiki.php?n=JDL.JDLTypes, last accessed August 2009
19. XML Path Language (XPath), Version 1.0, W3C Recommendation, http://www.w3.org/TR/xpath, last accessed August 2009

20. g-Eclipse: Tools for Grid and Cloud Computing, project webpage on Eclipse Foundation portal, http://eclipse.org/geclipse/, last accessed August 2009
21. g-Eclipse Consortium, g-Eclipse Final Architecture, http://www.geclipse.org/fileadmin/Documents/Deliverables/D1.8.pdf, last accessed August 2009
22. FZK's Savannah CVS source code repository, g-Eclipse example RPC applications – Gaussian, http://savannah.fzk.de/cgi-bin/viewcvs.cgi/fzk/geclipse/geclipse/demo_apps/eu.geclipse.gaussian.jmol/, last accessed August 2009
23. FZK's Savannah CVS source code repository, JSDL-Para library, http://savannah.fzk.de/cgi-bin/viewcvs.cgi/fzk/geclipse/geclipse/demo_apps/JSDL_Param/, last accessed August 2009
24. Eclipse Foundation, Eclipse Public License, version 1.0, http://www.eclipse.org/legal/epl-v10.html, last accessed August 2009
25. A. Chepstov, R. Keller, R. Pugliese, M. Prica, A. Del Linz, M. Plociennik, M. Lawenda, and N. Meyer. Towards deployment of remote instrumentation e-infrastructure (in the Frame of the DORII Project). Computational Methods in Science and Technology, 15(1), 65–74, 2009
26. Virtual Laboratory (VLAB) project webpage, http://vlab.psnc.pl/
27. EXPReS project website, http://www.expres-eu.org/
28. M. Okon, D. Stoklosa, R. Oerlemans, H. J. van Langevelde, D. Kaliszan, M. Lawenda, T. Rajtar, N. Meyer, and M. Stroinski. Grid integration of future arrays of broadband radio-telescopes moving towards e-VLBI. Grid enabled remote instrumentation. Springer, p. 571, 2008

The Grid as a Software Application Provider in a Synchrotron Radiation Facility

Roberto Pugliese, Milan Prica, George Kourousias, Andrea Del Linz, and Alessio Curri

Abstract In this chapter we introduce the concept of the Grid as an *Application Service Provider.* We discuss the implementation and deployment of a fully operational system in a Synchrotron Radiation Facility. The Grid is traditionally used for HTC computations and not often as a provider for Software on Demand; especially for non-Grid specific GUI based applications. As a computing paradigm, the Grid is successfully utilised by the experimental scientific community. Due to the data-parallelism of the computational problems in the general field of physical sciences and the very high data volumes in terms of storage, the Grid has been closely related to High Throughput and High Performance Computing (HTC/HPC). A substantial number of software applications in the form of scientific computing utilities that do not require HPC are used in Synchrotron labs and other similar establishments. We envisioned a common web portal where the lab user can access a defined set of pre-configured *scientific computing* applications[1] that rely on Grid resources. The pilot application has been in the SRF Elettra.[2] The model and the intermediate subsystems have been developed and suitably configured. In this chapter we propose (1) a "Software as a Service" model (SaaS) as an alternative deployment for these applications, (2) we introduce an infrastructure where our SaaS model is fully integrated to the Grid – utilising remote computation and storage resources, (3) we outline our technology layout that enables interactivity, transparent access to storage, Web Grid portals, virtual collaboration environments and remote access to SRF instrumentation.

M. Prica (✉)
Scientific Computing Group, Information Technology Department, ELETTRA Sincrotrone Trieste, Trieste, Italy
e-mail: milan.prica@elettra.trieste.it

[1]the Scientific Computing in the domain of an SRF often deals with mathematical models for the analysis of the collected data. Typical applications are those of Imaging.

[2]Elettra – Sincrotrone Trieste S.C.p.A.. URL: http://www.elettra.trieste.it

F. Davoli et al. (eds.), *Remote Instrumentation Services on the e-Infrastructure*,
DOI 10.1007/978-1-4419-5574-6_3, © Springer Science+Business Media, LLC 2011

1 The Classical Grid In An SRF

The modern High Energy Physics (HEP) establishments like the Synchrotron Radiation Facilities (SRF) have extended and specialised computing requirements. Approaches that offer High Performance Computing (HPC) and High Throughput Computing (HTC) have long been utilised in order to process, analyse, and archive the vast amount of the generated experimental data [1, 2]. Advanced computing paradigms like those of Parallel computing and Grid [3], are considered the major approach for computing in such facilities. The former, Parallel computing, is a mature solution that is well suited since the computational problems in HEP tend to be data-parallel. The latter, the Grid Computing paradigm, is less widespread but has proved its potential [4] and is widely used in major setups [5]. A major characteristic of the Grid is that it can include parallel computing infrastructures and often it blends as a superstructure for HTC, HPC and generic distributed computing. In this chapter we focus on a different and overlooked class of computational needs that may appear in Synchrotron Radiation Facilities but are not related to HTC [6].

2 Non-HTC Scientific Software For SRF Labs

Elettra and other SRFs host a number of beamlines and their corresponding end-stations and laboratories. These labs are hosting external users who perform experiments with the assistance of the scientific personnel of the facility. Due to the non-standard nature of the specialised instrumentation, the initial data analysis and processing is based on established workflows that do not necessarily require HPC. Such workflows may include the use of various software applications for tasks such as opening a custom data format, applying an instrument-specific filter, converting imaging data, calibrating, searching past data and configurations. The in-house scientists may train their hosted users to perform on their own the workflow and its sequential steps. In most of the labs, the process requires that the external users utilise an internal computer/terminal with pre-installed software or reinstall the required software in their personal computers if the licensing allows. This way of software deployment may lead to a series of inconveniences including installation problems, incompatibilities, variations in software configurations, unexpected processing times due to diversity of hardware, versioning issues, data redundancy, and other.

3 Software As A Service, On Grid

Our solution suggests the consideration of an alternative software deployment method, that of Software as a Service (SaaS) [7, 8] that can be viewed in its general form as On-demand software. Additionally we suggest the utilisation of the existing Grid infrastructure to act as the required Application Service Provider (ASP). Such a Vertical Market [9] solution should enable any authorised beamline user

to access the collection of software applications necessary for the lab's workflow and use them in a unified Web-based environment. The SaaS system we propose is suitably blended in the Grid so the computation, storage and sharing of the data is always remote. This yields the obvious advantage of utilising remote and potentially more powerful hardware than that of the user's personal computer. During the design phase the HTC orientation of the current Grid infrastructure was critically examined. We concluded that there are many software applications in the SRF community that do not need HTC but could still benefit from the established Grid infrastructures. This traditional infrastructure has been deployed mainly for the computation of large batch calculations [3] and network data I/O. In our case we had to enable a user friendly SaaS solution without altering the HTC character of the existing systems. In order to enable the solution we propose: (i) a Grid Web Portal for the user, administrator, and developer, (ii) further development of the Interactive Grid, (iii) suitable extensions of the Middleware to incorporate SRF needs.

4 A Web Portal for the Grid services

As a Grid Web portal, we have developed the Virtual Control Room (VCR) [10]. Its aim is to provide the user with access to Grid resources and services through a single login from a Web browser. The VCR allows profile and role based user accounts thus one can login as beamline personnel, external user, administrator, or developer. There is an ongoing development of additional features including functionalities such as on-line collaboration and remote instrument control. The core development of the VCR is on the server-side of the system where the actions on the user-friendly web interface are suitably forwarded to the Grid middleware. In the case we propose, an external user logs-in the VCR and based on the user's profile, a predefined set of software applications may be launched. This software is not necessarily web-based, but *web-delivered* since the computation is on a remote computing resource and the VCR provides the required technologies that enable user interaction through a web browser.

5 Interactivity, Data Storage and the VCR

The general notion of interactivity on the Grid, although it has been recently progressed and advanced by successful large projects [11–13], is still a relatively new area. For our solution we aimed at enabling standard applications that assume interaction with the user. The technology we investigated is that of i2glogin. This enables an i2glogin-initiated application running on a Grid Worker Node (WN) to communicate through secure channels and interact with the server side of i2glogin that usually resides on the system that launched the application; the User Interface (gLite UI)[3]

[3]The UI [14] should not be confused with the popular term of Graphical User Interface (GUI).

that the job was submitted from. The desired results were those of remote visualisation and application steering. In our case, the user has access to the VCR and not to a traditional UI, so the server instance of i2glogin has to be integrated on the VCR server. Through suitable tunneling technologies and Desktop sharing (VNC) techniques, the VCR client in form of a web portal accessed from the user's computer allows interaction even with applications with GUI. Since the user works only with a web browser and the applications runs on a remote WN, the user can be agnostic of the application's native platform, configuration, and hardware requirements. In the process of enabling legacy desktop applications it was desired to access Storage Elements (SE) for full I/O. There are two main approaches for I/O from an application to a SE. The first is a through *synchronisation* of the WN's temporary storage space with that of a SE through scripts and tools like GridFTP [15] and UberFTP. The second approach is that of a custom designed application where the necessary mechanisms are integrated in the application so that the I/O to a SE in done through a suitable API. The problem rises in the simple case that the user needs to open or save a file by means of a legacy graphical dialog (in a GUI based application) that displays only the WN local filesystem. For solving this issue we oriented towards virtual filesystem solutions [16] that could mount locally a remote SE. Such solutions proved to be intrusive since they required an installation overhead so we adopted an alternative approach; that of Parrot [17]. Parrot enables the attachment of existing programs to remote I/O systems through the filesystem interface; in our case the remote system is GridFTP in a SE. There are no installation requirements and not classical mounting but library call intercepting so there is a trade-off in performance. The *online collaboration* [18] functions of the VCR combined with the use of a common storage in the Grid (SE) can enrich the use of the software applications since the scientists can exchange information in real-time, access common datasets, discuss configurations and redefine workflows.

6 Enabling Scientific Instruments on the Grid

An advancement especially for SRF facilities, is the enhancement of the current Grid middleware with a suitable system that enables the inclusion of scientific instrumentation and devices such as robots, sensors, cameras, motors and other instruments. The concept of this grid middleware extension is the Instrument Element (IE). We introduce an evolved and advanced version of Instrument Element for remote operation and control of instruments[19–21]. It is an addition to the existing core Grid parts; the Computing Element (CE) and the Storage Element (SE) that serve the purposes that their names suggest. The IE that we are referring to, and the related technologies, have been developed in the EU project on the Deployment of Remote Instrumentation Infrastructure (DORII).[4] In the vertical market of Synchrotron

[4]The DORII project is supported by the European Commission within the 7th Framework Programme (FP7/2007–2013) under grant agreement no. RI-213110. URL: http://www.dorii.eu

Radiation Facilities "grid-enabling" scientific devices has a very high impact. The scientific instrumentation that has been Grid-enabled can be remotely accessed and steered by authorised scientists that may be off-site. By providing a solution that enables remote instrument control through the common web portal of the VCR, the SaaS deployed application can organically connect to the source of the data. In practical terms: through the VCR the user can securely control the instrument, virtually collaborate with other scientists, collect the data, access them from distributed storage, use a set of preconfigured applications and process the data in remote computing infrastructures on the Grid.

7 Conclusions

The Grid, by principle, should be flexible, extensible and scale well. The HEP community has utilised the characteristics of its infrastructure for hight throughput computational problems. We extend this functionality by deploying a total infrastructure for enabling software on-demand services on the Grid. The SaaS class we target is that on scientific computing tasks and especially that of non-HTC software for SRF beamlines and labs. We aim at an improved end user experience and this has resulted in a common interaction environment, the VCR, with numerous services and features. Additional extensions, like the IE, on the Grid middleware have bridged the gap between software, the Grid and specialised scientific instrumentation. The pilot deployment was for the Small Angle X-ray Scattering (SAXS) beamline[5] in the SRF Elettra. There is work in progress for the further development of the above concepts in order to enable a wider range of applications and use scenarios. Areas of technical interest include that of performance issues, data visualisation, optimisations and middleware advances. We consider the Grid a viable computing paradigm for handling the computational needs of Synchrotron Radiation Facility and we believe that the introduction of additional services like SaaS can be beneficial, assuming that does not interfere with the current uses of it.

Acknowledgements We thank our partners in the DORII project for the ongoing exchange of scientific ideas. Many thanks to Herbert Rosmanith (i2glogin) and Douglas Thain (Parrot). Finally we thank the two unknown referees for their helpful comments.

This works has been partially supported by the DORII EU project (European Commission within the 7th Framework Programme (FP7/2007-2013) under grant agreement no. RI-213110).

[5] A high-flux Small Angle X-ray Scattering (SAXS) beamline mainly intended for time-resolved studies on fast structural transitions in the sub-millisecond time region in solutions and partly ordered systems with a SAXS-resolution of 1–140 nm in real-space. URL:http://www.elettra.trieste.it/experiments/beamlines/saxs/index.html

References

1. J. Basney, M. Livny, and P. Mazzanti. Harnessing the capacity of computational grids for high energy physics. Conference on computing in high energy and nuclear physics, 2000
2. D.P. Benjamin. Grid computing in the Collider Detector at Fermilab (CDF) scientific experiment. Arxiv preprint arXiv, 0810.3453, 2008
3. I. Foster. The grid: A new infrastructure for 21st century science, Physics Today, 55, 42–47, 2002
4. I. Foster and C. Kesselman. Computational grids. Cern European Organization for Nuclear Research-Reports-Cern, 1998, pp. 87–114
5. A. Chervenak, I. Foster, C. Kesselman, C. Salisbury, and S. Tuecke. The data grid: Towards an architecture for the distributed management and analysis of large scientific datasets. Journal of Network and Computer Applications, 23, 187–200, 2000
6. C. Blanchet, R. Mollon, D. Thain, and G. Deléage. Grid deployment of legacy bioinformatics applications with transparent data access. Gene, 153, 224,000
7. M. Turner, D. Budgen, and P. Brereton. Turning software into a service. Computer, 38–44, 2003
8. M.P. Papazoglou. Service-oriented computing: Concepts, characteristics and directions. Proceedings of the Fourth International Conference on Web Information Systems Engineering, Washington: IEEE Computer Society Press, December, 2003
9. R.A. Belderbos and L. Sleuwaegen. Local content requirements and vertical market structure. European Journal of Political Economy, 13, 101–119, 1997
10. R. Ranon, L. De Marco, A. Senerchia, S. Gabrielli, L. Chittaro, R. Pugliese, L. Del Cano, F. Asnicar, and M. Prica. A web-based tool for collaborative access to scientific instruments in cyberinfrastructures. In Grid enabled remote instrumentation. Signals and communication technology, F. Davoli, N. Meyer, R. Pugliese, S. Zappatore, Eds., Springer, New York, NY, pp. 237–251, 2009, http://dx.doi.org/10.1007/978-0-387-09663-6_16.
11. B. Coghlan, E. Fernández, E. Heymann, P. Heinzlreiter, S. Kenny, M. Owsiak, I. C. Plasencia, M. Plóiennik, H. Rosmanith, M. A. Senar, S. Stork, and R. Valles. Int.eu.grid project approach on supporting interactive applications in the grid environment. In Grid enabled remote instrumentation. Signals and communication technology, F. Davoli, N. Meyer, R. Pugliese, S. Zappatore, Eds., Springer, New York, NY, pp. 435–445, 2009, http://dx.doi.org/10.1007/978-0-387-09663-6_28.
12. I. Marco. The interactive european grid: Project objectives and achievements. Computing and Informatics, 27, 161, 2008
13. S. Basu, V. Talwar, B. Agarwalla, and R. Kumar. Interactive grid architecture for application service providers. Proceedings of the International Conference on Web Services (ICWS), 2003
14. E. Laure, S.M. Fisher, A. Frohner, C. Grandi, P. Kunszt, A. Krenek, O. Mulmo, F. Pacini, F. Prelz, and J. White. Programming the grid with gLite. Computational Methods in Science and Technology, 12, 33–45, 2006
15. W. Allcock, J. Bester, J. Bresnahan, A. Chervenak, L. Liming, and S. Tuecke. GridFTP: Protocol extensions to FTP for the grid. Global Grid ForumGFD-RP, 20, 2003
16. L. Skital, R. Slota, D. Nikolov, and J. Kitowski. Grid enabled virtual storage system using service oriented architecture. Proceedings of the 2005 Conference on Software Engineering: Evolution and Emerging Technologies, IOS Press, Amsterdam, pp. 149–159, 2005.
17. D. Thain and M. Livny. Parrot: An Application Environment for Data-Intensive Computing. Scalable Computing: Practice and Experience, 6, 9–18, 2005
18. Wilson, P. (1991). Computer supported cooperative work: An introduction. Kluwer, Dordrecht
19. R. Pugliese, M. Prica, G. Kourousias, A. Del Linz, and A. Curri. Integrating instruments in the grid for on-line and off-line processing in a synchrotron radiation facility. Computational Methods in Science and Technology, 15(1), 21–30, 2009.

20. M. Prica, R. Pugliese, A. D. Linz, and A. Curri. Adapting the instrument element to support a remote instrumentation infrastructure. Remote Instrumentation and Virtual Laboratories, 11–22, 2010.
21. E. Frizziero, M. Gulmini, F. Lelli, G. Maron, A. Oh, S. Orlando, A. Petrucci, S. Squizzato, and S. Traldi. Instrument element: A new grid component that enables the control of remote instrumentation. Proceedings of the Sixth IEEE International Symposium on Cluster Computing and the Grid (CCGRID06)-Volume 00, IEEE Computer Society Washington, DC, 2006

Part II
Support of Grid Functionalities

An Optimized Architecture for Supporting Data Streaming in Interactive Grids

L. Caviglione, C. Cervellera, and R. Marcialis

Abstract The access to instrumentation and support of real-time requirements were already envisaged, in terms of time-sensitive control of devices and remote communications, in the original grid vision. Nevertheless, the integration of devices concretely rises the problem of supporting such operations with proper network requirements. On the other hand, as an evolution of peer-to-peer (p2p) file-sharing applications, overlay-based networks can be utilized to exploit application level strategies to overcome limitations imposed by the underlying network infrastructure, e.g., the lack of multicast support.

In this perspective, the chapter introduces an overlay Content Distribution Network (CDN) able to sustain the real-time delivery of data streams to support interactive Grid applications. To better use resources the control and optimization of the CDN are performed through a model predictive control scheme. Simulations are provided to show the effectiveness of the proposed solution.

1 Introduction

The access to instrumentation and support of real-time requirements were already envisaged, in terms of time-sensitive control of devices and remote communications, in the original grid vision [1–3]. Nevertheless, recent advancements in the integration of devices, both real [4] and virtual, within middleware components [5] and the exploitation of standardized signaling solutions [6], concretely rise the problem of supporting such operations with proper network requirements. At the same time, this makes such vision feasible.

Summing up, real-time access and control of grids require a non-trivial support of Quality of Service (QoS) from the underlying network architecture, and a careful planning to avoid the under/over provisioning of resources while delivering a service. Therefore, one of the basic building blocks of this "enriched" grid vision

L. Caviglione (✉)
Institute of Intelligent Systems for Automation (ISSIA), Via de Marini 6, 16149 Genova, Italy
e-mail: luca.caviglione@ge.issia.cnr.it

F. Davoli et al. (eds.), *Remote Instrumentation Services on the e-Infrastructure*,
DOI 10.1007/978-1-4419-5574-6_4,

needs sophisticated and strict QoS guarantees. However, being a highly distributed framework, the grid infrastructure often clashes with the even more heterogenous network deployments.

Some practical issues prevent to guarantee QoS for such applications, specifically:

- the grid could span over different Autonomous Systems (AS) or ISPs, having non-uniform Service Level Agreements (SLAs), thus impeding a global traffic engineering strategy (e.g., by using standard signaling solutions such as the MPLS-TE or RSVP-TE);
- the underlying infrastructure (or well-defined "isles") could not support QoS, being straight best-effort technologies;
- grid nodes are often deployed at the border of the network, thus preventing intervention on capacity allocations within the data bearer (e.g., adjusting RSVP-based reservation disciplines);
- the heterogeneity of the infrastructure in terms of transmission technology (e.g., mixed wired and wireless communications) accounts for difficulties in exploiting traffic management in a unified manner;
- for security or management reasons, a higher management layer could be required, e.g., in order to uniform the access to resources, or to provide a unified Application Program Interface (API) or infrastructure.

To develop an efficient solution allowing to cope with the aforementioned issues, we identify in peer-to-peer (p2p) or, more generally, overlay networks, the proper enabling technologies. In fact, p2p file-sharing services have been proven to support millions of users, thus producing an impressive amount of traffic in a scalable way [7]. Besides, the increasing popularity of Voice over IP (VOIP) architectures, relying upon p2p to page users, demonstrates that such technology could be adopted also for the management of sessions in a real-time manner. Therefore, the next step is to use p2p also to empower the data streaming in interactive grids.

Today there are many p2p applications already deployed for streaming data in the Internet, even if aimed at supporting the delivery of multimedia contents, rather than "raw" data. Among others, we cite *Zattoo*, *PPStream* and *GridMedia* [8]. Obviously, the real-time constraints enforced by the need of streaming contents impose to implement protocols and algorithms different from those employed for file-sharing operations. The most popular ones are called *pull-based* streaming protocols [8], where each peer independently selects neighbors creating an *unstructured* overlay. Even if such protocols are very simple [9], further methods relying on single distribution trees also exist. Among others, we mention optimization problems concerning the creation of multiple (i.e., one for a given group of flows) spanning trees [10].

A recent trend is to rely on tracker-based solutions, such as BitTorrent (http://www.bittorrent.com). The presence of a centralized component, and its clear separation from the distributed portion of the architecture (i.e., the p2p swarm), offer

some major benefits: performing the optimization of the overall service is simpler, and it is possible to change service-policies quickly.

As regards selection strategies of clients (in an architecture relying on overlays), reference [11] deals with optimally partitioning the clients into disjoint subsets according to different criteria, such as network resource usage and the degree of congestion. Besides, optimal rate allocation in an overlay-based Content Distribution Network (CDN) for efficient utilization of limited bandwidths has been investigated in [12]. Lastly, more formal works solving capacity allocation problems in CDNs, resource allocation and object placements can be found in [13, 14], respectively.

In this perspective, this chapter introduces a novel architecture employing optimization to build an overlay among nodes composing the grid to guarantee the real-time delivery of data streams. Relying upon an overlay juxtaposed (or at least, co-located) over the grid plane, provides other layers with a coherent and unified networking scenario. Then, to manage the proper degree of QoS, we deploy a CDN tightly coupled with the grid nodes, in order to exploit resource allocation to assure real-time requirements to time-sensitive data flows. Even if grid infrastructures are expected to exhibit much less churn (i.e., the continuous process of nodes arrivals and departures) than the one affecting p2p overlays, we still recur to optimization to better exploit resources and to react from changes in the user population.

The remainder of the chapter is structured as follows: Section 2 introduces the reference architecture, while Section 3 discusses the mathematical model and its optimization. Section 4 deals with numerical results collected via an ad-hoc software simulator and Section 5 concludes the paper and showcases future developments.

2 System Architecture

The aim of the proposed architecture is to deliver the data flow produced by an instrument in a real-time fashion to a given set of grid nodes and end-users, in such a way that the use of available resources is optimized. All the involved components are connected through an IP network, i.e., the Internet. The reference scenario is depicted in Fig. 1, where multiple domains with different QOS requirements could be "unified" by exploiting the isolation properties of the overlay acting as a CDN.

The content provider is also responsible of running the centralized component exploiting optimization, which could be co-located within a CDN node.

For the sake of simplicity, we assume CDN and Grid node co-located, thus they can be merged into a single entity. Concerning the architectural components proposed in Fig. 1, they are:

- Grid Node (GN): it is responsible both of forwarding the stream through the distribution overlay tree, which is composed by the interconnection of different GNs (depicted in Fig. 1 with bold lines), and of serving end users. We point out that a GN could be deployed for: (i) providing an entry point for the source; (ii) solely forwarding the stream to remote GNs (i.e., it is a simple data transit point); (iii) serving end users and (iv) a mix of the previous points.

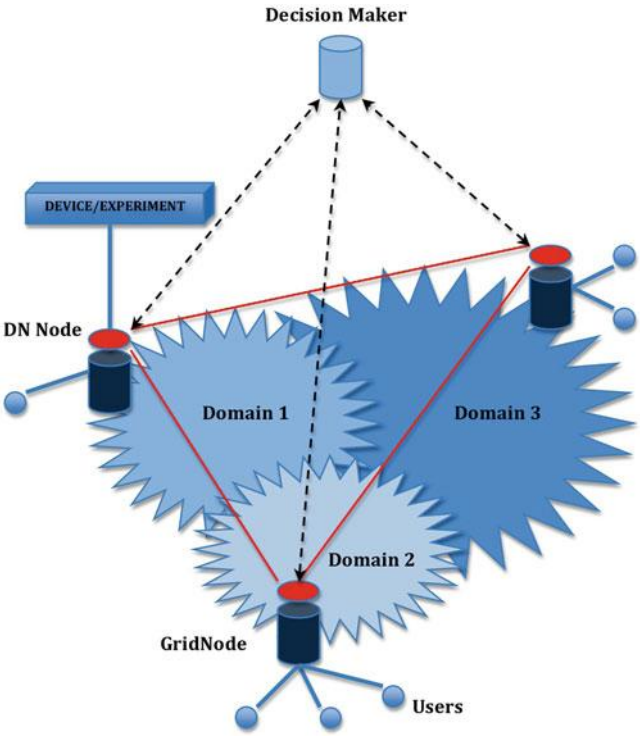


Fig. 1 The overall system architecture and a reference scenario

- Decision Maker: it is the tracker-like entity, that could be merged within a GN, in charge of optimizing the usage of resources, by exploiting different performance costs. For instance, it can be adopted to maximize the allotted user population. It exchanges status information with GNs and issues back to them the optimized strategies to be executed.
- Users: they consume the stream on their host or appliances. They could be collaborative environments adopted for displaying results, or hosts running services for data processing purposes. We point out the importance of the real-time feature, since we suppose that the produced data flow has significance only if received in a well-defined moment in time. For instance, this could be the case of an instrument sending measurements of a phenomenon that have to be processed "live" while observing impacts over the surrounding environment.
- Source: it provides the data for the stream. This could be also the resulting composition of different instruments (e.g., a synthetic aperture radar) or the final step of a complex measurement chain. From an architectural viewpoint, this is perceived as a monolithic entity. As an example, in Fig. 1 it is depicted as a device or an experiment.

3 Description of the Model

In order to optimize the delivery of the stream, we describe the evolution of the system by means of a discrete-time dynamic model. First, we provide basic definitions for a better understanding of the following sections.

Define T as the number of temporal stages corresponding to the length of the stream, sampled at time intervals of length Δt. Informally, at a given time t we call *stream front* of the source, of a GN or of the users attached to a GN the overall amount of data sent by the source, stored by a node or read by the users, respectively. Then, define $d(t)$ as the stream front of the source at temporal stage t, and B_d as the rate at which the data is streamed. As to the GNs, let M be the number of GNs and call $q_i(t)$ the stream front of node i, for $i = 1, \ldots, M$. The stream front of all the users attached to GN i at stage t, for $i = 1, \ldots, M$, is denoted by $p_i(t)$.

As to the bandwidths of the various GNs, define B_i^{tot} as the total bandwidth available to node i, for $i = 1, \ldots, M$, for the streaming process. Then, let S_i be the number of downstream GNs connected to the GN i (i.e., the number of GNs that receive the stream from the GN i), and define $B_{i,j}^{\text{out}}(t)$, for $i = 1, \ldots, M$ and $j = 1, \ldots, S_i$, as the bandwidth devoted by node i to upload data to GN j during temporal stage t.

Then, define $B_i^{\text{in}}(t)$ as the bandwidth devoted by GN i, for $i = 1, \ldots, M$, to download data from its upstream GN during temporal stage t. We assume that $B_1^{in} = B_d$ by default.

Finally, call $B_i^s(t)$ the bandwidth devoted by GN i, for $i = 1, \ldots, M$, to seed the stream to users at stage t. We always assume this bandwidth as the share that remains available to the GN after the upload and download bandwidths have been assigned, i.e.,

$$B_i^s(t) = B_i^{\text{tot}} - B_i^{\text{in}}(t) - \sum_{j=1}^{S_i} B_{i,j}^{\text{out}}(t) \tag{1}$$

3.1 State Equations

The state equation for the source stream front simply reflects its advancement through the time horizon, at the constant rate B_d

$$d(t+1) = d(t) + B_{\text{d}}\Delta(t)$$

Concerning the overall data $q_i(t)$ buffered in the GNs within the time interval [0,t], we have, for $i = 1, \ldots, M$

$$q_i(t+1) = q_i(t) + \min\{B_{j^*,i}^{\text{out}}(t), B_i^{\text{in}}(t)\}\Delta(t)$$

where $B_{j^*,i}^{\text{out}}(t)$ is the bandwidth share for the "upstreaming" GN j^* that transfers data to GN i. For the sake of simplicity, we assume that the buffering delays are negligible.

The min operator in the equation takes into account possible bottlenecks given by the bandwidth share $B_i^{\text{in}}(t)$ devoted by GN i to the upstream flow and $B_{j^*,i}^{\text{out}}(t)$.

Notice that buffers in the GNs are filled always with the most recent data available in the connected upstream GNs, thus data downloaded at a smaller rate than B_d may correspond to a "corrupt" stream when the users' front approaches that portion of the buffer. For example: (i) in the case of a video stream, users perceive a poor audio/visual quality due to missing encoded data blocks, or freezes in the playback; (ii) in the case of interactive grids data measurements could be incomplete resulting in additional "noise".

As to the evolution of the end users' reading front $p_i(t)$ at stage t for GN i, $i = 1, \ldots, M$, we have that the front can advance at the correct rate B_d only if node i has bufferized enough data; otherwise it stops at the level $q_i(t+1)$, minus a fixed quantity ϵ_L that can be considered as a "guard" useful to (i) stop early buffer underruns and (ii) avoid client interfaces to fetch data synchronously with those provided by the GN, thus preventing the buffer from being populated again.

Therefore, we can write the state equation as follows

$$p_i(t+1) = \min\{p_i(t) + B_{\text{d}}\Delta(t), q_i(t+1) - \epsilon_L\}$$

To clarify the concept, Fig. 2 depicts the two possible situations mentioned above.

Concerning the initial conditions, we assume the system starts with all the GNs having buffered a portion of the stream of length L, available to the users at the beginning of the horizon, i.e., $p_i(0) = 0$, $d(0) = L$ and $q_i(0) = L$.

3.2 End Users' Evolution and Management

The evolution of the number of end users served by GN i, for $i = 1, \ldots, M$, follows its own dynamic system, controlled by the bandwidth share devoted to them and the offered quality of the stream. The optimization of these parameters reflects in a proper Call Admission Control (CAC) mechanism deployed within the GN.

Specifically, denote by $N_i(t)$ the number of end users attached to GN i at the beginning of stage t, and by $n_i(t)$ the number of end users waiting in queue to be connected to GN i.

Also denote by $N_i^{\text{out}}(t) \in [0, N_i(t)]$ the number of users that willingly disconnect from the GN at the beginning of stage t, leaving the system permanently (i.e., they do not feed the queue $n_i(t)$).

Recall that $B_i^{\text{s}}(t)$ is the bandwidth share devoted by GN i to serve the end users, and assume that the GN can encode the offered stream at different levels of quality corresponding to $\alpha_i(t)B_d$, being $\alpha_i(t) \in [\alpha_{\min}, 1]$ a proper control that we need to decide at every stage. As regards the possible levels of quality, they can, for instance, reflect the following situations: (i) by "pruning" data collected from sensors involved in the experiment or (ii) by reducing the resolution of the data collected by the measurement instrumentation.

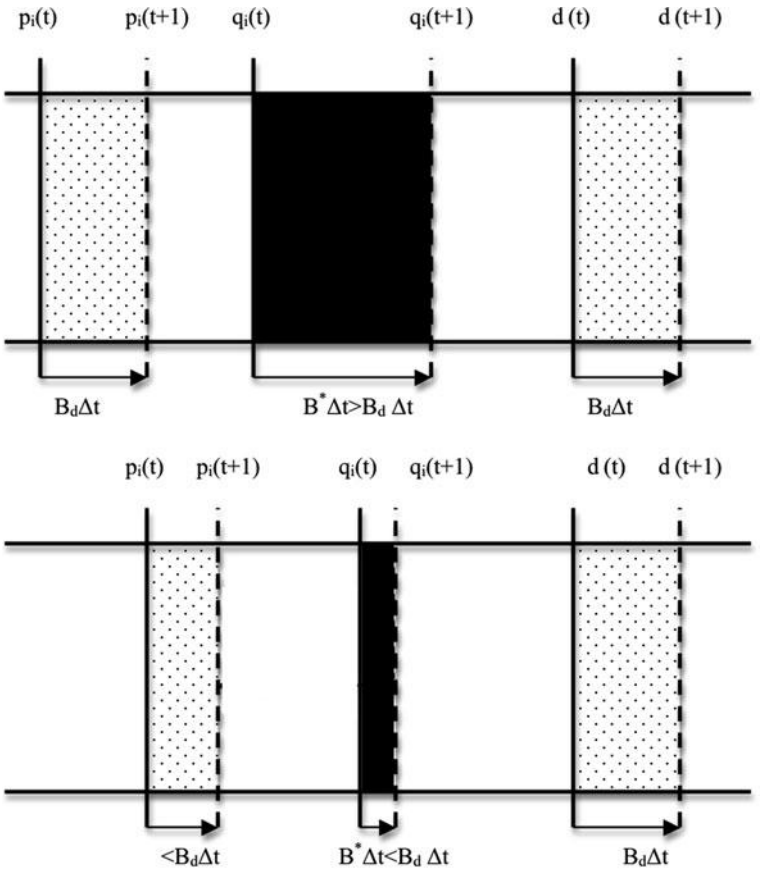


Fig. 2 Two possible cases of utilization of the internal buffer of GNs acting as CDN distribution nodes. *Upper*: the buffer is properly filled with new data, according to the bit-rate of the stream to be delivered. *Lower*: due to insufficient resources, the buffer is approaching an underrun situation

Thus, the maximum number of end users that can be served by GN i at stage t with quality $\alpha_i(t)B_d$ is given by $\left\lfloor \frac{B_i^s(t)}{\alpha_i(t)B_d} \right\rfloor$.

Notice that the end users' front p_i still moves at the constant rate B_d (if $q_i(t)$ allows it) even if $\alpha_i(t) < 1$. In fact, values of $\alpha_i(t) < 1$ merely reflect in a reduction of quality for the user, not affecting that actual timing of the stream. We point out that the level of quality of the stream can be changed only when delivering it to end users. Conversely, the stream exchanged among different GNs cannot be further encoded, as to prevent the perceived quality from quickly decreasing.

Now, we can write the number of end users connected to GN i at stage t, as follows

$$N_i(t) = \min \left\{ \left\lfloor \frac{B_i^s(t)}{\alpha_i(t)B_d} \right\rfloor, N_i(t-1) - N_i^{\text{out}}(t) + n(t) \right\}$$

initialized by $N(-1) = 0$. The min operator reflects the fact that the maximum number of users can access the stream only if the queue is sufficiently long.

Define $\Delta N_i(t) = N_i(t) - N_i(t-1) + N_i^{\text{out}}(t)$ as the variation in the number of connected users at stage t, not counting those that willingly disconnected. Thus, a positive value of ΔN_i means that the CAC lets new users connect, whereas a negative ΔN_i means that the CAC moves users into the the waiting queue. We point out that users have in their client interfaces a local buffer allowing to absorbe brief disconnections.

Then, the evolution of the waiting queue is represented by the following state equation

$$n_i(t+1) = \max\{0, n_i(t) - \Delta N_i(t) + n_i^+(t) - n_i^-(t)\}$$

where $n_i^+(t)$ and $n_i^-(t) \in [0, n_i(t)]$ are users that willingly join and leave the waiting queue at stage t, respectively.

In the present work, the quantities $N_i^{\text{out}}(t)$, $n_i^+(t)$ and $n_i^-(t)$ are modeled as random variables with Poisson distribution.

3.3 Performance Cost

For the purposes of controlling the stream delivery process and optimizing the performance of the system, we define at each stage a cost that is a function of the state, the controls and the random vectors.

Define the state vector as

$$x(t) = [d(t), q_1(t), \ldots, q_M(t), p_1(t), \ldots, p_M(t)]$$

the control vector as (recall that $B_1^{\text{in}}(t) = B_d$ by default)

$$\begin{aligned} u(t) = [&B_2^{\text{in}}(t), \ldots, B_M^{\text{in}}(t), B_{1,1}^{\text{out}}(t), \ldots, B_{1,S_1}^{\text{out}}(t), \\ &B_{M,1}^{\text{out}}(t), \ldots, B_{M,S_M}^{\text{out}}(t), \alpha_1(t), \ldots, \alpha_M(t)] \end{aligned}$$

and the vector of random quantities as

$$\begin{aligned} \xi(t) = [&N_1^{\text{out}}(t), \ldots, N_M^{\text{out}}(t), n_1^+(t), \ldots, n_M^+(t), \\ &n_1^-(t), \ldots, n_M^-(t)] \end{aligned}$$

Notice that we do not include the bandwidths $B_i^s(t)$ in the control vector, in order to reduce the dimensionality of the optimization problem, as we can obtain them by (1), as said.

Then, we use a cost that takes into account the requirements of (i) having the stream propagate through GNS without delays, so that it can be delivered on-time to the users without interruptions or corruption, and (ii) allowing the connection to as many users as possible with the highest possible quality.

Specifically, we employ a cost of this form

$$\begin{aligned} h[x(t), u(t), \xi(t)] = & \sum_{i=1}^{M} \beta_i |d(t+1) - q_i(t+1)| \\ & + \gamma_i e^{-|q_i(t+1) - p_i(t+1) - \epsilon_L|/\sigma} \\ & + \delta_i g(\alpha_i(t)) + \psi_i r(N_i(t)) \end{aligned}$$

where $\beta_i, \gamma_i, \delta_i, \psi_i \in \mathbb{R}^+$ for $i = 1, \ldots, M$ are weight coefficients, $\sigma \in \mathbb{R}^+$ and g and r are decreasing functions. The first term in the cost penalizes the distance between the received amount of the streaming at a given GN from the "live" streaming front (i.e., it reflects how the stream propagates through the overlay network), while the second term is high when the users' front gets too close to the guard ϵ_L (i.e., users are reaching the end of the buffered data available in the respective GN, and this results in an interruption of the stream). Notice that this term may be dropped and replaced by a proper constraint. The third and fourth terms are high when there are few users connected to the GN and the offered quality of the stream is low, respectively. Notice that the four terms correspond to "competing tasks", therefore the aim of optimization is to balance the resources and assign the best amount of bandwidth shares (together with the best coding of the stream) to the above mentioned tasks.

3.4 Constraints

In order to state the optimization problem, we need to impose constraints on the system and the controls. Specifically, we have the following

- positivity of the bandwidths

$$\begin{aligned} B_i^{\text{in}}(t) &\geq 0 \quad \text{for } i = 2, \ldots, M \\ B_{i,j}^{\text{out}}(t) &\geq 0 \quad \text{for } i = 1, \ldots, M \text{ and } j = 1, \ldots, S_i \end{aligned}$$

- the sum of the bandwidths for the single node must not exceed the overall GN bandwidth capacity

$$B_i^{\text{in}}(t) + \sum_{j=1}^{S_i} B_{i,j}^{\text{out}}(t) \leq B_i^{\text{tot}} \quad \text{for } i = 1, \ldots, M$$

- boundedness of the stream quality parameter

$$\alpha_{\min} \leq \alpha_i(t) \leq 1 \quad \text{for } i = 1, \ldots, M$$

- the new buffer front for a GN at the end of stage t must not exceed the buffer available at the "upstreaming" GN (i.e., the amount of data that can be delivered from the received live stream)

$$q_i(t+1) \leq q_{j^*}(t+1)$$

where j^* is the "upstream" GN that transfers data to GN i.

3.5 Optimization

The aim of the decision-maker is to control the evolution of the streaming process optimizing the bandwidth resources dynamically, until the end of the stream.

Then we want to minimize a total mean cost averaged over Q realizations of the random vectors, having this form

$$\min_{u(0),\ldots,u(T-1)} \sum_{t=0}^{T-1} \sum_{q=1}^{Q} \frac{1}{Q} h[x(t), u(t), \xi^q(t)]$$

subject to the state equation and the constraints introduced above, where $\xi^q(t)$ is a realization of the random vector $\xi(t)$ drawn according to its probability distribution.

Ideally, dynamic programming techniques could be used to devise off-line optimal strategies (see, e.g., [15, 16]). Yet, the computation of the optimal control vectors for the whole horizon would be unfeasible in a real-time context, thus, in order to keep computational times manageable, we adopt a model predictive control scheme that minimizes the cost over K stages (see, e.g., [17]).

In particular, we want to solve this problem: find

$$\arg \min_{u(t),\ldots,u(t+K)} = \sum_{k=0}^{K} \sum_{q=1}^{Q} \frac{1}{Q} h[x(t+k), u(t+k), \xi^q(t+k)]$$

subject to the state equation and the constraints introduced above.

Then, only the first control action $u^*(t)$ is applied to move the system from state $x(t)$ to state $x(t+1)$, according to the actual real-time value of the random vector, and then the optimization procedure is repeated at stage $t+1$ to obtain $u^*(t+1)$, and so on.

Notice that the resulting control is in closed-loop form, i.e., it depends on the current state $x(t)$.

Notice also that the cost $\sum_{k=0}^{K} \sum_{q=1}^{Q} \frac{1}{Q} h[x(t+k), u(t+k), \xi^q(t+k)]$ for a given sequence $u(t), \ldots, u(t+K)$ and a given initial point $x(t)$ can be computed online by means of simulation.

The minimization over $u(t), \ldots, u(t+K)$ can be performed through standard nonlinear programming solvers.

In particular, for the performance tests presented in the next section we employed a Monte Carlo procedure that is based on the exploration of the control space by sequences of G i.i.d. points extracted with uniform probability. The point corresponding to the lowest cost is taken as the solution. The method can be refined through clustering techniques (see, e.g., [18]), and by employing quasi-Monte Carlo sequences instead of i.i.d. sequences, for a faster convergence [19].

4 Performance Evaluation

In order to prove the effectiveness of the proposed solution, we carried out performance evaluation via an ad-hoc software simulator. As a test scenario, we selected the diffusion of a real-time data stream produced by a measurement. In particular, we employed a source producing a data stream at a constant bit-rate of 10 Mbit/s, i.e., B_d= 10 Mbit/s. The duration of the test has been set equal to 2 h, as to reflect a quite long experimental analysis. This corresponds to $T = 1440$ stages with a sampling time $\Delta T = 5$ s. The values of $\alpha_i(t)$ (defined in Section 3.2) have been set to reflect a scalable profile, where data is encoded at the 100, 75 or 50% of the original rate. The buffer length L has been selected taking into account IETF recommendations on transmitting real-time multimedia streams [20]; thus it has been selected in order to store about 10 s of playback. The guard ϵ_L is equal to $L/10$. We point out that even if we transmit data produced by experiments, instead of multimedia traffic, such rule is still reasonable to avoid buffer underruns.

As regards the overlay topology of GNs, we rely on a Virtual Organization (VO) depicted in Fig. 3, and we considered $M = 7$ GNs equipped with OC-1 or OC-3 links, which correspond to 155.52 Mbit/s and 51.84 Mbit/s, respectively, reflecting in the values of the corresponding B_i^{tot} for the various GNs. Specifically, GN 1 – 6 are equipped with OC-3 links, while GN 7 has OC-1 connectivity. Additionally, we assume such links to be dedicated (i.e., there are no other competing traffic flows) and without outage. End users are supposed not to be the bottlenecks, always having enough bandwidth to receive the stream. We also assume the needed signaling to feed the DN and to issue back controls as negligible. Therefore, the DN has not been considered as a "physical" entity.

Concerning the cost, the functions g and r have been taken as $g(z) = r(z) = -z$, while σ has been taken equal to 1000.

The churn affecting users has been modeled by using a Poisson process, with mean selected in such a way that n_i^+ equals averagely 20% of the user population attached to a given GN at each time t, i.e., $N_i(t)$. Similarly, the mean for n_i^- has been chosen in such a way that the value equals averagely 10% of the population, and the same holds for $N_i^{\text{out}}(t)$ as well. Therefore, we assumed the grid environment as less affected by churn than those characterizing standard p2p file-sharing applications.

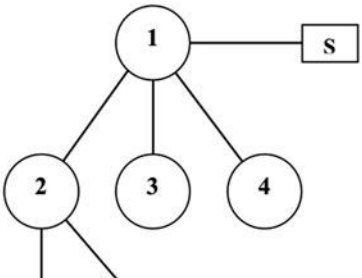


Fig. 3 The resulting overlay composed by CDN adopted for the performance evaluation. Node 1 s to 6 are equipped with OC-3 links, while node 7 has OC-1 connectivity. *S* is the source injecting data into the VO. Notice that the DN is not depicted here

Table 1 Cost function parameters

	β	γ	δ	ψ
Scenario 1	9×10^{-8}	10	5×10^{-4}	0.5
Scenario 2	2×10^{-8}	10	7.5×10^{-3}	0.5

In order to optimize and control the overall process, the decision maker employes $G = 800$ i.i.d. points to explore the control space, and the predictive control minimizes the cost over $K = 3$ stages. The number of disturbances Q has been selected equal to 10.

We present results related to two different scenarios. In the first (Scenario 1), higher priority is given to the delivery of the stream through the GNs, while in the second (Scenario 2) priority is given to the number of users attached to GNs and the quality of the offered stream.

Table 1 contains the parameters of the cost function corresponding to the two scenarios. Notice that we drop the index i since in these tests all the parameters have the same value for $i = 1, \ldots, M$.

To showcase the effectiveness of the proposed approach, Fig. 4 and 5 depict the behavior of the stream fronts of GN 2, which is the most critical (i.e., responsible of providing a transit for a large subset of the GNs), in Scenario 1 and Scenario 2, respectively. In particular, the difference between the GN front $q_2(t)$ and the users's

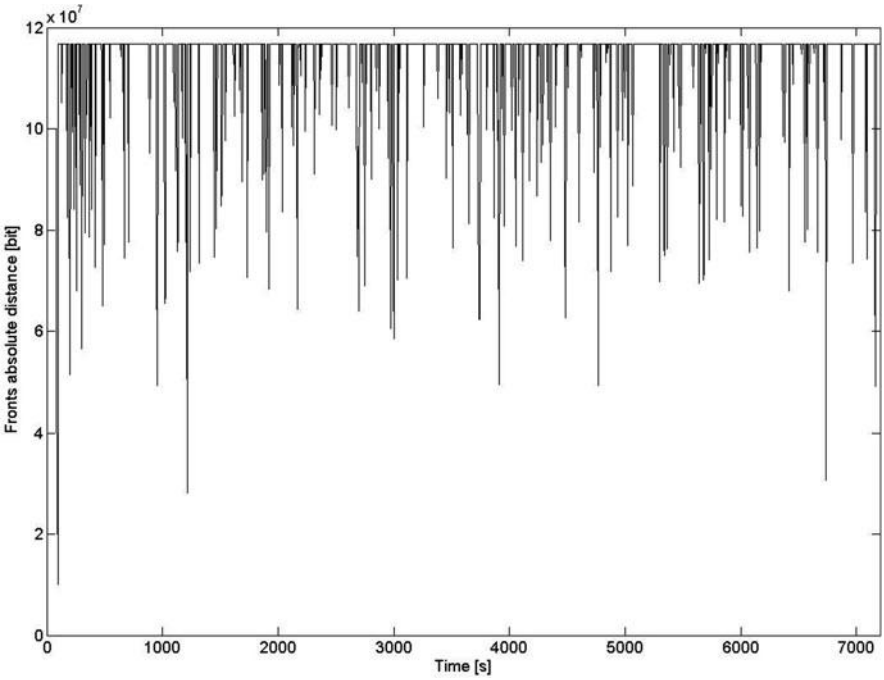


Fig. 4 Difference between the GN front and the users's front of GN 2 in Scenario 1

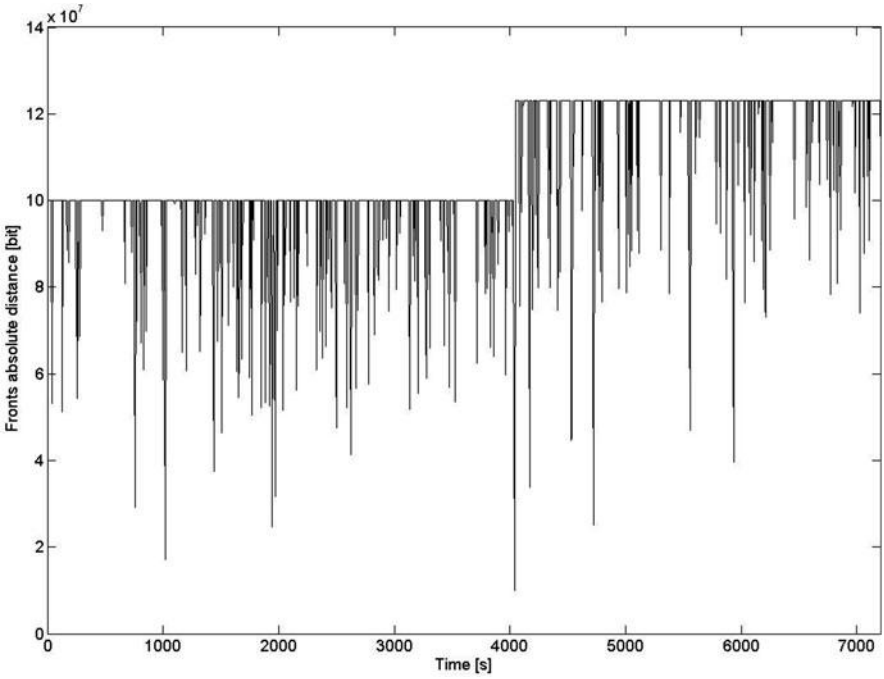


Fig. 5 Difference between the GN front and the users's front of GN 2 in Scenario 2

front $p_2(t)$ is shown. Notice that the points at which the difference equals ϵ_L correspond to the users' stream freezing. To elaborate on this point, Fig. 6 provides a close-up on such "critical" situation happening in Scenario 2 (the one around second 4040). In the figure, the evolution of the three fronts of the source, the GN and the users, respectively, are depicted. It can be seen that all the fronts proceed with the same slope when the GN receives data at rate B_d, then the rate becomes slower than B_d and users exhaust the available buffered data. After a few seconds, the GN manages to receive data at rate greater than B_d and it resyncs with the live stream produced by the source.

Figures 7 and 8 depict the evolution of the total number $\bar{N}(t)$ of users connected to the GNs at the various stages in Scenario 1 and Scenario 2, respectively, computed as

$$\bar{N}(t) = \sum_{i=1}^{M} N(t)$$

We point out that the continuous process of connection/disconnection of users shown in the figure (which depends on the CAC policies given by the decision maker) does not actually reflect in users perceiving interruptions in the stream, given client interfaces are typically endowed with suitable local buffers that can recover short periods of disconnection from the source.

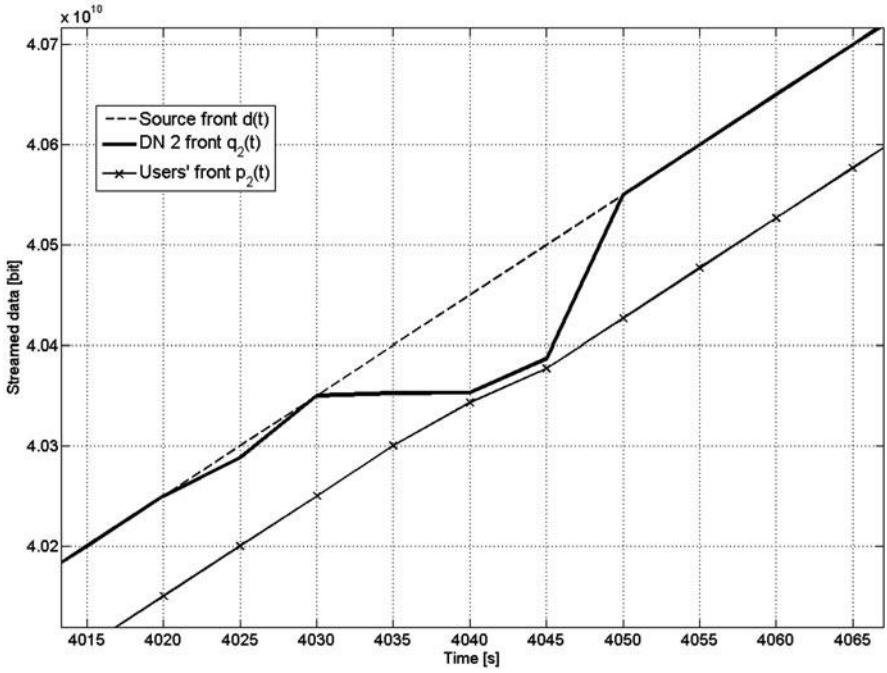


Fig. 6 Close-up on a critical situation in Scenario 2

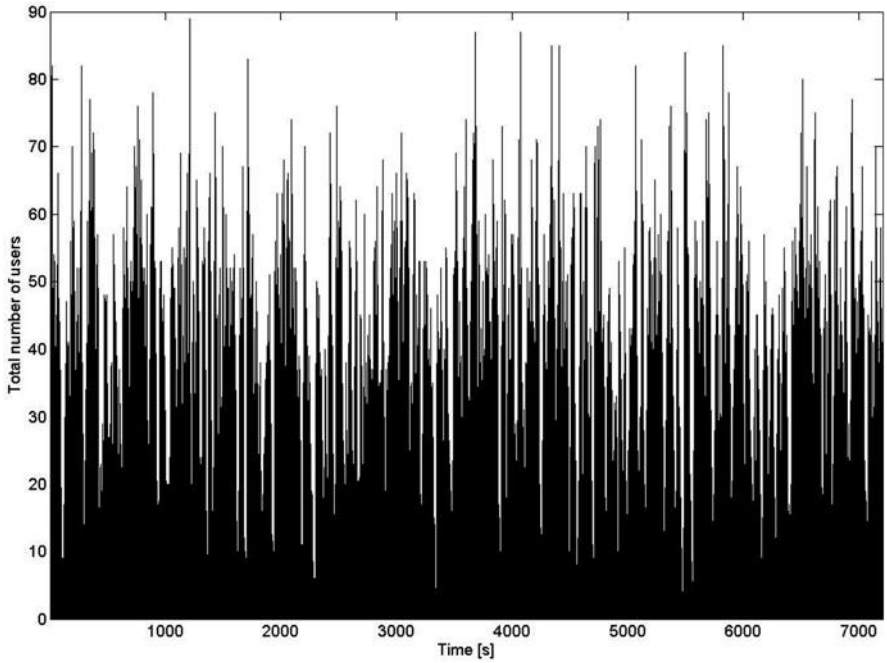


Fig. 7 Total number of users connected to the GNs in Scenario 1

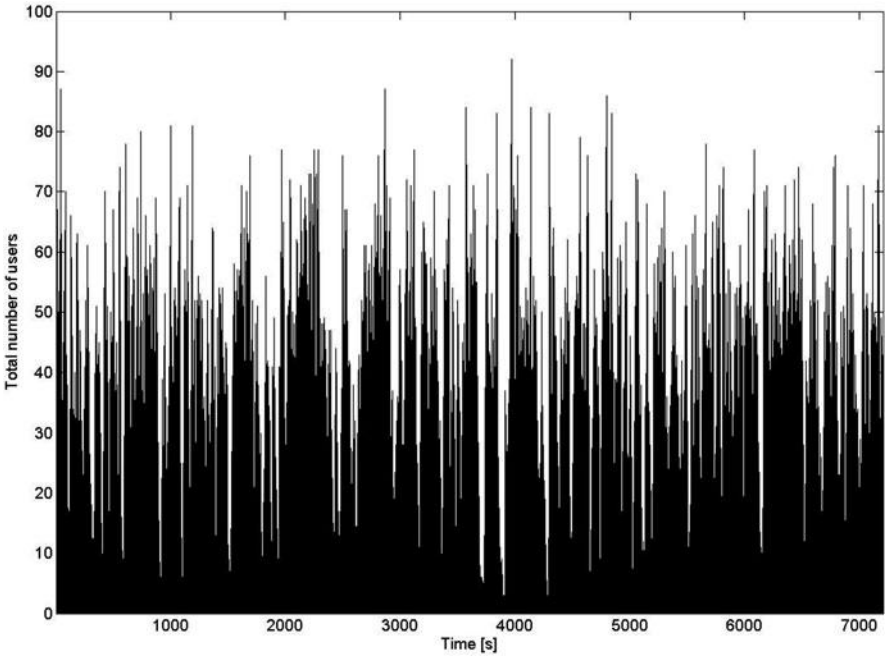


Fig. 8 Total number of users connected to the GNs in Scenario 2

4.1 Comments

As expected, the stream front d_2 of the GN 2 is averagely closer to the source stream front d in Scenario 1 with respect to Scenario 2, i.e., the GNs "follow" the source stream more timely. This reflects in the users' streams freezing (i.e., their front reaches the front of the GN minus the guard ϵ_L) only in Scenario 2, as can be seen both from Figs. 5 and 6. In fact, in Fig. 5 the value is equal to ϵ_L for a total freezing time of 15 s.

Instead, concerning the number of users, it turns out, as expected, that Scenario 2 is the most satisfactory, as Figs. 7 and 8 show. In particular, the average number of served users through the horizon is equal to 36 for Scenario 1 and 38 for Scenario 2, starting from an initial population of 12 users. Moreover, as to the quality of the received stream (in terms of encoding levels), averagely users tend to be served at a quality of 76 and 78% in Scenario 1 and 2, respectively.

5 Conclusions and Future Work

In this chapter we presented a novel content distribution overlay relying upon a centralized decision-maker to carry out optimization. Simulations proved the effectiveness of the approach. Future work aims at performing a thorough simulation campaign and injecting different costs and use cases in the framework. Specifically,

owing to the nature of the centralized DM, it would be useful to investigate the signaling volumes exchanged to exploit the optimization of the CDN, in order to quantify both the timing and resource requirements in the perspective of granting scalability. Lastly, we point out that interactive grids are often equipped with interactive multimedia collaborative tools. In this perspective, the framework could be also tested as a way to distribute real-time audio/video, e.g., to deliver lectures or comments while an experiment is ongoing.

References

1. I. Foster, C. Kesselman, and S. Tuecke Anatomy of the Grid. International Journal of Supercomputer Applications, 15(3), 2001
2. S. Tuecke, K. Czajkowski, I. Foster, S. Graham, C. Kesselman, P. Vanderbilt, and D. Snelling. Grid Service Specification. Open Grid Service Infrastructure Working Group (OGSI), Global Grid Forum, 2003.
3. I. Foster, and C. Kesselman. The grid: Blueprint for a new computing Infrastructure, Morgan Kaufmann, San Francisco 1999.
4. S. Vignola, and S. Zappatore. The LABNET-server software architecture for remote management of distributed laboratories: current status and perspectives. In Distributed cooperative laboratories, issues in networking, instrumentation and measurement, F. Davoli, S. Palazzo, S. Zappatore Eds., Springer, USA, pp. 435–450, 2006.
5. L. Caviglione, F. Davoli, P. Molini, and S. Zappatore. Design and preliminary analysis of a framework for integrating real and virtual instrumentation within a grid infrastructure. Proceedings International Symposium on Performance Evaluation of Computer and Telecommunication System (SPECTS'06), Calgary, AB, July 2006
6. L. Caviglione, and L. Veltri. A p2p framework for distributed and cooperative laboratories, 2005 Tyrrhenian international workshop on digital communications, Sorrento, Italy, July 2005. In *Distributed cooperative laboratories - networking, instrumentation and measurements*, F. Davoli, S. Palazzo, S. Zappatore, Eds., Springer, Norwell, MA, pp. 309–319, 2006
7. S. Sen and J. Wang. Analysing Peer-To-Peer traffic across large networks. IEEE/ACM Transactions on Networking, 12(2), 219–232, April 2004
8. M. Zhang, Q. Zhang, L. Sun, and S. Yang. Understanding the power of pull-based streaming protocols: Can we do better?. IEEE Journal on Selected Areas in Communications, 25(9) 1678–1694, December 2007
9. X. Zhang, J. Liu, B. Li, and T.-S.P. Yum. Coolstreaming/donet: A data-driven overlay network for efficient media streaming. IEEE INFOCOM 2005, Miami, FL, March 2005.
10. J. Li, P.A. Chou, and C. Zhang. Mutualcast: An efficient mechanism for one-to-many content distribution. Proceedings of ACM SIGCOMM - Asia Workshop 2004, Beijing, 2004
11. S. Chul Han, and Y. Xia. Network load-aware content distribution in overlay networks. Computer Communications, 32(1), 51–61 January 2009
12. C. Wu, and B. Li. Optimal rate allocation in overlay content distribution. Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet (NETWORKING 2007), in Lecture Notes in Computer Science, Springer, USA, Vol. 4479/2007 pp. 678–690, 2007
13. N. Laoutaris, V. Zissimopoulos, and I. Stavrakakis. On the optimization of storage capacity allocation for content distribution. Computer Networks, 47(3) 409–428, February 2005
14. N. Laoutaris, V. Zissimopoulos, and I. Stavrakakis. Joint object placement and node dimensioning for internet content distribution. Information Processing Letters, 89(6) 273–279, March 2004
15. D. Bertsekas. Dynamic programming and optimal control (2nd Edition), Athena Scientific, Belmont, 2000

16. C. Cervellera and M. Muselli. Efficient sampling in approximate dynamic programming algorithms. Computational Optimization and Applications, 38(3) 417–443, December 2007
17. F. Allgöwer and A. Zheng. Nonlinear model predictive control. In Applications of nonlinear predicticve control, F. Allgöwer, A. Zheng, Eds., Basel, Birkhäuser, Switzerland, 2000
18. A. Törn, and A. Žilinskas. Global optimization, Springer, Berlin, 1989
19. H. Niederreiter. Random number generation and Quasi-Monte Carlo methods, SIAM, Philadelphia, PA, 1992
20. J. van der Meer, D. Mackie, V. Swaminathan, D. Singer, and P. Gentric. RTP payload format for transport of MPEG-4 elementary streams. Network Working Group, IETF, RFC 3640, November 2003.

EDGeS Bridge Technologies to Interconnect Service and Desktop Grids

P. Kacsuk, Z. Farkas, and Z. Balaton

Abstract Although there are many production level service and desktop grids they are usually not able to interoperate. The European FP7 EDGeS project aims at integrating service grids and desktop grids to merge their benefits. The established production infrastructure includes EGEE (gLite based service grid) and BOINC and XtremWeb desktop grids. The chapter describes the bridging technology developed in EDGeS both for the DG → SG and SG → DG directions.

1 Introduction

Current Grid systems can be divided into two main categories: service grids (SG) and desktop grids (DG). Service grids are typically organized from managed clusters and provide a 24/7 service for a large number of users who can submit their applications into the grid. The service grid middleware is quite complex and hence relatively few managed clusters take the responsibility of providing grid services. As a result the number of processors in SGs is moderate typically in the range of 1,000–50,000. Even the largest SG system, EGEE[1] has collected less than 100,000 computers.

Desktop grids are collecting a large number of volunteer desktop machines to exploit their spare cycles. These desktops have no SLA requirement, their middleware code is extremely simple to install and hence the typical number of volunteer desktop grids is in the range of 10,000–1,000,000. However, their drawback is that they can execute only some very limited number of pre-registered applications and users can not submit other applications to these systems. The most well-known volunteer desktop grid is SETI@home[2] that collected over 2 Million CPUs.

Comparing the security mechanisms in SG systems any user who has an accepted grid certificate can submit any kind of applications into the grid. The grid trusts the certified users and as long as these users do not cause any problem to the grid they can use it with their own applications. User certificates are not used in DG

Z. Farkas (✉)
MTA SZTAKI, Budapset, Hungary
e-mail: zfarkas@sztaki.hu

F. Davoli et al. (eds.), *Remote Instrumentation Services on the e-Infrastructure*,
DOI 10.1007/978-1-4419-5574-6_5,

systems, DG workers (clients) trust the applications and not the users, and hence only well tested and validated applications can be used in DG systems. These are registered in the application database of the DG server and are typically long running grand-challenge scientific applications like climate research, protein folding, cancer research, etc.

Comparing the possible applications we can say that SG systems are much more flexible than DG systems. DGs are best suited to compute-intensive bag-of-task (master-worker or parameter sweep) applications where the server (master) sends out independent work units (WU) to the worker (client) machines. The code of the WUs is the same but the data they process are different. Although the SG systems can execute other type of applications (MPI, data-intensive, etc.) surprisingly in many cases SGs are also used for executing compute-intensive bag-of-task applications.

Comparing the price/performance ratio of SGs and DGs the creation and maintenance of DGs is much cheaper than that of SGs. Therefore it would be most economical if the compute-intensive bag-of-task applications could be transferred from the expensive SG systems into the cheap DG systems and executed there. On the other hand, when an SG system is under-loaded its resources could execute WUs coming from a DG system and in this way existing SG systems could support the solution of grand-challenge scientific applications.

The recognition of these mutual advantages of integrating SGs and DGs led to the initiation of the EDGeS (Enabling Desktop Grids for e-Science) EU project that was launched in January 2008 with the objective of integrating these two kinds of grid systems into a joint infrastructure in order to merge their advantages into one system. The EDGeS project integrates the EGEE service grid with BOINC[3] and XtremWeb[4] DG systems.

The present chapter describes the main achievements of the EDGeS project. Section 2 shows the EDGeS production infrastructure including its main services. Section 3 explains the concept of realizing the BOINC → EGEE direction bridge. Section 4 describes the principles and implementation details of the EGEE → DG interconnection. Section 3 and 4 put special emphasis on what a joining DG or EGEE VO should install in order to become part of the EDGeS infrastructure.

2 The EDGeS production infrastructure

The EDGeS production infrastructure integrates the EGEE (gLite-based) service grid infrastructure with BOINC-based and XtremWeb-based desktop grids. The EGEE grid infrastructure is partitioned as virtual organizations (VO). A certain VO is a collection of people who want to jointly work on a certain area and hence they share resources that also belong to this VO. In EGEE there are about 200 VOs organized either according to scientific application fields (e.g. physics VOs, biomed and bioinfo VOs) or according to territorial organizations (e.g. VOCE (Virtual Organization of Central Europe), SEE (South-East Europe) VO).

EDGeS also created a new VO (called as desktopgrid.vo.edges-grid.eu) in order to collect those resources that are willing to support the connected DG systems. This VO contains the usual EGEE VO services like VOMS, UI, WMS, BDII, LB, CEs and SEs but it also contains some additional services in order to enable the execution of work units (WUs) arriving from the interconnected DG systems.

The EDGeS VO provides a DG → EGEE bridge service installed on a UI machine. This bridge is based on the 3G Bridge (Generic Grid-Grid Bridge) technology[5] developed in the EDGeS project. The 3G Bridge with *N* BOINC input plugins (called as Job Wrapper Clients) can register to *N* different BOINC servers and can download WUs from these servers. These WUs are stored in the internal database of the 3G Bridge and then by the gLite output plugin of 3G Bridge these WUs are transformed into gLite jobs and submitted to the WMS of the EDGeS VO. Finally, the WMS allocates CEs for these jobs inside the EDGeS VO.

EDGeS provides a basic set of CEs as a starting service to execute the WUs of DGs connected to EDGeS. However their number is quite limited, 4 CEs provided by SZTAKI (16 CPUs), CIEMAT (20 CPUs) and CNRS (1554 CPUs) as shown in Fig. 1. In order to attract more and more DGs to EDGeS we have to increase the number of CEs in the EDGeS VO. Therefore, those EGEE VOs who want to run applications on the connected DG systems should provide new CEs for the EDGeS VO. This is organized in the usual way as supporting CEs are collected for any VO in EGEE.

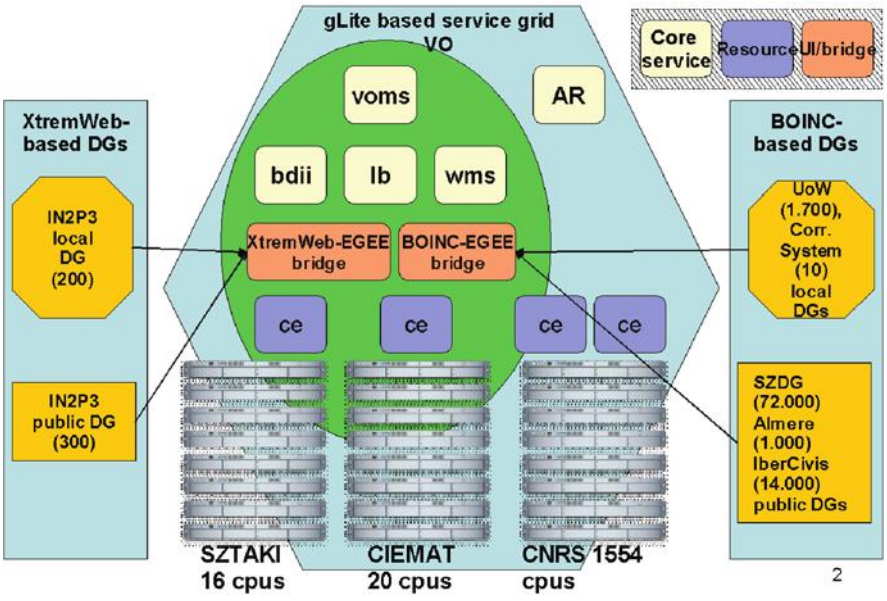


Fig. 1 Current EDGeS infrastructure

Figure 1 depicts the current EDGeS infrastructure showing those DGs that are interconnected with the EDGeS VO. These include volunteer (public) and institutional (local) DG systems:

- *SZDG (SZTAKI Desktop Grid)*[6 this is a volunteer DG system based on BOINC and established in 2005. It currently runs two applications and collected over 70.000 processors worldwide.
- *IberCivis* another volunteer DG system based on BOINC and established in 2008 with the goal of supporting applications of Spanish scientists. It currently runs 6 applications and collected over 14,000 processors.
- *AlmereGrid* a volunteer town grid collecting desktops from the citizens of Almere (a Dutch town) in order to run applications relevant for the citizens of Almere. This has over 1,000 processors and has both BOINC and XtremWeb version. When writing this paper only the BOINC version has been connected to the EDGeS VO.
- *UoW (Univ. of Westminster) local DG* this DG connects the computer lab machines of 6 campuses of the university into an institutional DG system. About 1700 desktops are connected based on BOINC. The primary goal is to run scientific research applications developed at the university.
- *Correlation Systems Ltd. local DG* this is a small size (10 desktops) company DG system to support the applications of the company. It uses the bridge to get more resources from EGEE.
- *IN2P3 public DG (300 processors)* it is based on XtremWeb technology and ready to accept applications not only from IN2P3.
- *IN2P3 local DG (200 processors)* it is based on XtremWeb technology and used to run applications coming from IN2P3.

3 Supporting BOINC DGs by EDGeS

In Section 2 it was described how the EDGeS production infrastructure is organized. In this section we explain the basic operational principles of supporting BOINC DGs by the EDGeS VO. Figure 2 shows the internal structure of the EDGeS VO when it supports two BOINC DGs. The main difference between an ordinary EGEE VO and the EDGeS VO is that the latter one provides a 3G Bridge service on a UI machine. The 3G Bridge service has three main parts:

1 JW Client (JobWrapper Client) that is a modified BOINC client. It registers to a BOINC server project as a powerful multi-processor system and hence it is able to download much more work units (WUs) than a usual desktop client. As the figure shows we have to apply as many JW Clients in the 3G Bridge service as many BOINC systems are supported by the EDGeS VO.
2 QM (Queue Manager and database) stores the incoming WUs in a database organized into different queues.
3 EGEE Plugin transforms the WUs into regular gLite jobs by creating the necessary JDL description. Using the UI services of the UI machine it transfers the

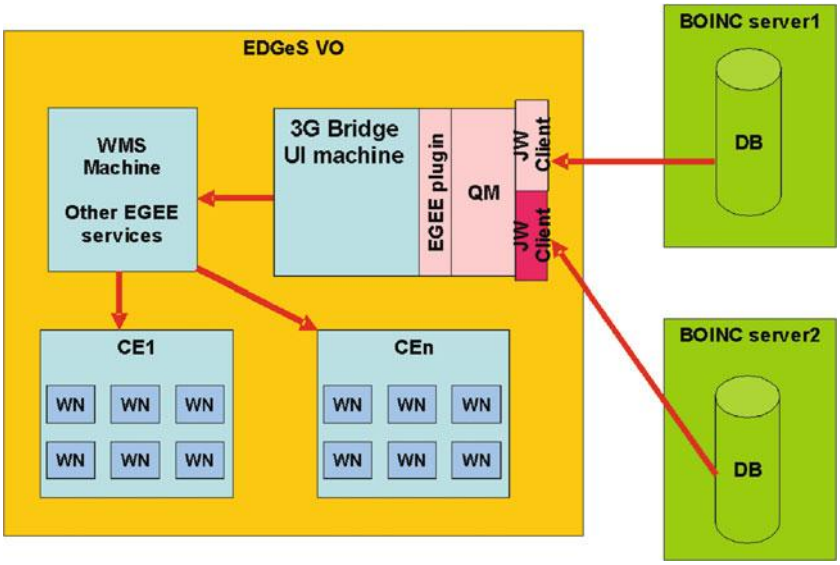


Fig. 2 Internal structure of the EDGeS VO

jobs to the WMS of the EDGeS VO. It is the task of the WMS to allocate the jobs coming from the 3G Bridge to the available CEs of EDGeS VO.

As Fig. 2 shows the structure of the EDGeS VO and the principle of supporting BOINC systems by the EDGeS VO is quite simple. The advantage of the solution is that it does not require any modification in the BOINC systems that are connected to EDGeS. In order to collect WUs from a joining BOINC server by the EDGeS VO there is no installation work at the BOINC side. The only thing the BOINC system administrator should do is to get a grid certificate that is valid and accepted by the EGEE community and then maintain a proxy delegation in the EDGeS MyProxy service. The bridge is then configured to transfer every WU arriving from this BOINC server to the EDGeS VO with a certificate proxy of the BOINC system administrator. This is important since an EGEE VO trusts the user and not the application (as described in the Introduction). A BOINC system typically has got only very few applications that are typically installed by the BOINC system administrator and hence he can act as the responsible EDGeS VO user on behalf of his BOINC system. In case of XtremWeb, users must submit their jobs with a valid proxy certificate.

4 Supporting EGEE VOs by EDGeS

The EDGeS infrastructure is an open infrastructure: any DG or SG system can join it. The goals of connecting a DG system to EDGeS might be that the joining DG would like to exploit the spare resources of the EDGeS VO. As shown in Section 2

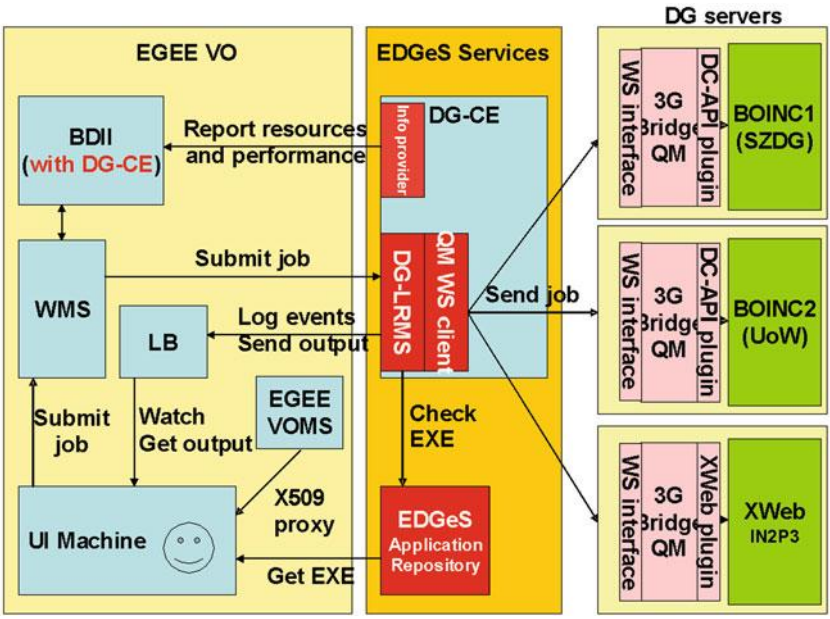


Fig. 3 EDGeS services and bridge technology to connect EGEE VO to DGs

currently more than 1500 processors are available in the EDGeS VO that is quite significant computing capacity and worth exploiting by a joining DG. However, the benefits should be mutual for both the DG systems and the EGEE user community and hence the joining DGs should enable the execution of some EGEE applications that are already ported to EDGeS. In this section we show how to support EGEE VOs by the connected DGs using the EDGeS technology and services.

The EDGeS solution of enabling EGEE VOs to send jobs to EDGeS connected DG systems is depicted in Fig. 3. The heart of bridging EGEE VO jobs to DG systems is represented by the two EDGeS bridge services shown in Figure 3:

1 the EDGeS Application Repository service and
2 the DG-CE computing element service.

As it was mentioned in the introduction, DG systems trust applications and not users. Therefore, not all applications of EGEE VOs can be sent to the interconnected DG systems, only those that were validated. Validation of applications is the task of the EDGeS project (together with the code developers) and the validated applications should be stored in the EDGeS Application Repository (AR) that is based on GEMLCA, a grid job submission and repository service developed by the University of Westminster[7]. Once an EGEE VO user would like to run an application that is available in the EDGeS AR he can exploit the resources of the connected

DG systems and hence can execute his applications much faster than using only the EGEE VO.

EDGeS also provides a special computing element, called as CE-DG in order to transform gLite jobs to the WU format of the interconnected DG systems. The task of this service is to accept from the WMS of the EGEE VO those jobs that should be transferred to one of the interconnected DG systems. The CE-DG first checks if the submitted application code is stored in the EDGeS AR and then checks if the connected DG supports the given application. If yes, then sends the job to the selected DG by invoking the web service interface of the 3G Bridge installed on the selected DG.

Notice that EDGeS AR and DG-CE are services of EDGeS provided for all the EGEE VOs that participate in the EDGeS infrastructure. To join an EGEE VO to the EDGeS infrastructure is extremely easy: the DG-CE should be registered into the BDII of the joining VO. However, the joining EGEE VOs that want to take advantage of the connected DGs should provide additional resources to extend the computational capacity of the EDGeS VO by allocating some percentage of their existing CEs.

Supporting job transfers from EGEE VOs to a DG system requires more activity on the DG side. In this case the DG administrator should install 3G Bridge on the DG server. The QM part of this bridge is the same as the QM part of the BOINC → EGEE bridge. In the case of BOINC DGs the output plugin is the DC-API-Single plugin developed in SZTAKI in order to generate BOINC WUs using DC-API function calls[7]. In the case of XtremWeb DGs the output plugin is the XtremWeb plugin developed in INRIA. The EGEE input plugin of this 3G Bridge is replaced with a WS Interface in order to call its services remotely by the DG-CE computing element. The real EGEE input plugin of the 3G Bridge is placed on the DG-CE computing element. Finally, the DG system administrator should register some applications from the EDGeS AR into his DG server in order to enable the execution of those EGEE applications.

Notice that a connected DG system does not have to support all the applications in the EDGeS AR. In order to support all the applications stored in the EDGeS Application Repository EDGeS plans to establish a brand new volunteer BOINC system in order to provide 100% support for EGEE applications. The advantage compared to other connected DG systems is that it will support every EDGeS application and will support only EDGeS applications. As a result all the resources collected in this DG will be 100% devoted to EDGeS application support. (Other connected DGs have their own applications and they do not have to support all the EDGeS applications.) Similarly to this dedicated volunteer BOINC system EDGeS plans to establish a dedicated volunteer XtremWeb DG system, too. The objective of EDGeS is that by the end of the project more than 100,000 CPUs should be connected to the EDGeS infrastructure in order to support EGEE applications.

The EDGeS monitoring infrastructure[8] is available to monitor the number of jobs and WUs that are passing the EDGeS bridge in both directions and hence both the connected DG and VO administrators can check the usage and contribution of their resources in the overall EDGeS system.

5 The User View of EDGeS

Our goal was to make the EDGeS infrastructure as transparent to the EGEE users as possible. It means that they should be able to run EDGeS applications in the EDGeS infrastructure in the same way as they do in the original EGEE infrastructure. First of all, we have to define what an EDGeS application means from the EGEE user point of view. An EDGeS application is an application that can run both in EGEE and desktop grids. Typically, EGEE applications are ported to desktop grids by the EDGeS consortium but it can happen that originally desktop grid applications are ported to EGEE.

In any case the application should be validated as mentioned before and then the validated applications are stored in the EDGeS Application Repository (AR). The EGEE users can only execute those applications in the connected DG systems that are stored in the EDGeS AR. The user interface of EDGeS AR is a public web page, through which EGEE users can browse and download the EDGeS applications in a form that is directly submittable for them using the usual JDL-based submission method. As Fig. 4 shows, this web page first lists all the applications available in the repository (at the moment "DSP compute" and "emmil"). Then a detailed description is provided for every application. The detailed description starts with a short description of the application, then comes the downloadable executable which could be used by the EGEE user when he wants to submit the application to the EGEE VO. Notice that all the VOs and CEs are listed for which this application is

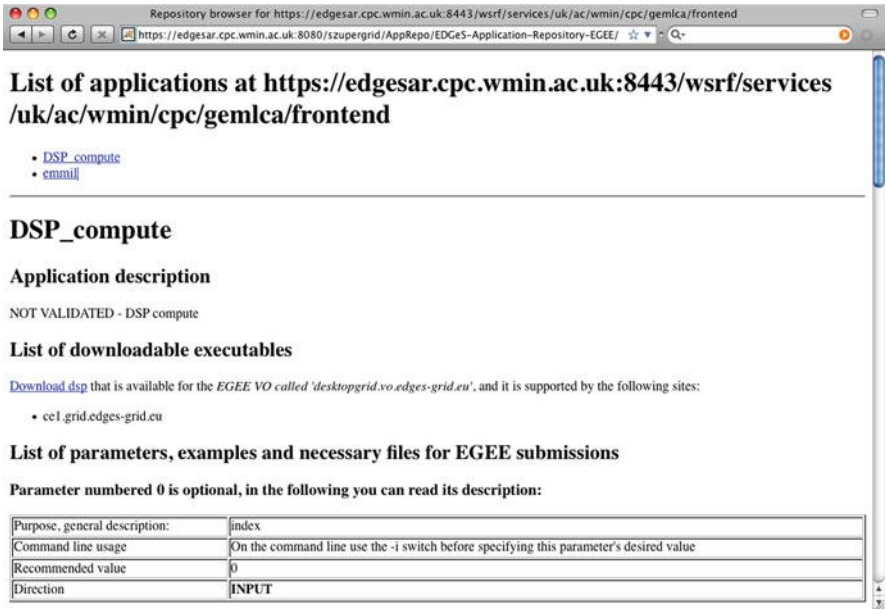


Fig. 4 User interface of the EDGeS services AR

submittable so EGEE users will know if their VO enables this application or not and if yes on which CE it can be executed. Finally a detailed list of parameters are given to help the users understand how to use this application and what to put in the JDL.

There is one problem we have to solve when users submit their jobs to an EGEE VO that uses the EDGeS bridge services. We have to prevent jobs coming from DGs to go back to the DG-CE computing element service and have to ensure that users not using the bridge are not effected in any way and their JDLs work as before without any changes. In order to solve this problem we exploit the "role" feature of the VOMS.

By using a new VOMS role, the VO administrator can select users who can get VOMS attributes that contain the given role and who not. Also, on the CE side, the DG-CE administrator can set up the queues on the DG-CE to accept only proxies that have the given role among the VOMS attributes. Finally, if a normal EGEE CE is not configured to support such roles by default, it will accept proxies with such roles. As a consequence, jobs submitted using this special role will be able to run on both DG-CEs and traditional EGEE CEs, but jobs submitted using a proxy lacking this special role will be able to run only on traditional EGEE CEs. For example, in case of the production EDGeS VO, this special role is called "BridgeUser". Users can create proxies usable by the DG-CE using this command:

```
voms-proxy-init -voms \
 destopgrid.vo.edges.grid.eu:/desktopgrid.vo.edges-grid.eu/
    Role=BridgeUser \
 -order /desktopgrid.vo.edges-grid.eu/Role=BridgeUser
```

Once the user has created such a proxy, he or she can force the WMS to send jobs to the DG-CE using one of these methods:

- During job submission, use the "-r" command-line argument of the glite-wms-job-submit command to send the job the DG-CE. The string following the "-r" argument represents the destination DG-CE.
- Use the "SubmitTo" attribute in the JDL file to specify which DG-CE should run the job. If this attribute has a valid value in the JDL file, the WMS will send the job to the given DG-CE without matchmaking (see the example JDL below).
- Use the "Requirements" attribute in the JDL file to specify which DG-CEs the WMS is allowed to match the job to.

The first two methods are equivalent, the third gives more freedom for both the user and the WMS in terms of resource selection. An example JDL using the second method is the following:

```
[
     JobType = "Normal";
     Executable = "app.exe";
     Arguments = "-d -n -m";
```

```
    InputSandbox = "in.1", { "in.2" };
    OutputSandbox = { "out.file" };
    SubmitTo = "ce1.grid.edges-grid.eu:2119/
       jobmanager-edges-szdgr";
]
```

The advantage of the above solution is that minor user work is needed to run a supported application through the EGEE → DG bridge, and EGEE users who are not interested in using the EGEE → DG do not have to change anything in their existing jobs at all.

6 Conclusions

The objective of creating the EDGeS infrastructure is to interconnect or integrate service grids and desktop grids into a joint infrastructure where users can seamlessly exploit all the available resources both from the connected DGs and SGs. The established EDGeS production infrastructure interconnects one EGEE VO (with more than 1500 processors) and 7 DG systems (with nearly 90,000 processors).

The EDGeS infrastructure is an open infrastructure: any DG or SG system can join it. Currently two projects work on the interconnection to the EDGeS infrastructure:

1 The European SEE-GRID-SCI project has got a gLite based grid infrastructure consisting of 3 VOs. At least one of its VOs will be connected to EDGeS.
2 The Institute for System Analysis of Russian Academy of Sciences creates an institutional DG system that will be connected to EDGeS VO.

The long term goal is to connect many different VOs of EGEE to EDGeS and extending them with large DG capacity. In return they will provide CE resources for the EDGeS VO. Another important goal of EDGeS is to connect more and more DG systems in order to increase their computational capacity with the resources of the EDGeS VO and in return to provide computing power for those EGEE applications that are validated and stored in the EDGeS Application Repository.

Very importantly, the EDGeS technology can be used independently from the EDGeS infrastructure. Any EGEE VO and DG can create their own integrated system based on the EDGeS technology/software that can be downloaded from the EDGeS web site [9]. For example, this enables for any EGEE VO to extend its capacity by creating for example a local university DG and connect it to the VO as a new resource.

The ultimate goal of EDGeS is to provide a generic SG-DG bridge technology that can be used beyond gLite, BOINC and XtremWeb. The 3G Bridge developed in EDGeS seems to be a generic technology that can be easily adapted to any SG and DG system by writing the corresponding input/output plugins. For example, members of the European/Latin-American EELA-II project adapt the 3G Bridge in order to interconnect their gLite-based and OurGrid-based labs[10].

Acknowledgements The EDGeS (Enabling Desktop Grids for e-Science) project receives Community funding from the European Commission within Research Infrastructures initiative of FP7 (grant agreement Number 211727).

References

1. M. Lamanna and E. Laure (Eds.). The EGEE user forum – experiences in using grid infrastructures. Journal of Grid Computing, 6(1), 1–123, 2008
2. D. Anderson, J. Cobb, E. Korpela, M. Lebofsky, and D. Werthimer. Seti@home: An experiment in public-resource computing. Communication ACM, 45(11):56–61, 2002
3. D. Anderson. BOINC: A system for public-resource computing and storage. Proceedings of the 5th IEEE/ACM International GRID Workshop, Pittsburgh, PA, 2004
4. G. Fedak, C. Germain, V. Nri, and F. Cappello. XtremWeb: A generic global computing platform. Proceedings of 1st IEEE International Symposium on Cluster Computing and the Grid CCGRID-2001, Special Session Global Computing on Personal Devices, Brisbane, QLD, IEEE/ACM, IEEE May 2001 pp. 582–587
5. P. Kacsuk, Z. Farkas, and G. Fedak. Towards making BOINC and EGEE interoperable. Int. Grid Interoperability and Interoperation Workshop, IGIIW, Indianapolis, In, 2008
6. Z. Balaton, G. Gombas, P. Kacsuk, A. Kornafeld, J. Kovacs, A. C. Marosi, G. Vida, N. Podhorszki, and T. Kiss. Sztaki desktop grid: A modular and scalable way of building large computing grids. Proceeding of the 21th International Parallel and Distributed Processing Symposium, March 2007, Long Beach, CA, 26–30, 2007
7. T. Delaittre, T. Kiss, A. Goyeneche, G. Terstyanszky, S.Winter, and P. Kacsuk. GEMLCA: Running legacy code applications as grid services. Journal of Grid Computing, 3 (1–2), 75–90, 2005
8. F. Araujo. Monitoring system for the EDGeS infrastructure, INGRID workshop, Alghero, Italy, 2009
9. EDGeS project web page: http://www.edges-grid.eu/, last access 2010
10. W. Cime, F. Brasileiro, N. Andrade, L. B. Costa, A. Andrade, R. Novaes and M. Mowbray. Labs of the World, Unite!!!. Journal of Grid Computing, 4(3), 225–246, 2006

Management Challenges of Automated Service Level Agreements

Constantinos Kotsokalis and Philipp Wieder

Abstract Service Level Agreements (SLAs) are formal electronic contracts between customers and providers of services, describing the rules governing service consumption. They can be established automatically by software agents given proper target utilities, or may be long-standing indicating less volatile business relationships. This position paper reflects history and state of the art in management of automated Service Level Agreements, and discusses main challenges in this area, in the context of complex service hierarchies and the corresponding SLA hierarchies. Then, it goes on to propose future directions for key topics such as modeling SLAs, planning them during the negotiation phase, monitoring them and enforcing them through adjustment actions.

1 Introduction

In the last few years we have seen a conversion of traditional IT into *Service-Oriented Architectures* (SOA) and *Service-Oriented Infrastructures* (SOI). Service orientation refers to exposing a certain piece of functionality, or value, to be consumed by customers who own neither the software nor the hardware, in a manner that usually implies remote access. As such, customers pay only for the value offered to them by means of some remotely executed algorithm on infrastructure they did not have to buy. Such developments are bliss for plenty of enterprises, of all sizes, as they do not need to spend funds on infrastructure, and do not have to develop their own software – but rather, they decide which service(s) to use based on functionality, and pay according to their usage patterns. Even in cases where there is no outsourcing, but rather there is an enterprise (company-level) SOA, separation of concerns facilitates productivity in the same manner.

This paradigm shift has already affected the way people conduct business today, started new markets and a complete new sector of IT, typically referred to as *Service*

C. Kotsokalis (✉)
Dortmund University of Technology, Dortmund, Germany
e-mail: constantinos.kotsokalis@udo.edu

F. Davoli et al. (eds.), *Remote Instrumentation Services on the e-Infrastructure*,
DOI 10.1007/978-1-4419-5574-6_6,

Computing. Service computing does not differentiate between different kinds of clients to a service, and whether they are humans or software agents. As such, the customer of an IT service may be another IT service; this combination of services into service chains lies in the core of service computing concepts. Software re-usability is one of the main advantages, as it increases its value tremendously and allows enterprises to focus on what they do best, instead of re-implementing things that other companies have already implemented. Therefore, it is not uncommon nowadays to have services which consist purely by invocations and orchestration of other services, gluing together results in appropriate ways to create new functionality. Such services may then be reused by services themselves, effectively resulting in *complex service hierarchies* that possibly span more than one administrative domains and service providers.

The exact business and architectural models of service computing differ from one year to the next: Until recently, *Grid Computing* was the sole big player in the IT services arena. Currently, it is *Cloud Computing* that has captured the interest of businesses and researchers alike. Additional models with lower market penetration rates appear, and recently are gathering under the Cloud Computing grouping. Nevertheless, from a manageability perspective, all service computing models present similar challenges.

One such challenge is related to receiving guaranteed *Quality of Service* (QoS) from the service invoked/used by the consumer. This has actually become a very central issue, which to some extent also obstructs the further adoption of service computing, especially in mission-critical systems. More specifically, businesses are rightfully considerate of the fact that they do not own the equipment, they have not developed the software, and their data is stored (or processed) elsewhere. Such lack of control also means lack of the ability to react to failures, in the way that each business sees best according to its procedures. *Service Level Agreements* (SLAs) try to solve this problem, by augmenting service computing with acceptable levels of determinism, so that customers can know what to expect, and what compensation they will receive if things go wrong. Extending SLAs to more than one services, into a hierarchy of them, results in corresponding *SLA hierarchies*, which may have the same or a much different structure than the respective service hierarchies.

This position paper tries to provide an overview of the area of management for single SLAs as well as SLA hierarchies: What such management includes, current state of the art, and current challenges for which we also give proposals for future directions. The rest of the chapter is organized as follows: Section 2 discusses SLAs in general: What they are, and how they are expressed. Section 3 focuses more specifically on the topic of SLA hierarchies, reflecting service hierarchies in SOA environments. Section 4 is the main part of this paper, discussing current challenges and suggested directions for future research. Finally the chapter concludes with Section 5 which summarizes the discussion.

2 Service Level Agreements

Service Level Agreements can be considered as the IT-equivalent of real-life contracts. As such, a SLA defines obligations and rights for the parties involved, as regards the service consumed. We typically say that the SLA *governs* the service consumption. It was already mentioned that SLAs try to augment services with acceptable levels of determinism. What this means is that, purely on a document level, SLAs also include measures of business value and penalty constructs for those cases that the agreed QoS level cannot be maintained by the service provider. It is then expected that, during the negotiation phase and the planning that takes place in parallel, SLA-aware middleware can take those values into account.

Service Level Agreements are being used already extensively in various parts of the IT industry, but their first applications were in computer networking by means of facilities such as *Differentiated Services* [1] (DiffServ). SLAs were initially contracts that defined things such as allocated bandwidth, quality of networking circuits, etc. SLA research in the networking area is still very active (e.g. [2,3,4]), nevertheless, with the advent of service computing and the "everything as a service" principles, the concept was applied to plenty of different areas and *automated SLAs* [5] are considered in more recent research. Automation here is related to the absence of human intervention to decide what is acceptable and what not, and perform the negotiation and runtime management. Conversely, we assume autonomous agents that can perform all decision-making during negotiation and runtime, given predefined policies and business logic to drive decisions.

In order to achieve automated message exchange and reasoning over agreement offers, we need at the very least some formalisms and conventions that the agents must comply with. The first popular formalism for interoperable automated Web-Service based SLAs was the *Web Service Level Agreement* [6] (WSLA) language introduced in 2003. Shortly after that, an effort started within the Open Grid Forum to build the *Web Services Agreement Specification* [7] (WS-Agreement) in the context of a standardization body that would bring together all important industry and research players. Indeed, in 2007 WS-Agreement was released, nevertheless it is adopted rather slowly, also for reasons discussed later in Section 4.1.

3 SLA Hierarchies

In Section 1 we already briefly mentioned how service dependencies reflect on SLA dependencies. The most typical case for service dependencies, is that of business processes (workflows) that are made available as services themselves [8]. In this case, an orchestration engine for the workflow is invoking external services and uses

the results of the invocation as input to the next one, or as the output of the complete process. In their seminal paper from 2004, Cardoso et al. [9] provided a model for calculating the QoS of a workflow, starting from the QoS offered by the individual services that the workflow invokes. In this case, QoS comprises time, cost, reliability and fidelity. Looking at the problem independent of the exact QoS parameters for a service, and modeling QoS not simply as what the service offers (perhaps in a best-effort fashion) but rather as what the service *commits to support and maintain*, we have the core of SLAs and their inter-dependencies. We can generalize the problem for services that rely on other services in any way. For instance, software services need computing infrastructure on which to execute. If this is offered externally in the form of a service (for instance, a Cloud computing infrastructure) then this service dependency has to be reflected on SLAs that concern the software service. Or, if some policy dictates that a service cannot operate without another service in operation (even if they do not invoke each other or rely on each other directly), this may affect the SLAs established for the former. Figure 1 illustrates this concept. *Service A* depends on *service B* and *service C*. The consumption of A by the customer is governed by *SLA 1*, and the consumption of B and C by A is governed respectively by *SLA 2* and *SLA 3*. Then, SLA 1 depends on SLA 2 and SLA 3, forming a similar hierarchy reflecting the hierarchy of service dependencies.

The problem of managing SLA hierarchies is the main topic of the SLA@SOI EU project,[1] which inspired and supported this work. The project addresses all stages in the life-cycle of SLAs taking into account such inter-dependencies. Although IT services are the core of the project's focus, very distinct use cases are also evaluated, such as one for e-Government services. [10] provides preliminary information on the framework to be produced by SLA@SOI.

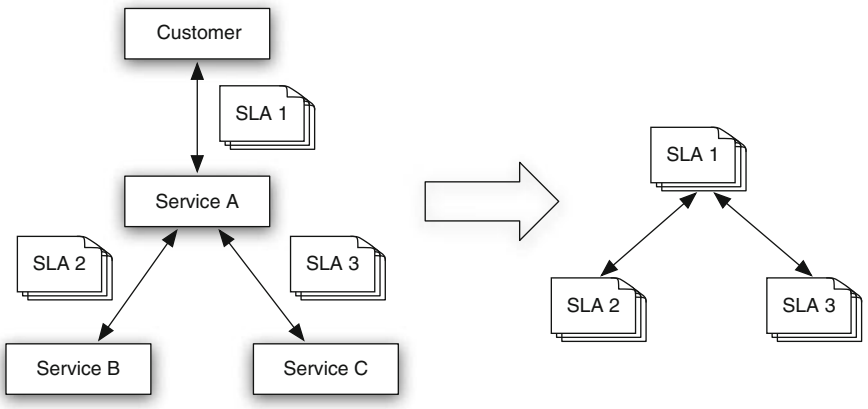


Fig. 1 Service and SLA dependencies forming hierarchies

[1] http://www.sla-at-soi.eu/

4 SLA Challenges and Proposed Future Directions

Looking at the various factors that we have to take into account to realize the SLA life-cycle for a specific domain or a specific use case (Fig. 2), it becomes obvious that splitting between abstract and concrete when defining a SLA model is a challenging task. Regarding Service-Oriented Infrastructures, a number of potential solutions for SLAs exist in addition to WSLA and WS-Agreement, as described by Parkin et al. [11]. Those solutions are often built upon one of the formalisms introduced in Section 2, making domain or use case-specific additions. This leads to the question what the missing pieces and remaining challenges for a standardized solution are, especially with regard to handling SLA hierarchies.

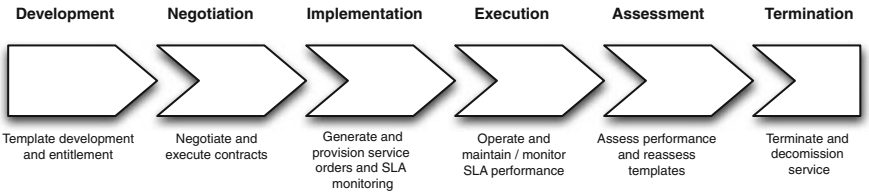


Fig. 2 The SLA life cycle according to the TeleManagement Forum [12]

In the following we try to answer this question discussing the issues, which we think are the most important to be solved in the near future. This includes (i) modelling and describing SLAs (related to the development phase within the SLA lifecycle), (ii) SLA planning (executed during the negotiation phase), (iii) monitoring of the negotiated QoS parameters (during the execution phase) and (iv) adjustment to SLA violations (also related to the execution phase).

4.1 SLA Model and Description

WS-Agreement has been specified to provide a domain-independent and standard instrument to establish Service Level Agreements. The specification [7] provides a formalism to describe agreement templates and agreements, a definition of an agreement protocol, and an interface to monitor the agreements state. The domain-independence makes WS-Agreement on one hand suitable for various application scenarios, but on the other hand also difficult to implement. Furthermore, the use of its extension mechanisms led to a variety of different domain-specific implementations which are not interoperable. As of now, there are no two interoperable WS-Agreement solutions which implement the full standard.

Regarding the other parts of the specification, negotiation and monitoring, the instruments provided by WS-Agreement are a good foundation for domain-specific solutions. Those issues are discussed in the respective sections.

In addition, we see a number of other issues which may obfuscate the wider acceptance of WS-Agreement, as there are (i) its dependency on the stateful

Web Services Resource Framework (WSRF) [13], (ii) its dependency on the WS-Addressing specification [14] and (iii) the combination of SLA formalism and negotiation protocol in one specification. Regarding issues (i) and (ii), execution environments implementing the respective standards and tooling support is needed, which may be discontinued especially since more light-weight stateless solutions, e.g. following the REST approach [15], are favored by a growing majority of projects. With respect to the third issue it is mainly standards compliance which cannot be maintained in those cases where the negotiation approach provided by WS-Agreement is not enough for a certain use case. This makes it impossible to follow the standard just using the SLA formalism and realizing a different negotiation protocol.

One approach to overcome the aforementioned limitations is to define a generic SLA model which is not XML-specific and also considers SLAs established in non Web Services environments. Furthermore, it should be possible to map this model to WS-Agreement, but also to WSLA, so that it does not carry the aforementioned limitations. Ongoing work in the SLA@SOI project is already producing the first results in this direction.

4.1.1 SLA Negotiation

The purpose of the negotiation phase within the SLA life-cycle is to reach an agreement between the customer and the service provider. To this extent, not only technical aspects are of importance, but also parties' responsibilities, payment details, or compensations. Negotiating multiple criteria can be a costly process (as also discussed in Section 4.2), which may cause uneconomically high transaction costs. For this reason, McKee et al. [16] propose to balance the advantages and disadvantages of negotiation carefully and execute multi-step negotiation only where its cost are justifiable.

An approach to keep negotiation costs low is the "supermarket approach" [16] also known as the "take-it-or-leave-it" approach. Its name correlates with the business model of supermarkets, where customers can choose between a range of brands buying a certain product. Regarding service offers those are often ordered according to the QoS offered, e.g. into gold, silver and bronze services. If the customer does not find the "brand" (i.e. the service level) they require, the may chose another "supermarket" (i.e. another service provider).

The WS-Agreement specification is also constrained to express either acceptance or rejection of an offered SLA. The GRAAP Working Group of the Open Grid Forum which is maintaining WS-Agreement, and other stakeholders consider this a limitation. Therefore extensions (or replacements) of the WS-Agreement protocol towards multi-step negotiation have been proposed [17] for use cases where legal or QoS benefits outweigh expensive negotiation. This is, for instance, the case for certain sophisticated scheduling applications. But also negotiating a SLA hierarchy, as described in Section 3, requires approaches to adjust the service terms to reflect dependencies and the current execution state of the corresponding services. Further discussion on the topic of decision-making during negotiation is provided later in Section 4.2.

4.1.2 Micro-Specifications

Having introduced a generic solution for SLAs such as WS-Agreement, it is *de rigueur* to allow implementers to extend it. Otherwise it is not possible to realize certain domain-specific aspects. The challenge of such an approach is to prevent the loss of interoperability. It might be the case, for example, that the service provider extends its service offer, but the consumer is unable to benefit from it as the extension cannot be processed at the consumer site. Therefore it is essential to define so-called micro-specifications under the umbrella of WS-Agreement to allow their wide adoption instead of defining multiple similar proprietary extensions for the same domain.

One example of such a micro-specification currently being defined is the one to allow reservation of resources or services in advance. Many use cases, from HPC parallel job scheduling, to execution planning in virtual environments, require a reservation of resources or services before the actual execution. This micro-specification qualifies, due to the large number of stakeholders and already existing proprietary solutions, for being standardized to work towards interoperable solutions.

4.2 Planning

4.2.1 Domain Specificity

SLA planning refers to the process leading to decisions whether an agreement offer can be accepted, or to calculations of utility resulting from incoming and for outgoing offers in an effort to optimize the value of the SLA. Planning is essentially driving negotiation, providing it with quotes, offers and counter-offers, etc. Even outside the context of SLA hierarchies and inter-dependencies, for isolated 1-to-1 negotiations, the problem of optimal planning remains extremely complex from a computational point of view. Ideally, we would like to find an algorithm that can perform well across all kinds of problems, and all kinds of SLA planning for different domains. Nevertheless, this is not possible: In [18], Wolpert and Macready have proved the theorem also known as “no-free-lunch in optimization”, which states that “any two optimization algorithms are equivalent when their performance is averaged across all possible problems” [19]. In other words, there is no single optimization method that performs equally well on any problem and domain. Therefore, it is impossible to create a framework for SLA planning/optimization that can be applied universally, since a contract (and therefore, a SLA) may concern very different types of services, resources, and guarantees.

Tackling this problem needs to take place in two levels: Architecturally for a given framework, and algorithmically for classes of problems. The former refers to pluggable frameworks that can support the straightforward integration of different planning methods, as long as they expose a common predefined interface. Such an approach would contribute to software reuse and could be an important advantage for any SLA management framework. On the algorithmic side, since we cannot

find a single algorithm that applies successfully to all problems, we need to create a classification of problems and identify those solutions which fit well to specific classes and domains. This classification may refer to resource types, planning deadline requirements, known combinatorial problems, etc. The more complete such a classification is, as regards the categories applicable, the more useful it would be for organizations that try to apply SLA planning and optimization to a particular use case.

4.2.2 Parallel Competing Negotiations

Service economies, as envisaged in recent years, assume that customers can easily discover services and start negotiating with them in such automatic ways as discussed earlier in this paper. This kind of automation also implies concurrency, and parallel negotiations that compete with each other. This becomes a serious synchronization problem, and a very complex one when we consider the interdependencies of services and the respective parallel negotiations. As an example, one may think of a workflow of two services invoked sequentially, and for which the service (workflow) provider wishes to achieve a total execution time of N time units. Each offer negotiated with the first service directly affects the second one, since their guarantees for response time must not exceed N, when added.

We argue that this problem can be addressed only by multi-disciplinary research, that combines many different facets of information technology. The areas that appear to be most relevant are game theory (which has already been taken advantage of in previous work for SLAs, e.g. [20,21]) and transactional information systems. We also believe that simulations in combination with pattern recognition methods can also facilitate this requirement: they can make it possible to monitor ongoing parallel negotiations, and based on pre-calculated knowledge (from simulation databases) decide what is the best course of action in order to minimize conflicting offers and converge faster.

4.2.3 Inference of SLA Hierarchies

It has already been discussed in Section 3 that some methods already exist to infer higher-level service QoS from the knowledge of lower-level services QoS. Yet, a full dynamic Internet of services dictates the requirement to waste no resources on static configurations, rather establish service consumption requirements (and the subsequent resource reservation) based on dynamic customer needs. In other words, we need to make it possible that, given a SLA request for a service that depends on others, this service can produce a near-optimal set of SLA requests towards these other services. From a computational point of view, this is a problem extremely difficult to tackle efficiently. The search space can be vast even for the simplest of cases: Consider the 2-task workflow from the previous paragraph. If given continuous time space, the problem is already resulting for a mere two tasks in countless combinations of response times which add up to N time units. Although this is not practically an issue for most of the cases, it remains true that unbounded search

spaces may occur – especially in the absence of some templating system that will drive initial decisions during planning and negotiation.

Additionally to considering a "super-market" approach when the use case allows, this problem can become tractable only if services always present templates for acceptable options and limits as regards the provided guarantees. Additionally, if an appropriate forecasting mechanism is established based on data from previous negotiations and monitoring data, we believe that a system can produce subsequent SLAs (starting from a single, top-level SLA) with reasonable efficiency. The problem then is transformed and the question becomes how to bootstrap such a system in the absence of historical information. For this, simulation may be an option again.

An additional problem is related to representing the knowledge of service dependencies. Such dependencies may not be obvious, for instance, services existing in stand-by mode for reasons of redundancy. Therefore, a proper generic method to achieve this representation is required. We propose an abstract system that can be trained by domain and use case experts, and which covers at the very least invocation dependencies with timing information (e.g. business processes), and infrastructural requirements (e.g. the Domain Name System for a web server). Given appropriate representation models and a system that can be trained with respective rules, it will be possible to come up automatically with a list of relevant dependencies for a specific negotiation, and take them into account for producing subsequent SLA offers given an initial, top-level offer.

4.3 Monitoring

Monitoring Service Level Agreements is essential to actually confirm the attained QoS as negotiated. Therefore all information necessary to assess the compliance with the contract, e.g. related to the terms specifying the service, has to be collected. This information needs to be up to date and accessible at any point in time. However, it is challenging or impossible to get a consistent view on a large-scale, service-oriented environment. In the case of a network SLA, for instance, it might contain an agreement regarding the minimum bandwidth. However, the actual monitoring of such information can often only be conducted in a time-quantized way which introduces a certain inaccuracy regarding the SLA assessment and makes it difficult or impossible to evaluate whether a SLA requirement was actually fulfilled. To address this topic, we foresee a need for active monitoring agents that intercept service activity continuously, and being aware of SLA properties, they are emitting monitoring events according to the requirements of specific use cases. Such agents should include generic functionality for event emission and therefore their integration to a more complete infrastructure, nevertheless they would also need to be customizable so that they can fit different kinds of services and different SLA requirements.

An additional problem has to do with policy and authorization. Most of the providers would be unwilling to allow customers with access to resource/service monitoring information, although they have a legal interest in that. This is less of a

technical issue, and it is recently being circumvented by specialized companies that act as trusted third parties between service customers and service providers. The service offered by such companies can greatly facilitate the flourishing of SLA-based service infrastructures, as it solves one of the most important practical obstacles to their uptake.

4.4 Adjustment

Using monitoring information, we need to respond to violations as soon as they are detected, so that the costs of these violations are minimized. Assuming a distributed monitoring infrastructure, the main problem is how to aggregate incoming data to detect the violations, especially given the complexity that a SLA may have, and its possible dependencies on other SLAs. Just like the case of SLA planning, this can become a very complex computational problem, given the fact that the decision includes re-negotiation as an option – and that needs to take into account penalties and other business concepts in general. Our proposal in this area is, instead of simply reusing planning algorithms, to facilitate their work by means of adjustment actions priorities. In other words, a service provider should be aware well in advance, not only what kind of adjustment options exist, but also what is most preferred according to its business model and the repercussions of violations on what is most important according to this model.

A different problem, is that the highly domain-specific nature of SLAs magnifies the problem of deciding in a general way whether a violation is relevant or not. For instance, a service may be offering availability guarantees, which are violated during a time period that the customer does not invoke it. In this case, there is a violation but it does not affect the customer (and the customer's own clients, if any). Thus, if a penalty is (in general) applicable for the violation of the SLA, there needs to be a clear picture whether it should be paid, in the case that the customer is not affected. The example above illustrates that the SLAs need to be very detailed with regard to their business value, what penalties apply, and under what conditions they should be paid.

5 Conclusions

This position paper discussed a number of issues related to management of SLAs negotiated automatically by autonomous or semi-autonomous agents, and with a focus on problems introduced or magnified by the introduction of SLA hierarchies. The main problems we see unaddressed are related to modeling SLAs, planning and optimizing them, monitoring them, and adjusting when a violation occurs. For those issues, we made suggestions regarding areas that should be researched and approaches that can be followed. It is clear that some of these problems present

extreme difficulty, nevertheless, we trust that with reasonable compromises on expectations they can be tackled to provide usable frameworks.

Acknowledgments The research leading to these results is supported by the European Community's Seventh Framework Programme (FP7/2007–2013) and the SLA@SOI Integrated Project under grant agreement no.216556.

References

1. K. Nichols, S. Blake, F. Baker, and D. Black. Definition of the differentiated services field (DS Field) in the IPv4 and IPv6 Headers. Technical report, RFC 2474, 12 1998
2. C.H. Chang, P. Kourtessis, and J. Senior. GPON service level agreement based dynamic bandwidth assignment protocol. Electronics Letters, 42(20), 1173–1174, (September 2006)
3. W. Fawaz, B. Daheb, O. Audouin, M. Du-Pond, and G. Pujolle. Service level agreement and provisioning in optical networks. Communications Magazine IEEE, 42(1), 36–43, (Jan 2004)
4. D. Nowak, P. Perry, and J. Murphy. Bandwidth allocation for service level agreement aware Ethernet passive optical networks. Global Telecommunications Conference, 2004. GLOBECOM '04. IEEE. Vol 3. pp. 1953–1957, (2004)
5. A. Sahai, A. Durante, and V. Machiraju. Towards automated SLA management for Web services. Hewlett-Packard Research Report HPL-2001-310 (R. 1), 2001
6. A. Keller, and H. Ludwig. The WSLA framework: Specifying and monitoring service level agreements for web services. Journal of Network and Systems Management, **11**(1), 57–81, (2003)
7. A. Andrieux, K. Czajkowski, A. Dan, K. Keahey, H. Ludwig, T. Nakata, J. Pruyne, J. Rofrano, S. Tuecke, and M. Xu. Web services agreement specification (WS-Agreement). Open Grid Forum, 2007
8. M. Papazoglou, and D. Georgakopoulos. Service-oriented computing. Communications of the ACM **46**(10), 25–28, (2003)
9. J. Cardoso, A. Sheth, J. Miller, J. Arnold, and K. Kochut. Quality of service for workflows and web service processes. Web Semantics: Science, Services and Agents on the World Wide Web, **1**(3), 281–308, (2004)
10. W. Theilmann, R. Yahyapour, and J. Butler. Multi-level SLA management for service-oriented infrastructures. Towards a Service-Based Internet 324–335, (2008)
11. M. Parkin, R. Badia, and J. Matrat. A Comparison of SLA Use in Six of the European Commissions FP6 Projects. Technical Report TR-0129, Institute on Resource Management and Scheduling, CoreGRID – Network of Excellence, 2008
12. The TeleManagement Forum: SLA Management Handbook, Volume 2, Concepts and Principles, Release 2.5. Technical report, The TeleManagement Forum, Morristown, NJ, 2005
13. OASIS: Web Services Resource Framework (WSRF) TC http://www.oasis-open.org/committees/wsrf, accessed December 2008
14. W3C: Web Services Addressing. http://www.w3.org/Submission/ws-addressing/, accessed December 2008
15. R.T. Fielding. Architectural styles and the design of network-based software architectures. PhD thesis, 2000
16. P. McKee, S. Taylora, M. Surridge, R. Lowe, and C. Ragusa. Strategies for the service market place. Grid Economics and Business Models. Volume 4685 of Lecture Notes in Computer Science., Springer, Berlin/Heidelberg pp. 58–70, (2007)
17. M. Parkin, P. Hasselmeyer, B. Koller, and P. Wieder. An SLA Re-negotiation Protocol. Proceedings of 2nd Non Functional Properties and Service Level Agreements in Service Oriented Computing Workshop (NFPSLA-SOC'08), Dublin, Ireland

18. D. Wolpert, and W. Macready. No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation, **1**(1), 67–82, (Apr 1997)
19. D. Wolpert, and W. Macready. Coevolutionary free lunches. IEEE Transactions on Evolutionary Computation, **9**(6) 721–735, (December 2005)
20. S. Fatima, M. Wooldridge, and N. Jennings. A comparative study of game theoretic and evolutionary models of bargaining for software agents. Artificial Intelligence Review, 23(2), 187–205, (2005)
21. C. Figueroa, N. Figueroa, A. Jofre, A. Sahai, Y. Chen, and S. Iyer. A game theoretic framework for SLA negotiation. Technical report, HP Laboratories, 2008

Storage and Analysis Infrastructure for Data Acquisition Systems with High Data Rates

M. Sutter, T. Jejkal, R. Stotzka, V. Hartmann, and M. Hardt

Abstract The purpose of process data processing is the analysis of process data. These data are normally measured using a data acquisition system. Several projects use distributed systems, making the controlling from a central instance with secure access to the infrastructure necessary. Special for several projects are also very high resource demands, e.g. the produced data rates. In order to fulfill the requirements of such projects the development of a data acquisition system is necessary. The main requirements are the secure access to the infrastructure, handling of high data rates and the ad hoc analysis of the measured data. In order to make the system scalable and reliable it should be based on open standards. This chapter evaluates already existing remote control software packages and data management systems for the development of distributed data acquisition systems to fulfill the requirements of process data processing projects. The evaluation identifies that the usage of Grid technologies provides the necessary features as the Grid offers, e.g., Grid security and access to storage and computation infrastructure. The provided architecture uses the GRIDCC framework for controlling the detectors, the communication with data management systems to handle the amount of measured data and the communication with Service Oriented Architectures for ad hoc analysis.

1 Introduction

The purpose of process data processing (PDP) is the analysis of process data, which is normally measured using a data acquisition system. PDP always had special projects with very high demands on computing and storage resources, outbidding the resources available at the corresponding time. Nevertheless, in the near future the requirements of many PDP projects will even grow further up to a point where data processing and data storage on local workstations will not be feasible any more. Examples for demanding such extreme resources are the Karlsruher

M. Sutter (✉)
Karlsruhe Institute of Technology, Institute for Data Processing and Electronics,
Hermann-von-Helmholtz-Platz 1, 76344, Eggenstein-Leopoldshafen, Germany
e-mail: michael.sutter@kit.edu

F. Davoli et al. (eds.), *Remote Instrumentation Services on the e-Infrastructure*,
DOI 10.1007/978-1-4419-5574-6_7,

Tritium Neutrino Experiment (KATRIN) [17], the International Thermonuclear Experimental Reactor (ITER) [16] and Ultrasound Computer Tomography (USCT) [27] at Karlsruhe Institute of Technology. All are using data acquisition systems that measure data at very high data rates.

Another problem in some PDP projects is that the sensors are deployed in a distributed system in an area of several thousand square km, e.g. in the Auger project [2]. For handling of these distributed systems a strict requirement is that all involved sensors are controlled from one central campus and that the measured data is stored in one data sink. The central campus is needed to simplify the controlling of the sensors and the data sink is necessary with regard to the data analysis purposes, allowing the access to the data in a unique manner.

Monitoring–data about the health of the sensors increases the amount of data rapidly, when used in addition within the corresponding project. These data are required for automatic control technologies and the evaluation of the sensors during development and runtime.

For development of a high throughput data acquisition system the mentioned boundary conditions are very important, especially if a requirement is that data acquisition and data analysis should be possible within near real–time. To solve the near real–time requirements a necessary feature is that warranties about the possible data throughput to the data sink can be established and maintained.

The biggest advantage of Grid technologies is that they can fulfill the requirements for PDP, as they allow access to an enormous amount of computing and storage resources. For example the EGEE Grid infrastructure offers about 10 PByte of storage resources, 80,000 CPUs and Gbit networks for communication at the moment [30]. The access to all these resources is not the only advantage of Grid technologies, the integrated security features are another one, prohibiting the unauthorized access to the resources and instruments located within the Grid. Using Grid technologies offers not only advantages; the biggest disadvantage is that the Grid is still under development and scientists willing to use the functionalities are forced to learn a lot about the underlying technologies and how to use them.

2 Fundamentals

This section introduces the requirements for development of data acquisition systems with high data rates and technologies in order to fulfill those requirements. The communication for controlling and monitoring various types of sensors or detectors is possible with remote control software packages. The storage of the measured data can be fulfilled by using a data management system.

2.1 Data Acquisition Techniques

A data acquisition system is a combination of several technologies that measure data of physical properties. The reasons to observe the physical properties are to

analyze the data, to increase the knowledge of the properties and how they interact with other properties [32]. A data acquisition system normally consists of hard- and software components. The hardware components are sensors for measuring the data, e.g. a temperature sensor and are assembled with other electronic components and cables. Usually the analog signals of a sensor are digitized within an analog digital converter (ADC). The software components are for the communication with and controlling of the hardware. This includes starting and stopping of the measurement, retrieving and storing of the measured data, as well as the analysis of the data.

The immediate analysis with near real–time properties is a requirement becoming more and more important for several data acquisition systems. The reasons therefore are on the one hand to use control technologies for the hardware and on the other hand to retrieve the results right after measurement, which is extremely important, e.g. in medicine diagnosis. There it is not possible to wait several hours or even days until the data is processed and results are available. The communication between software and hardware components usually includes the implementation of a specific protocol to access the hardware.

For the measurement of data with data acquisition systems several methods exist:

- Continuous measurement systems begin to measure data directly after startup. Usually they need no interaction from an operator. They run autonomously, even if the measurement takes some weeks up to several months. To prevent the hardware of a continuous measuring system from damage control technologies are used, e.g. the temperature of the data acquisition system is monitored and the system switched off if the temperature reaches a defined maximum value. As such systems produce data permanently a requirement might be to stream the data directly to an appropriate infrastructure.
- Externally triggered systems are typically in a kind of idle mode until an event, normally initialized by an operator, starts the data acquisition. After the measurement is performed the system usually switches back to idle mode, allowing to retrieve the measured data via software. The measurement with externally triggered systems is typically very short – a few seconds up to some minutes, whereas

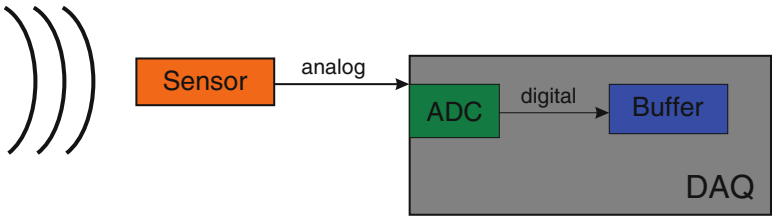


Fig. 1 Schematic design of a data acquisition system. The sensor is measuring the physical properties. The input data is digitalized with an analog digital converter and buffered within the data acquisition system.

the analysis usually needs much longer, depending on the amount and complexity of the data.
- Event based systems are continuously measuring and analyzing data to observe specific kinds of events after startup. As such systems usually have very high data rates specialized trigger algorithms are used to obtain the events and to reduce the data to the interesting parts. The trigger algorithms have a hierarchical structure, whereas the first algorithm (first level trigger) must be very fast. The subsequent algorithms (second level trigger, third level trigger ...) don't have as strict time requirements and can evaluate the data more specifically. The time for evaluation increases according to the position in the trigger hierarchy, whereas the amount of data decreases with every trigger level. An example for a trigger system: A first level trigger is used to digitize and analyze the complete data stream. If a possible event candidate is detected the data is written to a buffer. The following trigger algorithms can analyze the data in more detail, whereas the run is stopped as shortly as possible and as few as possible data is dropped. To fulfill the time requirements the first level trigger is normally implemented in hardware, whereas the second and following trigger algorithms can be implemented in hardware or in software, depending on the requirements of the system.

2.2 Requirements

Section 2.1 described the general methods for measuring with data acquisition systems and the corresponding hardware. Nevertheless, for development of data acquisition systems especially for PDP purposes several requirements must be fulfilled:

- The secure access to the data acquisition hardware, measured data and analysis infrastructure must be possible to ensure that the measured and analyzed data is authentic. Another reason is that only authorized and authenticated users should be allowed to access the system.
- The control of sensors must be possible from one graphical user interface, even if the hardware is at a remote location like a distributed system. The graphical user interface is called virtual control room.
- The analysis of the measured data must be possible with near real–time properties.
- The handling of data of any amount and complexity is needed, as several projects use more or less similar data acquisition hardware, but have different amounts of measured data and also different data formats to store.
- Guarantees of the data throughput capacities must be given in order to solve the near real–time properties.
- Monitoring of the controlled hardware must be possible, including on the one hand the usage of control technologies to prevent the system(s) from damage and on the other hand the storage of the monitoring data for evaluation purposes.

2.3 Remote Control Software

In this section a selection of already existing remote control software packages is described. The packages are chosen, because they are well known in the community and expandable with drivers for new hardware components needed to integrate the communication with custom developed hardware. Another reason is that they are developed especially for data acquisition purposes. The described packages are only an exemplary summary and make no claim to be complete.

- LabVIEW from National Instruments is a development environment for a visual programming language called G, whereas G can be easily used for representation of data flows. LabVIEW is mainly used for data acquisition, controlling of instruments and industrial automation tasks. Every component (e.g. a sensor) is visualized as a block and has a well defined functionality. Different blocks are connected via wires in the graphical user interface, allowing data transfer and build up of data flows [22].

  LabVIEW is a standard product and very popular, including that most hardware manufacturers deliver LabVIEW drivers for their commercial products. The integration of custom hardware using the provided interfaces is also possible, allowing the communication with the hardware devices. LabVIEW is available for Windows, Mac OS and Linux.
- The Grid Enabled Remote Instrumentation with Distributed Control and Computation (GRIDCC) framework [24] was funded by the European Commission. The aim of GRIDCC is to provide access to and control of distributed instrumentation components using Grid technologies, as well as the access to Grid storage and computing resources directly from the sensors. To allow the access to Grid resources physical sensors are abstracted in so called Instrument Elements (IE) [20].

  GRIDCC is written in Java and provides an Application Programming Interface (API) for the development of IEs. This makes the integration of new hardware devices possible and allows the new IEs to use the infrastructure provided by GRIDCC, e.g. a virtual control room for managing different IEs. Since the framework is completely written in Java every Operating System with a suitable Java runtime can host a GRIDCC installation. Linux Operating Systems are used as they provide the whole Grid functionalities and libraries.
- TACO [14] is an object oriented distributed control system originally developed at the European Synchrotron Radiation Facility (ESRF). The aim of TACO is to control accelerators, beamlines and data acquisition systems. All sensors to control are treated as objects in a network space and implemented in separate classes. The whole network communication from the client to the servers and vice versa is handled by the system libraries of TACO and based on SUN's remote procedure calls.

  The enhancement of TACO with new classes can be done in C++, in C (using a methodology called Objects in C), in Python and in LabVIEW. Furthermore TACO is designed portable and runs on Linux, Windows, Solaris and HP-UX.

- TANGO [6] is based on TACO, but CORBA is used for the network communication. The provided object model in TANGO supports methods, attributes and properties. All objects are representations of hardware devices and can be located on a single computer or distributed over the network. To simplify the usage TANGO provides a kernel API, hiding all the details of network access and providing object browsing, discovery and security features. Bindings for commonly used commercial software are available, allowing communication from this software packages with TANGO installations [6].

  New objects can be implemented in C++, Java and Python, whereas a standard installation already provides a large amount of device servers. Linux, UNIX and Windows can be used with TANGO.
- EPICS (Experimental Physics and Industrial Control System) are a set of Open Source software tools, libraries and applications developed in a collaborative manner and used to create distributed soft real–time control systems for scientific instruments, e.g. particle accelerators and telescopes [18]. EPICS uses a client/server model, whereas an elementary dataset is a process variable (PV), accessible on an input/output controller (IOC). The collected measurement and control data from the connected instruments is sent over a network protocol called Channel Access (CA) to the clients. CA defines how the data of PVs is transmitted between client and server. EPICS support soft real–time conditions for the IOCs, for which Linux or Windows can be used. Hard real–time conditions are normally provided by RTEMS [26] or VxWorks [33].

  CA interfaces, needed for the communication with EPICS are available for C, C++, Java, Matlab, Octave, Scilab, Perl and Python. Installations can be done with Linux, Windows, Mac OS, RTEMS and VxWorks.
- ORCA (Object–oriented Real time Control and Acquisition) is a framework designed for the development of modular data acquisition and analysis systems [15]. Applications are distributed via a client/server model, whereas servers provide the data acquisition and analysis features. Clients are responsible for the access to the servers. The communication between clients and servers is implemented using CORBA and the complex details of the communication are hidden in an interface library that exposes only a simple API.

  ORCA is implemented in Objective-C; enhancements must be done also in this programming language. At the moment only Mac OS is supported, a port to Linux is under development.

2.4 Data Management Systems

For the development of a data acquisition system not only the control and monitoring of the involved sensors is necessary. The communication with storage resources is also very important. Therefore a specification already exists, called Storage Resource Manager (SRM) [28]. SRM is integrated in several Grid middlewares like Globus Toolkit 4 (GT 4) [12] and EGEE/gLite [30] and several data management systems. The purpose of the specification is to provide a set of functionalities,

enabling the communication with storage infrastructures from Grid environments. In order to provide access to the storage system an SRM implementation has to provide dynamic space allocation and file management, including e.g. space reservation, file handling, file pinning, protocol negotiation and space quotas [28] – to name just a few ones.

An SRM only provides file management and space allocation, data transport is handled using different protocols. If a file should be transferred a Uniform Resource Locator (URL) is retrieved by the client. Such a URL includes the protocol for data transportation, host name of the server, port and path where the file can be retrieved. Among usuable protocols:

- The GridFTP is based on the File Transfer Protocol (FTP) and is a high-performance, secure, reliable data transfer protocol optimized for high–bandwidth wide–area networks [3]. Therefore GridFTP implements extensions that were either already specified in the FTP specification, but not commonly implemented or that were developed. One of those extensions is the Globus based security for authentication of the users.
- Xroot is a protocol designed for the access to large data sets and optimized for analysis purposes. To allow the access to large data sets it is enhanced with extensions for High Performance Computing. After development started the protocol was soon integrated in ROOT for distributed data access.
- The dCap protocol is for local, trusted access and data transportation, because authentication is not integrated. For the access from outside the GSIdCap protocol can be used, which is the dCap protocol with a GSI authentication [8].
- The Network File System (NFS) is a protocol for accessing files over the network as easily as if the file is stored on local hard disk. Up to NFS version 3 only client computers are authenticated at the server. User authentication is possible since version 4.

Below different data management systems are introduced. Common for all is that they allow the communication with storage infrastructures from Grid environments. Available systems are listed below.

- The Storage Resource Broker (SRB) provides a hierarchical logical namespace and a storage repository abstraction for the organization of data and can be used as a Data Grid Management System [4]. The logical namespace is provided via a Distributed Logical File System, allowing the user to access data even if stored distributed across multiple storage systems. The integration of tape systems and a wide variety of client applications are an advantage of SRB. SRB has been ported to a variety of UNIX platforms including Linux, Mac OS X, AIX, Solaris, SunOS, SGI Irix and to Windows, whereas not the whole functionality is available in Windows, but data can be stored and retrieved.
- The Integrated Rule–Oriented Data System (iRODS) is a Data Grid Management System based on the expertise learned from the development of SRB and more or less the replacement of SRB, even though SRB is still supported [25]. The main

difference is that the management policies and consistency constraints in iRODS are characterized in so called iRODS rules and state information, which are interpreted by the iRODS core to decide how the request and conditions should be resolved. This is also different from other data managements systems as they implement the requests and conditions directly within the software, making the adaptation of the software with every new condition necessary.

- The dCache [11] project provides a system for storage and retrieving, whereas the data can be distributed over a large amount of server nodes. For this purpose dCache offers a single virtual file system tree, which is accessible from several technologies, hiding the physical location of the data from the user. DCache also allows the migration of the data from one server to another, making addition or removal of server nodes possible.
- The CERN Advanced STORage manager (CASTOR) [21] is a hierarchical storage management system and developed at CERN to handle the measured data of different data acquisition systems at CERN. CASTOR provides a UNIX like file system structure over a virtual namespace, where the data can be accessed via command line tools or applications built on top of the data transfer structures.
- The StoRM [10] project is a storage resource manager for disk based storage solutions and provides just the SRM functionality, supporting guaranteed space reservation, direct data access and standard libraries for I/O. StoRM can benefit from high–performance parallel file systems, e.g. General Parallel File System (GPFS) from IBM, but the usage of any POSIX file system, e.g. ext3, is also possible.
- The Gfarm project is a reference implementation of the Grid Datafarm architecture and provides the Gfarm Grid file system [29], which is a virtual file system that integrates disks of compute/file system nodes. The aim of Gfarm is to provide an alternative solution to NFS. The Gfarm file system daemon is running on every integrated node to facilitate remote file operations with access control in the Gfarm file system, as well as file replication, fast invocation and node resource status monitoring.
- Xrootd is a data access daemon for fast and scalable data access, organized as a hierarchical file system–like namespace and based on the concept of directories [31]. The purpose of xrootd is to allow the setup of storage infrastructures with no reasonable size limit and linear scaling capabilities. This means that the performance of the installation should increase with the factor the installation increases.

  Nowadays xrootd is also named Scalla (Structured Cluster Architecture for Low Latency Access) and provides an xrootd server for data access and an olbd server for building scalable xrootd clusters [31].
- The xrootd/SE system is an enhancement of the xrootd system and integrates additional features, e.g. an SRM interface via BeStMan. BeStMan is a full implementation of the SRM v2.2 specification for disk based and mass storage systems. It was designed to work on top of UNIX file system, and has been reported to work on file systems such as NFS, PVFS, AFS, GFS, GPFS, PNFS, HFS+ and Lustre [19].

Table 1 The table shows the integration of SRM, different data transport protocols and communication with tape storage systems for several data management systems

	SRM	GridFTP	xroot	dCap	NFS (Posix)	MSS/HSM backend
SRB	√	√	√			√
iRods	√	√	√			
dCache	√	√	√	√	√	√
CASTOR	√	√	soon		√	√
StoRM	√	√			√	√
Gfarm		√				
xrootd/ Scalla			√		√	√
xrootd/ SE	BeStMan	√	√		√	√
DPM	√	√	√		√	

- The Disk Pool Manager (DPM) [1] is a lightweight solution for disk storage management, mainly developed for the Large Hadron Collider (LHC) experiment of CERN and available within the LHC computing Grid. DPM implements the required SRM interfaces for a storage resource manager and allows the access to the stored data.

3 Use Cases

This section describes the use cases needed to develop a data acquisition system especially for PDP purposes with variable data rates. Common for all use cases is controlling different kinds of instruments remotely from a virtual control room and the usage of strong security features to access the resources. With respect to access the instruments can be deployed on two different kinds of locations:

- Deployed on a local site means that the access is only available from inside the site. Access from outside over the Internet is not possible. The security requirements for accessing the instruments inside a local site are not very hard, as the access to the site requires the security features.
- Deployed on a remote location means that access to the instruments is possible from the Internet, including strict security requirements to deny unauthorized access and to ensure that the measured data is authentic.

As the first entry has nearly no security requirements and the focus is on worldwide distributed systems, the first entry is not covered in this paper. The reasons are on the one hand that security is a strict requirement for worldwide distributed systems and on the other hand that a system developed with security is normally useable without security within a site where security isn't needed.

To develop a remote data acquisition system not only the control of the instruments is needed, but also another architectural detail related to the storage of the

Table 2 The table shows the possible data throughput of the streaming provider between two computers with different security levels and the theoretical amount for 100 Mbit and 1 Gbit Ethernet

	100 Mbit	1 Gbit
Theoretical	12.5 MByte/s	125 MByte/s
Insecure	6–8 MByte/s	100 MByte/s
Signed	3–4 MByte/s	40 MByte/s
Encrypted	1 MByte/s	n/a

measured data is extremely important. If the instrument does not provide sufficient memory or storage space, or if the data has to be analyzed immediately, the recorded data has to be streamed to a second location that provides the required capabilities. Streaming is not a common feature for the remote control software packages mentioned in Section 2.3, as they normally use a more common way: The measured data is buffered locally on the instrument and replicated after or during measurement.

Especially for streaming the measured data a streaming provider was developed at the Institute for Data Processing and Electronics (IPE). This provider is based on GT 4 security and allows streaming of data with two security levels. GT 4 is a widely used Grid middleware, allowing the communication with Grid resources via services, making the development of Service Oriented Architectures based on GT 4 installations possible. The built in security features are based on X.509 certificates, allowing to sign or to encrypt the messages sent to GT 4 servers [12]. The amount of possible data throughput between two computers with the streaming provider was measured and evaluated in table 2.

3.1 Low Data Rates

These kinds of data acquisition systems have no hard requirements. They consist of one or more instruments. Every instrument produces a low data rate of only a few kByte/s. The required network bandwidth for such systems is no problem, as already a 100 Mbit Ethernet offers a theoretical data rate of 12.5 MByte/s. A normal desktop computer offers the capabilities to store and analyze data of this amount. A solution can be to analyze the data within the instrument directly, as introduced in Fig. 2.

An example for such a detector is the MBARI Ocean Observing System (MOOS), consisting of several kinds of detectors with different capabilities, e.g. limited bandwidth [23]. One kind of such detectors are autonomous underwater vehicles (AUV), a kind of robot traveling underwater and measuring data. During measurement the data is stored on the AUV and replicated after appearing via e.g. radio telemetry or satellite communication, which can have a low bandwidth depending on the location of the AUV or when travelling on the borders of the communication range.

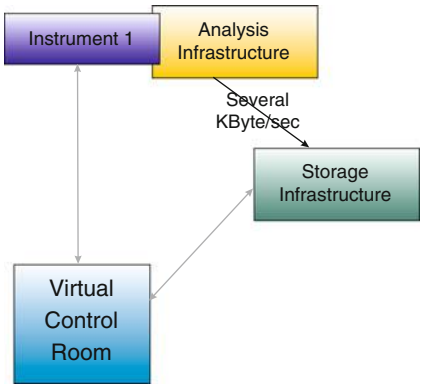


Fig. 2 The schematic design of a data acquisition system with a low data rate of several kByte/s. Visible is the whole infrastructure with virtual control room, instrument, data storage and analysis resource(s). The analysis resources are close to the instruments as the analysis of the data is possible within the instruments

3.2 Medium Data Rates

A data acquisition system with medium data rates specifies sensors that produce data rates of less than 10 MByte/s (see Fig. 3). Modern network components can fulfill the required data rates, e.g. a 1 Gbit Ethernet for the data transportation. The storage of the data on a simple desktop computer is not feasible, especially if multiple instruments are controlled and must store the data on the same machine. To fulfill the storage requirements the usage of data management systems from Section 2.4 are necessary. The analysis infrastructure is another component, which must be increased for immediate data analysis to process the data as fast as possible. Service Oriented Architectures offer the needed capabilities for the analysis, especially adding of resources to solve peak data rates.

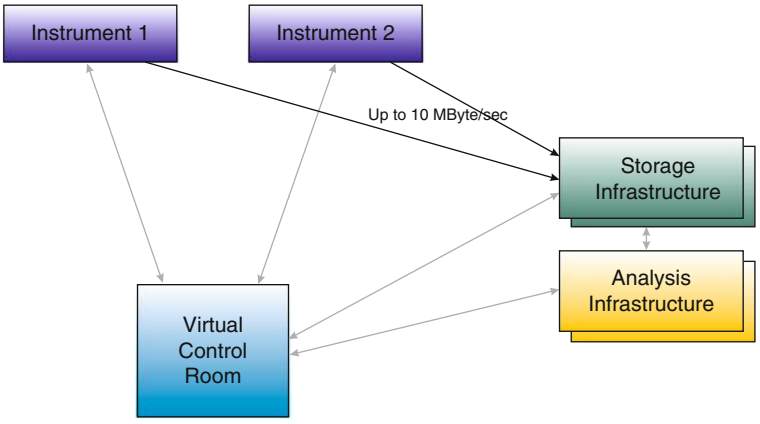


Fig. 3 The schematic design of a data acquisition system with medium data rates of less than 10 MByte/s. Shown is the whole infrastructure with virtual control room, instruments, data storage and analysis resources

The Pierre Auger Observatory in Argentina [2] is an example for such a data acquisition system. The Auger Observatory studies the universe's highest energy particles, which shower down on Earth in the form of cosmic rays. Special for those particles is that they have very high energies of over 10^{20} electron volts (eV) and an estimated arrival rate of just 1 particle per square km and century. The observatory consists of two independent measurement methods; one consists of fluorescence detectors (FD). They use an event based data acquisition technology with several trigger algorithms [2]. Every FD consists of 440 photomultiplier tubes connected to a front–end–crate. The crate has 22 front–end–boards for the first level trigger included, whereas every board is responsible for 22 pixels of the camera. The digitization is done at 10 MHz with a 12 Bit ADC, which is in total a peak data–rate of 9600 MByte/s for the first level trigger. The data rate is minimized by the subsequent trigger algorithms so that the medium data rates use case is applicable.

3.3 Distributed with High Data Rates

Distributed data acquisition systems with high data rates are the supreme discipline for data acquisition systems and at the moment a system for the future. Such systems categorize acquisition hardware that produces data of up to 100 MByte/s and is fully distributed (see Fig. 4). Distribution indicates that the sensors are deployed worldwide, e.g. in different countries and organizational domains and the controlling must be possible from one virtual control room. Such high data rates require an

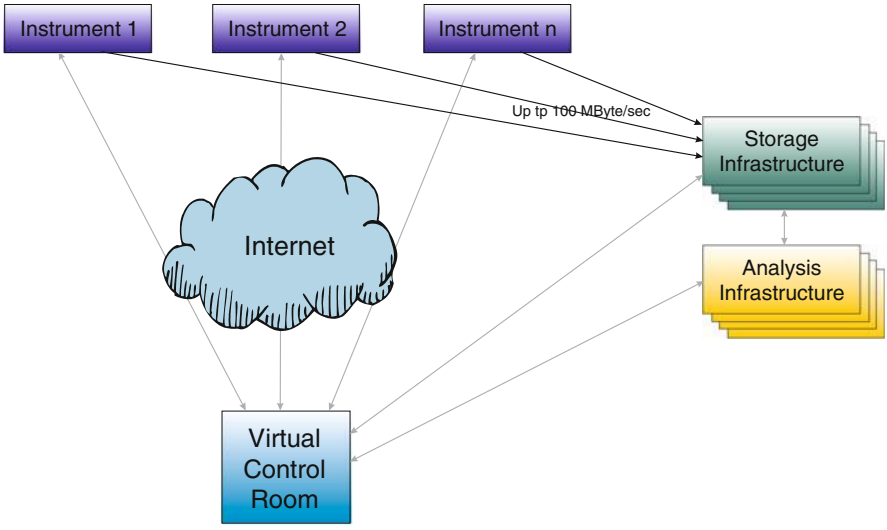


Fig. 4 The schematic design of a distributed data acquisition system with high data rates of up to 100 MByte/s. Visible is the whole infrastructure with virtual control room, instruments, data storage and analysis resources

own infrastructure in order to establish such systems. For communication extremely fast and dedicated network links have to be provided. They directly connect the instruments with the storage resource managers. For avoiding problems with the used networks they have to be strictly evaluated to check if the behavior corresponds to the desired one. For this purpose [13] offers some possibilities, [9] and [7] introduce solutions for measurement over Ethernet in real–time. The required data management system must also be able to handle such amounts of data, especially if several instruments are involved. Also the requirements must be guaranteed for the analysis infrastructure. In order to develop such a system successfully all used components and technologies must be strictly evaluated to check if they can fulfill the requirements and what guarantees for data throughput can be offered.

The low frequency array (LOFAR) is an example for such a system. LOFAR is a new kind of telescope using an array of simple antennas instead of a radio telescope built as a dish antenna. The radio frequencies observed are below 250 MHz and only the deployment as an array can achieve the sensitivity needed to observe the secrets of the early universe, the main aim of LOFAR [5]. The measured analog signals of the antennas for one station are digitized and transported to a central processing unit, where the signals are combined in software to emulate a conventional antenna. As the antennas are very cheap and simple to build, it is possible to deploy a software antenna of 350 km in diameter with about 25,000 antennas. As a first step an array of about 15,000 antennas and a diameter of about 100 km will be deployed. The data acquisition of LOFAR in the first phase consists of a compact core area and 45 remote stations. The data rate of every antenna is about 1.8 Gbit/s, for every station about 220 Gbit/s as the data is reduced by every station.

4 Proposed Architecture

To develop a remote data acquisition system that fulfills the requirements for PDP projects a decision has to be made if one of the described remote control software packages is feasible and what data management system can provide the capabilities for the storage of the data. In order to make a decision the used systems should be based on open standards, to make the developed data acquisition system reliable, scalable and future–proof. If none of the introduced systems is useable the only solution is to develop a new system from scratch, which is not advisable as several components already exist and were developed to fulfill some of the mentioned requirements. For the decision of a remote control software package from Section 2.3 the most important requirements are the security and near real–time features for analysis of the data, including that it is possible to store the measured data simultaneously with the measurement. The introduced software packages are evaluated to check if they can solve these requirements in the following.

LabVIEW is a commercial product and offers security and real–time features in separate modules. The advantage of LabVIEW is the available support of the manufacturer. The biggest disadvantage is that license fees must be paid for every installation. The cost for every installation makes the usage of LabVIEW in

scientific projects nearly impossible, especially for systems that consist of a large amount of distributed components.

ORCA offers neither security nor real–time features in the standard package. The integration of both is too time–consuming and consequently not applicable. Furthermore ORCA is only available for Macintosh computers. A port to Linux is under construction, but not yet available.

The GRIDCC framework has the integrated Grid security, making the system very secure and offering the needed security requirements. The disadvantage is that no real–time requirements are offered and at the moment no measurement of the data throughput is available.

TACO offers hard and soft real–time properties and also security features. The problem with TACO is that the project seems not really active and maintained. So the usage of TACO is not advisable.

TANGO offers some security and real–time features, but both are not strict enough. The security is based on CORBA with SSL for the communication and in this way not useable for a worldwide distributed system as the features are not strong enough for a remote data acquisition system.

EPICS offers the needed real–time requirements as special real–time Operating Systems can be used. The problems are the security features provided. They are not strict enough for developing a remote data acquisition system.

The evaluation pointed out, that only EPICS and GRIDCC offer a part of the features with the required strength. EPICS offers hard real–time requirements via the usage of special real–time Operating Systems for the IOCs, but is missing the security features. GRIDCC offers the security features by using the Grid Security and X.509 certificates, but is missing the real–time requirements.

The decision for the remote control software package is to use the GRIDCC framework. The reasons are that GRIDCC already offers the needed security features and is based on open standards. EPICS misses the security features and provides an own protocol for communication from clients with servers. The near real–time properties for data analysis will be solved via the integration of the communication with dynamic infrastructures, which have to be integrated in GRIDCC and EPICS. The provided real–time features of EPICS are for reading or writing to PVs, a feature solved by the custom developed hardware for which the corresponding data acquisition system is used. The measured data is stored by using a data management system, a feature which also has to be integrated in GRIDCC and EPICS.

The evaluation of the data management systems is much easier. All the introduced systems can fulfill the requirements, as they are developed to handle data of such an amount and are already in production in different projects. So the decision for a system depends on the required additional features, e.g. the communication with some systems is already integrated in Grid middlewares, whereas others offer APIs for different programming languages or the communication with tape storage devices. In regard to both evaluations the development of a new system from scratch is not advisable, as the adaption of the existing components is feasible. Currently the

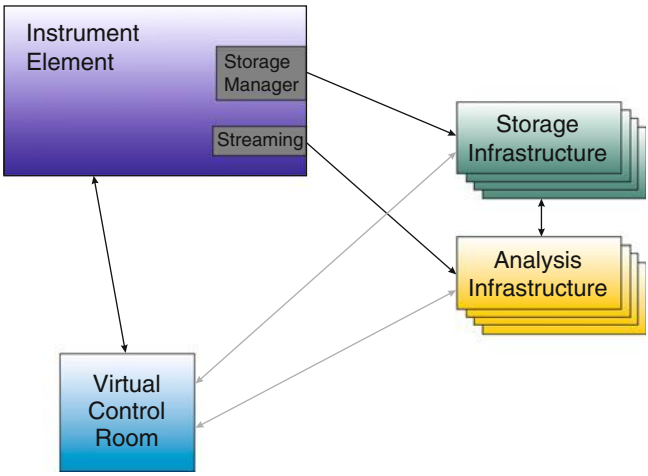


Fig. 5 Schematic design of the IE with integrated communication with a storage resource manager and data streaming in the proposed architecture for a data acquisition system

evaluation of different storage resource managers is still in progress, so it is not clear which system will be used.

Another solution for data storage is the integration of streaming providers introduced in Section 3, allowing e.g. the storage of the data on GT 4 based Service Oriented Architectures. This makes the immediate analysis of data with near real–time properties possible, but only if the bandwidth for data replication and the capacities of Service Oriented Architectures can fulfill the requirements.

Figure 5 shows the proposed architecture with integrated communication with a storage resource manager and streaming provider in the GRIDCC framework. Normally only one of the storage solutions is integrated in the system, as both would overstrain the resources of the instrument(s) and network.

5 Discussion and Future Work

As introduced, none of the software packages offer all needed capabilities by default. Some offer real–time, but no security features. Provision of the needed guarantees for data throughput over the network is very difficult and none of the provided packages offer such capabilities, especially if the systems are deployed in different countries and across organizational domains. Therefore an out of the box usage of any of the packages is not feasible. For every remote control software package different components must be developed or adapted to solve all requirements. One feature, which has to be integrated, is data streaming, as no system offers this functionality for ad hoc data analysis within near real–time over Service Oriented Architectures.

What about the use cases introduced in Section 3? How relevant are they, especially for PDP purposes? The low data rate use case has nearly no relevance for the development of a new system, as the problems of the described architecture, especially the amount of data, have been solved and almost every available product can fulfill the requirements. This is in most cases possible even with standard desktop computers. The average data rate use case has actual relevance, as the Institute for Data Processing and Electronics is currently building such systems – some of them are already in production and others are under development. These systems offer a subset of the described requirements, as all features are not needed by the special projects at the moment. Nevertheless, the development of an all–in–one system offering all capabilities is needed to solve the requirements of our upcoming projects. The distributed use case with high data rates is a system for future work, but becoming more and more important as our institute is building hardware that produces data in the described order of magnitude. At the moment such an amount of data is reduced by several trigger algorithms in hardware, so that data with no information for the project can be discarded. Nevertheless, with the increasing storage, computational and transportation resources the projects are more and more interested in this data and in the near future they want to be able to use the data. This will increase the amount of data rapidly and requires the development of the introduced new system.

The next steps are to make a decision for a data management system and the integration of all features for one project our institute is involved in.

One question still pending belongs to the data throughput requirements. How can they be guaranteed, especially if the instruments are deployed in a remote location where the infrastructure is not as good as e.g. in central Europe?

References

1. L. Abadie, P. Badino, J.-P. Baud, J. Casey, A. Frohner, G. Grosdidier, S. Lemaitre, G. Mccance, R. Mollon, K. Nienartowicz, D. Smith, and P. Tedesco. Grid–Enabled Standards–based Data Management. 24th IEEE Conference on Mass Storage Systems and Technologies, 2007. MSST 2007., pp. 60–71, September 2007
2. J. Abraham, M. Aglietta, I.C. Aguirre, et al. Properties and performance of the prototype instrument for the pierre auger observatory. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 523(1-2),50–95, 2004
3. W. Allcock, J. Bresnahan, R. Kettimuthu, and M. Link. The Globus Striped GridFTP Framework and Server. Supercomputing, 2005. Proceedings of the ACM/IEEE SC 2005 Conference, pp. 54–54, November 2005
4. C. Baru, R. Moore, A. Rajasekar, and M. Wan. The SDSC storage resource broker. CASCON '98: Proceedings of the 1998 conference of the Centre for Advanced Studies on Collaborative research, 5, IBM Press, 1998
5. J. D. Bregman. Towards a concept design for a LOFAR. The Universe at Low Radio Frequencies, Proceedings of IAU Symposium 199, 484, 2002
6. J.-M. Chaize, A. Götz, W.-D. Klotz, J. Meyer, M. Perez, and E. Taurel. TANGO an object oriented control system based on CORBA. ICALEPCS, 1999

7. P. Daponte, D. Grimaldi, and M. Marinov. Real–time measurement and control of an industrial system over a standard network: Implementation of a prototype for educational purposes. IEEE Transactions on Instrumentation and Measurement, 51(5),962–969, October 2002
8. dCache.org. http://www.dcache.org/manuals/Book September 2010. The dCache Book
9. A. Depari, P. Ferrari, A. Flammini, D. Marioli, and A. Taroni. A new instrument for real-time ethernet performance measurement. IEEE Transactions on Instrumentation and Measurement, 57(1),121–127, January 2008
10. F. Donno, A. Ghiselli, L. Magnoni, and R. Zappi. StoRM: GRID middleware for disk resource management. Proceedings of the International CHEP 2004, Interlaken, Switzerland, October 2004
11. M. Ernst, P. Fuhrmann, M. Gasthuber, T. Mkrtchyan, and C. Waldman. dCache, a Distributed Storage Data Caching System. Proceedings of the International CHEP 2001, Beijing, September 2001
12. I. Foster. Globus toolkit version 4: Software for service-oriented systems. IFIP International Conference on Network and Parallel Computing, LNCS 3779, Springer, pp. 2–13, 2005
13. G. Giorgi and C. Narduzzi. Detection of anomalous behaviors in networks from traffic measurements. IEEE Transactions on Instrumentation and Measurement, 57(12),2782–2791, December 2008
14. A. Götz, W.-D. Klotz, P. Mäkijärvi, J. Meyer, E. Taurel, and J. Quick. TACO: An object oriented system for PC's running Linux, Windows/NT, OS-9, LynxOS or VxWorks. The second workshop on PCs and Particle Accelerator Control, Tsukuba, 1999
15. M. A. Howe, G. A. Cox, P. J. Harvey, F. McGirt, K. Rielage, J. F. Wilkerson, and J. M. Wouters. Sudbury neutrino observatory neutral current detector acquisition software overview. IEEE Transactions on Nuclear Science, 51(3),878–883, June 2004
16. ITER Technical Basis. ITER EDA Documentation Series, 24, IAEA Vienna 2002
17. KATRIN Collaboration. KATRIN Design Report 2004. Technical Report FZKA-7090, February 2005
18. R. Lange, A. Johnson, M.R. Kraimer, W.E. Norum, and J. Hill. Epics: Recent developments and future perspectives. 9th International Conference on Accelerator and Large Experimental Physics Control Systems, 278, 2003
19. Lawrence Berkeley National Laboratory Scientific Data Management Research Group. BeStMan, http://datagrid.lbl.gov/bestman/ September 2010
20. F. Lelli, E. Frizziero, M. Gulmini, G. Maron, S. Orlando, A. Petrucci, and S. Squizzato. The many faces of the integration of instruments and the grid. International Journal of Web and Grid Services 2007, 3(3),239–266, 2007
21. G. Lo Presti, O. Barring, A. Earl, R. Garcia Rioja, S. Ponce, G. Taurelli, D. Waldron, and M. Coelho Dos Santos. CASTOR: A distributed storage resource facility for high performance data processing at CERN. 24th IEEE Conference on Mass Storage Systems and Technologies, 2007. MSST 2007., pp. 275–280, September 2007
22. National Instruments. NI LabVIEW - The Software That Powers Virtual Instrumentation - Products and Services - National Instrument, http://www.ni.com/labview/ September 2010
23. T. O'Reilly, D. Edgington, D. Davis, R. Henthorn, M. McCann, T. Meese, W. Radochonski, M. Risi, B. Roman, and R. Schramm. "Smart network" infrastructure for the MBARI ocean observing system. OCEANS, 2001. MTS/IEEE Conference and Exhibition, 2,1276–1282, 2001
24. R. Pugliese, F. Asnicar, L.D. Cano, L. Chittaro, R. Ranon, L.D. Marco, and A. Senerchia. Collaborative Environments For The GRID: The GRIDCC Multipurpose Collaborative Environment, chapter IV. Springer, New York, NY, 2006
25. A. Rajasekar, M. Wan, R. Moore, and W. Schroeder. A prototype rule-based distributed data management system. HPDC workshop on "Next Generation Distributed Data Management", May 2006

26. RTEMS Steering Committee. Welcome to the RTEMS home page, http://www.rtems.com/ September 2010
27. N. Ruiter, M. Zapf, R. Stotzka, et al. First Images with a 3D–Prototype for Ultrasound Computer Tomography. IEEE International Ultrasonics Symposium, 2005
28. A. Shoshani, A. Sim, and J. Gu. Storage resource managers. Grid Resource Management: State of the Art and Future Trends 1st edition, chapter 20 , pp. 321–340. Springer, 2003
29. O. Tatebe, Y. Morita, S. Matsuoka, N. Soda, and S. Sekiguchi. Grid datafarm architecture for petascale data intensive computing. 2nd IEEE/ACM International Symposium on, Cluster Computing and the Grid, 2002. May 2002
30. The EGEE project. EGEE: Enabling Grids for E-sciencE; http://www.eu-egee.org/ September 2010
31. The xrootd team. The Scalla Software Suite: xrootd/cmsd, http://xrootd.slac.stanford.edu/ September 2010
32. Wikipedia – The Free Encyclopedia. Welcome to Wikipedia, http://en.wikipedia.org September 2010
33. Wind River Systems. Wind River; http://www.windriver.com/ September 2010

GridWalker: A Visual Tool for Supporting Advanced Discovery of Grid Resources

Alessio Merlo, Daniele D'Agostino, Vittoria Gianuzzi, Andrea Clematis, and Angelo Corana

Abstract Grid management has always been a complex issue, especially when Grids are composed of a large number of resources and the efficiency is a key aspect (e.g to support applications with QoS requirements). For this, Grid administration becomes increasingly important, and advanced instrumentation for Grid management is needed. In this chapter we present GridWalker, a visual application that allows the efficient management of information about the resource characteristics and their updated status, in order to foster a proper allocation of user jobs on available resources.

1 Introduction

Grid platforms are distributed architectures that allow to share potentially any number and kind of resources, no matter their geographic location. Using suitable middleware services, these resources are made available in a transparent and secure manner to user communities which constitute Virtual Organizations (VOs). The development of new middleware services makes Grid platforms increasingly used not only in scientific environment but also in industrial and business applications.

In this context, it is a trivial consideration to point out that the performance of a Grid Computing platform as a resource provider depends strictly on the efficiency of the resource discovery services, supplied by Grid middleware. Such services have to grant that, whenever a resource required by a user is available, it is found and provided to the user.

In a Grid environment, Virtual Organizations are composed by a set of Physical Organizations (POs), in which resources are offered and proper mechanisms for their discovery are needed. Basic services for resource discovering are provided by Grid middleware like Monitoring and Discovering Service (MDS) of Globus Toolkit [1], but they are in most cases rather limited, and unable to meet the user requirements.

A. Merlo (✉)
IEIIT-CNR, Via De Marini 6, 16149, Genova, Italy
e-mail: alessio@ieiit.cnr.it

F. Davoli et al. (eds.), *Remote Instrumentation Services on the e-Infrastructure*,
DOI 10.1007/978-1-4419-5574-6_8,

Moreover, the growing demand of Quality of Service (QoS) from novel Grid applications (e.g. soft real-time applications, remote instrumentation access, urgent computing, Massive Multiplayer On-line Games (MMOG) [2]) requires improved performance from such services like discovery, that has to take into account a larger number of parameters (including the QoS ones), and has to complete within a maximum time bound [3].

For these reasons, the offered functionalities and the complexity of discovery services increased during years; hybrid architectures like the ones based on the peer-to-peer (P2P) [4] or super-peer paradigms [5] have been implemented on Grids for supporting improved discovery activities, especially on very large Grids, with the aim of making available *advanced discovery services* [6]. It is worth noting that, to grant a satisfying discovery activity, it is important to have systems able to manage efficiently both discovery activities within a single PO and among different POs.

In general, the efficiency of the advanced discovery architectures depends strictly on the definition and maintenance of an *overlay network* that operates over Grid middleware level and among the various POs to improve the discovery performance. Generally, strong assumptions are made on the overlay network (e.g. the network must form in a automatically way, and it must be completely connected and fault tolerant), but tools for their building and management are not investigated. Furthermore, the mechanisms for merging information among the various index services at the level of single PO are not deeply studied.

To solve this open issue we propose GridWalker, a visual tool for supporting both the organization of index services within a single PO by the PO administrator, and the management of the overlay network among POs by the Grid administrator.

The chapter is organized as follows: in Section 2 the state of art on index service and advanced discovery is presented; Section presents the tool GridWalker; in particular, Section 3.1 describes the functionalities of the tool for managing local index services within a single PO on Globus Toolkit 4 (GT4) middleware, while Section 3.2 shows the functionalities for managing the overlay network among POs. Section 4 concludes the chapter.

2 Related Work

Globus Toolkit 4 [1] is the current *de facto* standard for Service Oriented Grids. Resources are mapped to Grid Services and information on their state is provided through proper Monitoring and Discovering Services (MDSs) to which the Grid Services are registered. MDSs are organized hierarchically, and the retrieving of resources within a single PO depends strictly on the way such MDS are connected.

The behavior and efficiency of sets of connected MDSs have been investigated deeply [7]. The results of such analysis underlined that MDS systems suffer from some critical drawbacks. First, the MDSs system is not fault tolerant, meaning that the fault of a MDS node causes the inability to retrieve information managed by nodes lying at lower levels in the hierarchy, unless they are connected to other higher level MDSs. Moreover, information on resources is exchanged among MDSs,

resulting in a replication. Such replication makes the MDS system hardly scalable. Finally, the distribution of information implies delays, creating a freshness problem (i.e. the obtained information can be obsolete).

Globus Toolkit does not provide a resource matchmaking/brokering as a core service, but the GridWay metascheduler [8] was included as an optional high-level service since June 2007. GridWay provides dynamic scheduling, performance monitoring, opportunistic and on request job migration, and some fault recovery mechanisms. However, users can specify only a fixed and limited set of resource requirements, most of them related to the queue policies of the underlying batch job systems. As a consequence, GridWay can be inadequate to fulfil complex user/application requirements.

Another tool for resource discovering and matchmaking is gLite [9], that takes into account a richer set of requirements, including benchmark values for a better characterization of resource performance. However, this service is based on a semi-centralized approach, resulting in long waiting time for job execution and limited scalability.

For advanced discovery purposes, it is clear that new mechanisms need to be designed to improve the MDS system efficiency and to overcome the previous limitations on single POs. Some general criteria must be followed to provide a suitable and effective tool for advanced discovery of Grid resources: management of both syntactic and performance requirements; independency of the underlying middleware, to assure interoperability; a distributed structure, to support a good scalability; a suitable graphical interface, to allow ease of use.

In [10, 11] we proposed GEDA (Grid Explorer for Distributed Applications), a tool for providing advanced discovery services, through a proper organization of retrieved information, and refreshing and monitoring activities on controlled MDSs. GEDA, implemented as Grid service, allows the discovery and selection of resources and the execution of sequential/parallel jobs on single machines, homogeneous or heterogeneous clusters; moreover, it allows reservations on resources. GEDA provides a structured and updated view of resources, allowing a more easy and efficient use of services available from the underlying middleware. Computational resources are divided into three classes, namely Single Machine (SM), Homogeneous Cluster (HoC) and Heterogeneous Cluster (HeC). The single machine is the base unit, with a set of resources such as: node name, IP address, processor model and clock, number of available processors, total and free RAM, total and free disk space, type and release of O.S. A homogeneous cluster is a set of machines having the same properties, whereas a heterogeneous cluster is a set of machines with different characteristics, that can be convenient to organize into a logical structure.

Regarding the overlay network, many proposals have been made to build a connected and scalable overlay network based on P2P architectures [7] or Voronoi/Delaunay approaches [12]. Under an overlay network point of view, P2P architectures are divided into structured and unstructured ones [13]. In structured P2P there is a strict organization of the interconnections of the overlay network and of the information on resources managed and exchanged between peers; discovery

is supported by proper Distributed Hash Tables (DHTs). In unstructured P2P, peers are randomly connected to a fixed number of other peers, and there is no guarantee that the overlay network is connected. For this, it is useful to connect peers through a Delaunay triangulation that provides such guarantee. The main drawbacks of P2P architectures are that structured ones have huge maintenance cost, while unstructured ones could suffer from partitioning.

3 The GridWalker Tool

GridWalker is a visual tool, developed in Java, able to provide support for managing both the index services on resources within a single PO (by the PO administrator), and the overlay network inter-POs (by the Grid administrator) in a Virtual Organization. GridWalker operates over Globus Toolkit middleware, working on MDSs for managing the resource information within the PO, and relying over a set of GEDA [10], one for each PO, that acts as super-peer of the overlay network.

3.1 GridWalker as a PO Resource Manager

GridWalker provides support for managing index services of the Grid middleware. Although in this paper we show how GridWalker works over Globus Toolkit 4 and its index services, namely the Monitoring and Discovery Services (MDSs), we point out that GridWalker could easily be adapted to other Grid middleware index services on a Service Oriented Grid.

On Globus Toolkit 4, MDSs are organized hierarchically, so that if MDS A (father) is connected to MDS B (son), MDS A is aware of the services published by MDS B; in this sense, a query to MDS A provides information also on services registered to MDS B. In this context, full freedom is left to the PO administrator that could organize MDSs according to his needs, through a direct editing of *hierarchy.xml* configuration file on every instance of Globus Toolkit installed on single machines or clusters. Such operation is hardly scalable, and manual organization of a large number of MDSs is tricky; moreover, the choice on how to connect MDSs is strategic, owing the issues we introduced above. Finally, it is difficult to manually check every node in the hierarchy for failure detection purposes.

Previous issues can be managed by GridWalker on a two-level MDS hierarchy. Low-level MDSs are MDSs of single machines, whose scope is limited to resources directly registered to them. Up-level MDSs are index services without own services registered; they act as a central index service to which low level MDSs are hierarchically connected. Every low-level MDS is registered to all up-level MDSs. The number of up-level MDSs is greater than one for fault tolerance purposes. Up-level MDSs are connected to a GEDA service [10] which regroups, parses and organizes information for GridWalker, managing failures of up-level MDSs (Fig. 1).

On such scenario, GridWalker, interacting with the GEDA service, provides a visual interface for: (1) building the hierarchy graph among MDSs; (2) monitoring the hierarchy, so that failures on MDSs can be detected; (3) allowing the visual management of the hierarchy (e.g. in case of failure of some MDSs); (4) monitoring single resources registered to a given MDS (Fig. 2).

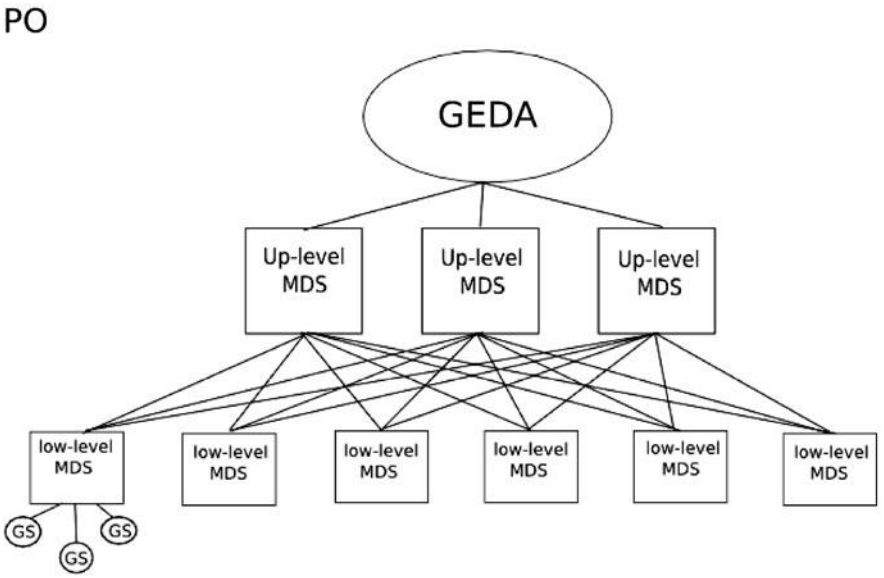


Fig. 1 MDSs organization within a PO

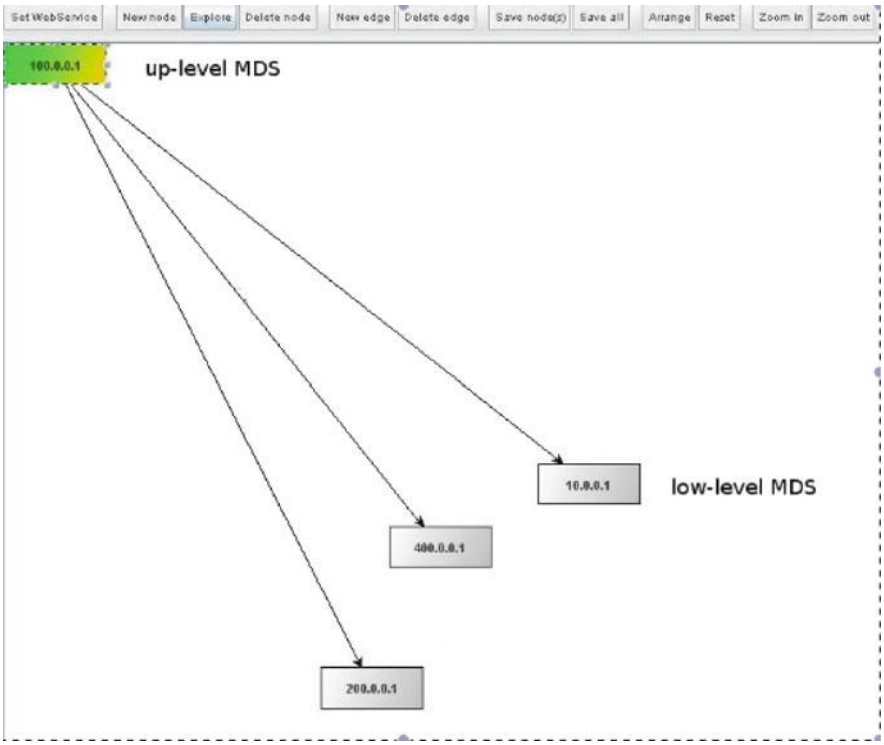


Fig. 2 GridWalker: management of MDSs within the PO

3.2 GridWalker as an Overlay Network Manager

As described in the previous Subsection, each instance of GEDA represents all and only the resources of a Physical Organization. Therefore, the union of the various GEDA instances represents all the resources of a Virtual Organization. The design of the interaction mechanism among the various GEDAs is a major issue in order to provide an efficient discovery of resources that best suit user requirements.

The typical centralized or hierarchical approaches used in most Grids are inefficient for many reasons. At first, in a large Grid we have to tackle the issues related to propagation delays and possible bottlenecks represented by the nodes at the highest level of the information system hierarchy. In both cases this results in possible heavy overheads. In EGEE, for example, where a semi-centralized approach is implemented, half of the submitted jobs wait for more than 5 min before their execution [14]. Moreover, Grid resources are dynamic entities, and the fault of a node may make unavailable all the resources it represents.

As remarked, alternative approaches that proved their effectiveness for the distributed resource discovery in Grid environments are based on P2P technologies [4], which allow to tackle dynamicity and scalability issues. So, we adopted an unstructured P2P approach to manage the interactions among the various GEDAs (Fig. 3).

The GEDA instances know each other by registering themselves using a proper Grid Service called PO Registration Service (PORS). Every GEDA registers itself during the start-up phase, and it receives back the addresses of its neighbors. The PORS organizes GEDAs as a Delaunay triangulation by associating to each instance two random coordinates within a fixed rectangle. This solution gives a degree of six for the triangulation vertices and a good load balancing. The creation of the triangulation is performed in an incremental way, as described in [12]. Periodically the

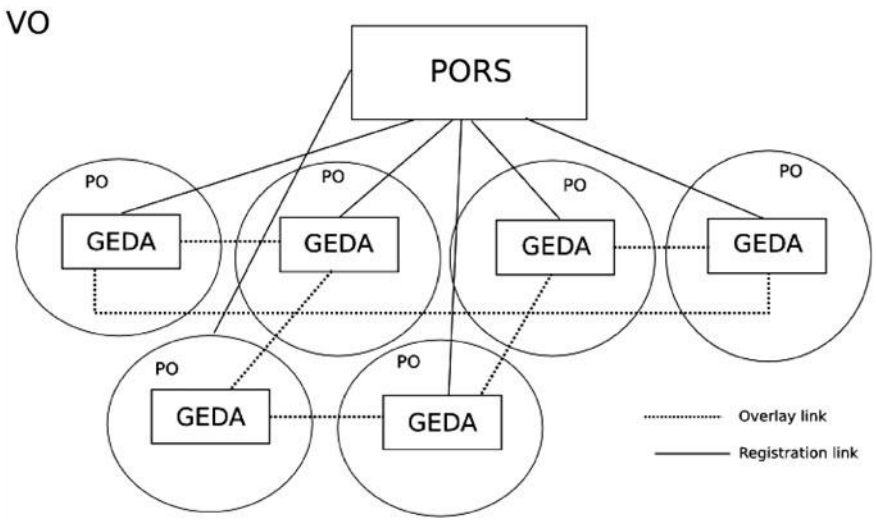


Fig. 3 Overlay network over POs

PORS checks the state of the registered GEDAs, in order to update the triangulation if some instances are no more available. In this case, it sends to the GEDAs involved in the re-triangulation the updated list of the neighbors.

This list is important for the selection of the resources according to user requests. We avoided solutions based on flooding or too complex routing mechanisms. In fact, we have to consider that the parameters that a user may specify for its job are fixed by the query language. For this reason each GEDA processes the information about the resources it represents and produces a single table for the whole PO with entries of type <*attribute, minvalue, maxvalue*>. For example, CPU_NUMBER represents the range of the CPUs available within a single cluster (e.g. its range may vary from 16 to 512).

Each GEDA makes available its table to the first neighbors and it receives theirs. When it is queried by a user or by another GEDA, at first it searches within its resources. If it does not find any possible candidate it forward the query in a random way to one of the neighbors satisfying the requirements. If none of them has the required resource a further step occurs in which each node forwards the request to one of the neighbors (except possibly the one that forwarded the request) randomly. Such mechanism is used in SoRTGrid [3] overlay network among Facilitator Agents, to perform timed-constrained advanced discovery.

Figure 4 shows how our tool allows to manage in a graphical way the overlay network, composed of the various GEDA instances, one for each PO.

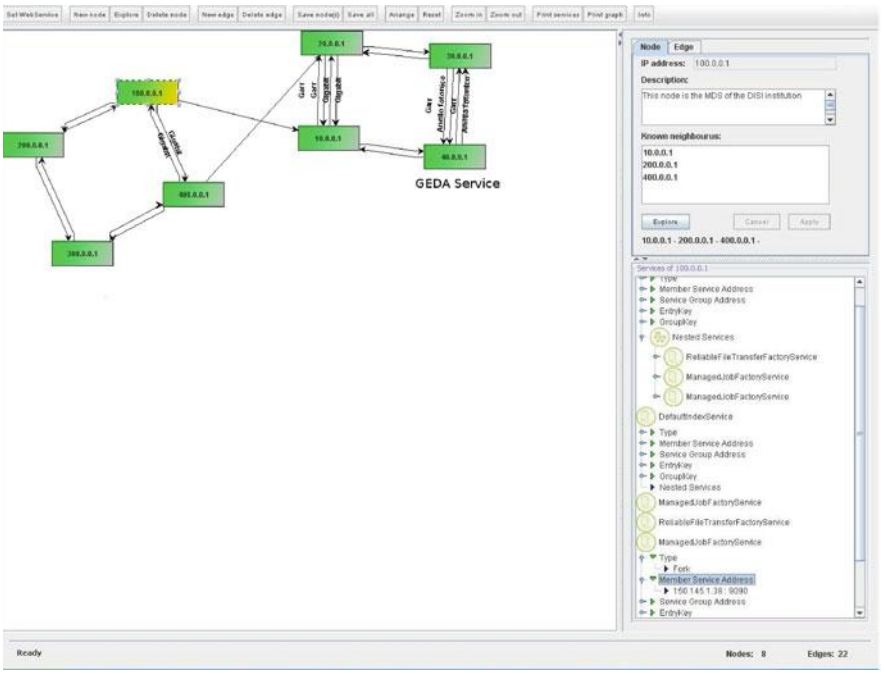


Fig. 4 GridWalker: management of GEDAs within the VO

The underlying architecture allows a very good scalability, as each instance of GEDA at the PO level only manages local resources, and the efficient interaction among the various GEDA instances is assured by the overlay network, based on a super-peer approach and on a Delaunay triangulation.

4 Conclusions

GridWalker supplies advanced support for managing local index service structures and overlay networks. At the single PO level, it can work on every disposition of index services, providing functionalities to save and restore MDSs hierarchies. Moreover, it provides the Grid administrator with non-native Grid middleware control functionalities like fault discovery and higher level management of both local index services connections and overlay networks.

GridWalker is designed to be easily ported to different Service Oriented Grid middleware like future versions of Globus Toolkit. Its distributed architecture assures scalability, as large Grids are partitioned into a number of POs, each with a GEDA instance that manages resources at the PO level. The visual features of GridWalker make it simple and user-friendly.

5 Table of Acronyms

– DHT – Distributed Hash Table
– GEDA – Grid Explorer for Distributed Applications
– GT4 – Globus Toolkit 4
– MDS – Monitoring and Discovering Service
– MMOG – Massive Multiplayer Online Game
– P2P – Peer-to-Peer
– PO – Physical Organization
– PORS – Physical Organization Registration Service
– QoS – Quality of Service
– VO – Virtual Organization

References

1. I. Foster. Globus toolkit version 4: Software for service oriented systems. Journal of Computer Science and Technology, 21-4:513–520, July 2006
2. S. Gorlatch, A. Ploss F. Glinka, J. Muller-Iden, R. Prodan, V. Nae, and T. Fahringer. A grid environment for real-time multiplayer online games. Technical Report CoreGRID Technical Report - TR-0133, May 15, 2008
3. A. Merlo, A. Clematis, A. Corana, D. D'Agostino, V. Gianuzzi, and A. Quarati. Sortgrid: A grid framework compliant with soft real-time requirements. Proceedings of 3rd International Workshop on Distributed Cooperative Laboratories: Instrumenting the Grid (INGRID 2008), 2008

4. P. Trunfio, D. Talia, H. Papadakis, P. Fragopoulou, M. Mordacchini, M. Pennanen, K. Popov, V. Vlassov, and S. Haridi. Peer-to-peer resource discovery in grids: models and systems. Future Generation Computer Systems, 23(7):864–878, 2007
5. C. Mastroianni, D. Talia, and O. Verta. A super-peer model for resource discovery services in large-scale grids. Future Generation Computer Systems, 2, pp. 1235–1248, 2005
6. R.J. Al-Ali, O.F. Rana, D.W. Walker, S. Jha, and S. Sohail. G-qosm: Grid services discovery using qos properties. Computing and Informatics, 21, 4:363–382, 2002
7. A. Di Stefano and C. Santoro. A peer-to-peer decentralized strategy for resource management in computational grids: Research articles. Concurrency and Computation : Practice and Experience, 19(9):1271–1286, 2007
8. E. Huedo, R.S. Montero, and I.M. Llorente. A framework for adaptive scheduling and execution on grids. Software – Practice & Experience, pp. 631–651
9. Glite Middleware. http://glite.web.cern.ch/glite/.
10. A. Clematis, A. Corana, A. Merlo, V. Gianuzzi, and D. D'Agostino. Resource selection and application execution in a grid: A migration experience from gt2 to gt4. Lecture Notes in Computer Science n. 4276, 'On the Move to Meaningful Internet Systems 2006: CoopIS, DOA, GADA, and ODBASE', R. Meersman, Z. Tari et al. (Eds.), pp. 1132–1142, Springer, Berlin, 2006
11. A. Clematis, A. Corana, D. D'Agostino, V. Gianuzzi, A. Merlo, and A. Quarati. A distributed approach for structured resource discovery on grid. Proceedings of International Conference on Complex, Intelligent and Software Intensive Systems, Barcelona, IEEE Computer Society, pp. 117–125, 2008
12. V. Gianuzzi, A. Merlo, A. Clematis, and D. D'Agostino. Managing networks of mobiles entities using the hyvonne p2p architecture. 3PGIC Workshop at CISIS International Conference 2008 on Complex, Intelligent and Software Intensive Systems, Barcelona, Spain, March 4–7, 2008.
13. P. Trunfio, D. Talia, H. Papadakis, P. Fragopoulou, M. Mordacchini, M. Pennanen, K. Popov, V. Vlassov, and S. Haridi. Peer-to-peer resource discovery in grids: models and systems. Future Generation Computer Systems, 23(7):864–878, 2007
14. T. Glatard, D. Lingrand, J. Montagnat, and M. Riveill. Impact of the execution context on grid job performances. CCGRID '07: Proceedings of the 7th IEEE International Symposium on Cluster Computing and the Grid, pp. 713–718, Washington, DC, IEEE Computer Society, 2007

A Bio-Inspired Scheduling Algorithm for Grid Environments

Antonella Di Stefano and Giovanni Morana

Abstract The design of an effective scheduling policy represents one of the open issues in the field of grid computing research. The dynamism and the heterogeneity of grids, in fact, make difficult the creation of a scheduler able to satisfy, at the same time, all the needs required by these complex environments.

The scientific literature has proposed several solutions based on meta-heuristics techniques: these approaches, in fact, have demonstrated to be able to solve many optimization problems, as the grid scheduling one, adopting behaviors inspired by nature. In this chapter, the authors discuss the implementation of the Aliened Ant Algorithm, a new technique that, forcing the adoption of a "non natural" behavior, exploits the self-organization ability of an ant colony to obtain an effective scheduling policy for a multi-broker grid environment.

1 Introduction

In the last years, the Grid computing [1–3] has affirmed itself as one of the most applied supports in the field of computer-aided scientific research and, even today, many studies are led in order to increase the performance of such systems. A considerable effort, for example, has been made to create Grid middlewares [4–6] capable of guaranteeing robustness and security resources management.

The correct utilization of resources (like CPU load, memory usage, and available network bandwidth) and their optimal exploitation (through of a valid jobs scheduling policy) remain a critical issue in Grids.

The issues related to job scheduling in a grid environment have many particular features that differentiate it to other traditional schedulers. In particular, the presence of different administrative domains composing the Grid Virtual Organization[1] (VO)

G. Morana (✉)
Department of Information and Telecommunication Engineering, Catania Univerisity, Catania, Italy
e-mail: gmorana@diit.unict.it

[1] Scalable groups of research organizations without a particular geographic characterization.

F. Davoli et al. (eds.), *Remote Instrumentation Services on the e-Infrastructure*,
DOI 10.1007/978-1-4419-5574-6_9,

[2], the heterogeneity of the resources and the fluctuations of workload, render the scheduling a NP-hard problem. In this chapter, an innovative heuristic algorithm, inspired by the behavior of ants, is proposed as a solution of the distributed job allocation in a Grid environment. A well-known class of algorithms, called Ant Colony Optimization (ACO[7–10]), bases its functioning on the emulation of the ability of real ants to exploit their self-organization when they are looking for food. The solution given here differs from the ACO in that it changes the search strategy of the ants: our ants are alienated in that they stay away from the others and prefer to follow the paths that were covered by no one else or, at most, by few other ants.

Exploiting this "non natural" behavior, it is possible to obtain a scheduling algorithm characterized by a good load balancing ability and, above all, by a high job throughput value attributable to the short amount of time spent by each job in the waiting queue. The proposed algorithm has been tested and evaluated through extensive simulations on a real Grid infrastructure, currently shared between several Sicilian Research Organizations participating in the PI2S2 project [11]. Experimental results show that it satisfies expectations.

The rest of the chapter is organized as follows. Section 2 presents the related work. Section 3 discusses and compares the Ant Colony Optimization and the Alienated Ant Algorithm. Section 4 explains how the Alienated Ant Algorithm can be applied on a multi-broker grid. Section 5 presents the simulation results. Finally, Section 6 gives a brief conclusion of this work.

2 Related Work

The Ant Colony Optimization[7–10, 12] is a population-based stochastic metaheuristic approach that, simulating the cooperative behavior of the ants, is able to solve complex combinatorial and multi-constraint optimization problems. In the literature there are several studies that propose ACO based solution to solve combinatorial optimization problems like the Quadratic Assignment Problem(QAP)[13], Traveling salesman problem (TSP [14]) or sequential ordering problem (SOP [15]). The "ants" have demonstrated their ability also to face complex applicative scenarios, like Internet and telecommunication routing [18, 16], traffic management techniques [17] and, in general, a wide series of assignment problems.

The ACO approach has been applied with success also to scheduling problems, in many different applicative scenarios. The ACO has been applied to the single machine weighted tardiness (SMWT) problem [18], to the flow shop scheduling problem [19] and to the group scheduling problem [20].

Moreover, several ACO based solutions have been proposed in grids, where it is focused the scheduling algorithm proposed in this paper. An original implementation of an ACO algorithm for Grid Computing is introduced by Fidanova end Durchova in [21]. In that paper, the authors handle the job-machine issue using a graph-based solution. The novelty of this solution consists in the use of multiple nodes for a single machine instead of using the one-to-one standard representation (also used in the proposed algorithm). This allows an increment of the dimension

of the search space given to ACO algorithms, providing the opportunity to explore a greater number of possible solutions. However, this algorithm is based on a static vision of the grid environment: the graph used for the simulations, in fact, is set up just once, considering a fixed number of machines and jobs to schedule. In the real world this approach is limited if one considers that one of the fundamental characteristics of the grid is the dynamism of the resources.

In [22] an algorithm is proposed that maps the ACO solution on the grid in a similar way to the one here proposed; it considers every resource as a node of a graph, the single job as ant and the decisions to be taken as possible routes (each of which corresponds to a resource). This algorithm takes into account both the traffic on the resources and the characteristics of the computational resources. The main difference with the proposed approach regards the mechanism used for the release of pheromone: it is laid down only if the job executes successfully.

The proposal of this chapter is to introduce a "freely" ACO inspired job scheduling for a multi-broker grid environment. This algorithm differs from the other ACO approaches in that it is based on an inverse interpretation of pheromone trails. As will be shown in the follow, the algorithm performs a scheduling policy that assures to each job a short queue waiting time, provides a good load balance over all available resources and is able to adjust itself dynamically onto changes in the job workload.

3 A Comparison Between the Ant Colony Optimization and the Alienated Ant Algorithm Approaches

The functioning of the Ant Colony Optimization is inspired by the ability of real ants to cooperate together in order to solve the issues related to the colony's survival. The most suitable example of this self organizational capability is represented by the search for food and its storage: although blind, the ants are able to coordinate their movements and decisions exchanging knowledge about the path from the food source to their anthill. This is possible because each ant, in looking for food, releases a quantity of a chemical substance (called *pheromone*) proportional to the quantity, quality and proximity of the food, marking the path it has taken. An isolated ant moves itself essentially randomly but, when it smells a pheromone trail, it can decide to follow that new path reinforcing the trail with an additional quantity of pheromone. The greater the pheromone quantity in a path, the greater the probability that an ant will follow it. The repetition of this mechanism represents an auto-catalytic behavior: the higher the number of ants that follow a trail, the more attractive the trail becomes; after a certain number of iterations, all ants will follow the path with the highest concentration of pheromone, identifying it as the shortest path from the anthill to the food source.

The main aspects of the ACO technique are the "attractive" power of the pheromone and its capacity to lead the ants to identify quickly a near-optimal solution for a wide range of generic problems (see Section 2 for more details). But,

what happens if the pheromone is not used as an attractive substance but as a repulsive one?

What is the behavior of an ant colony that instead of using the pheromone to coordinate themselves, uses the pheromone in order to stay away from each other?

In this chapter the authors investigate the "non natural" behavior of these "alienated" ants in order to understand if it is possible to take advantage from it. Each alienated ant stays away from the others and prefers to follow the paths that were covered by no one else or, at most, by few other ants.

In order to achieve this, the ant smells the pheromone trails and takes the least marked path instead of following the path where the pheromone trail is stronger (as the other normal ants would do). The alienated ant interprets a tenuous pheromone trail on a path either as a route few other ants have taken or where a large amount of time has elapsed since the path has last been used. In the following, the "non natural" behavior here depicted will be called Alienated Ant Algorithm (AAA).

Both ACO and AAA approaches uses *indirect communication* based on the pheromone trails to share knowledge on the available food but, while the ACO strategy aims to realize a unique flow of ants that converge to the same food source, the AAA one tries to find the paths to food which the other ones have discarded.

These strategies reflect technical scenarios where intelligent entities (the equivalent of the ants) explore, in parallel and iteratively, the search space looking for solutions to the problems. Each generic problem can be considered to be composed by a certain number of sub-problems. Each ant, for each sub-problem, has to evaluate a set of feasible solutions. For each solution, a pheromone trail, proportional to the times that the solution has been considered, is given. To build a solution, each ant (both normal and alienated) must find a solution for each sub-problem. To choose a solution, both the normal and the alienated ants evaluate the related pheromone trails. The choices of these entities/ants are driven by a probabilistic evaluation of the partial knowledge (trails of pheromone smelt) of the set of feasible solutions. Each entity/ant influences the others by leaving a pheromone trail, transmitting its knowledge related to search space.

Both algorithms follow the same schema in terms of executed steps and differentiate themselves because they adopt an opposite selection technique for the candidate solution.

In general, for each algorithm four different phases can be considered. The first one is the *solution construction phase*, where the ants build the set of the feasible solutions for the given problem. From this set, each ant extracts a solution basing on a specific policy (*solution selection phase*).

Then, each ant updates the *pheromone trails' vector*, releasing on the relevant selected path (vector entry) a certain quantity of pheromone to mark its choice (*pheromone trail update phase*). Finally, the value of each entry of the *pheromone trails' vector* is decreased by a certain quantity in order to simulate the real trail evaporation (*pheromone trails evaporation phase*).

Although there are many similarities between the algorithms, the differences generated by the different selection mechanisms, reinforced by the iteration, render the behavior of the algorithms very different. As said above, to build a solution, both

normal and alienated ant must find a solution for each sub-problem. To choose a solution, both the normal and the alienated ants evaluate the related pheromone trails.

For each sub-problem, the "normal" ant selects, with a high degree of probability, the trail which has a higher pheromone level. After that, the ant marks every one of its choices, increasing the related pheromone quantity. This is the most important step: increasing the pheromone quantity of a given trail, i.e. solution, increases the probability that it will be chosen by other ants and consequently the pheromone quantity will growth larger. At the same time, the evaporation mechanism reduces the pheromone quantity, and consequently the probability of it being chosen, over the other solutions. Iteratively, all the ants will converge to the same general solution.

The alienated ant, for each sub-problem, selects with an high probability the trail, i.e. solution, having the lower pheromone level. As the normal ant, the alienated ant marks its choice by increasing its pheromone quantity: this mechanism, however, reduces the probability that the same solution will be selected by other alienated ants. This means that, for every iteration, each alienated ant will follow a different path, i.e. will choose a different solution. The normal ant, in its route, moves in order to maximize the quantity of smelt pheromone and to identify a unique solution. The alienated ant, instead, provides, at each iteration, a different solution (i.e. load balancing ability), each one characterized by a lower pheromone quantity (minimization ability). These characteristics makes AAA suitable to solve easily a great number of problems, like flows management, traffic management and routing problems in which, in general, an efficient load distribution and a run time adaptation is needed. Instead, due to its selection mechanism (that forces the convergence to a unique path) and its off-line pheromone updating method (that needs to know all results before choosing the best one), it is impossible to apply ACO to the above mentioned scenarios without re-adapting the problems. In the next section an example of an AAA application is given: in particular, it will be shown how this algorithm can be applied to the grid scheduling problem.

4 The Alienated Ant Algorithm for a Multi-Broker Grid Environment

This section discusses how the Alienated Ant Algorithm can be adopted to solve the problem of grid scheduling.

As described in [23], the grid environments are composed by a large quantity of different types of processing nodes, storage units, scientific instrumentation and information, belonging to different research organizations, exploited by users of a virtual organization (VO)[2]. Each time a VO user submits a job to the grid, it has to make a reference to one or more Resource Brokers (RBs): the aim of each RB is to receive all job submission requests and foreward them to the most suitable Computing Element (CE).

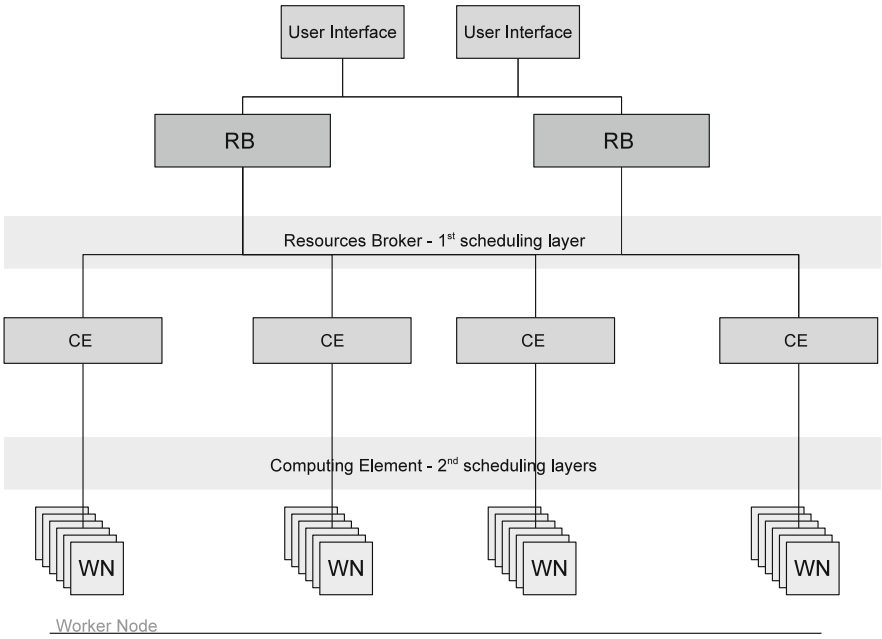


Fig. 1 A typical grid organization

Each CE analyzes the jobs and sends them to one of the underlying Worker Nodes (WNs). The aim of the WNs is to execute jobs and to return their results to VO users.

In Fig. 1, a multi-broker grid environment is shown. In this type of hierarchical organization, three scheduling layers can be identified. At the first level the RBs distribute jobs among CEs; the algorithm adopted in this level strongly influences the performance of the whole system. The second scheduling layer manages job allocation by part of the CEs onto respective underlying WNs. Finally, the lowest scheduling activity consists of the execution by the OS of the WN of the jobs received from the CE. Here only the first two levels of scheduling will be taken into account: for the third level of scheduling, done by the local OS, a non pre-emptive FIFO scheduler is used, that admits only one job per CPU at time, as used in High Performance Computing (HPC) systems like the one considered in the simulations illustrated hereafter.

Considering a job as an ant, the RBs as anthills and the WNs as different food sources, it is possible to transform the behavior of the alienated ant into a job scheduling algorithm.

The functions of the Alienated Ant Algorithm for the grid environment are described below.

Initially, each ant/job builds its own solution to the scheduling problem, i.e. identifies the path linking the RB to a WN, passing through a CE. Each path is composed by two different sub-paths: considering rb_ce_i the path linking RB to the i-*th* CE and

$ce_i\_wn_j$ the one linking the i-*th* CE with the underlying j-*th* WN, the generic solution is represented by the pair (x, y):

$$(x, y)|x \in \{\text{rb\_ce}_i\}, y \in \{\text{ce}_i\_\text{wn}_j\}, i \in [1, n], j \in \left[1, \text{m}_{CE_i}\right] \tag{1}$$

In each scheduling layer one of the two components of the (x, y) pair[2] is chosen. Basing on the pheromone trails evaluation, the scheduler maintains a trail for each related sub-path in a vector (one for each RB and one for each CE): the value of each trail is used to assign to the related path a certain probability of being chosen.

For the x_i sub-path ($x_i \in \{\text{rb\_ce}_i\}$) starting from RB pheromone trails vector, considering ϕ_{x_i} its pheromone trail value, the probability Ψ_{x_i} is calculated as:

$$\Psi_{x_i} = 1 - \frac{\phi_{x_i}}{\sum_{i=0}^{n} \phi_{x_i}}|_{x \in \{\text{rb\_ce}_i\}} \tag{2}$$

For the y_{jk} sub-path ($y \in \{\text{ce}_j\_\text{wn}_k\}$) linked to pheromone trails vector stemming from the j-*th* CE, trails vector, considering $\phi_{y_{jk}}$ its pheromone trail value, the probability $\Psi_{y_{jk}}$, is calculated as:

$$\Psi_{y_{jk}} = \Psi_{x_i}(1 - \frac{\phi_{y_{jk}}}{\sum_{k=0}^{m_j} \phi_{y_{jk}}}) \tag{3}$$

It should be noted that $\Psi_{y_{jk}}$ is a conditional probability, in that the selection of the y_{jk}th path is conditioned by the selection of x_i_ith path, with $i = j$.

The use of these probability functions differentiates the behavior of the alienated ant from the "normal" ants. As opposed to the standard ACO approach, the alienated ant has a high probability of choosing the path with the lightest trail instead of the strongest one. After it has chosen its sub-path, the alienated ant marks it and updates the value of the related pheromone trails' vector entry. On the chosen sub-path, the ant will release a certain quantity of pheromone related to the characteristics[3] of the relative job. The characteristic considered in this paper is an estimate of the job execution time, expressed in minutes. This estimate can be obtained either from a scheduler job profiling [24–27] or from the user indication.

Considering $j\text{et}_i$ as the value of job execution time estimation for the ith job, the update value for the trail representing the jth sub-path ($\Delta\phi_{\text{sp}_j}$) is:

$$\Delta\phi_{\text{sp}_j} = j\text{et}_i \tag{4}$$

In order to simulate the pheromone's evaporation at each scheduling cycle, an update of the trail vector entries value is done. Each pheromone trails' vector entry is updated according to:

- the computational power of the associated resource
- the elapsed time

[2] In particular, the x component will be chosen for the RB and the y component for the CE.

[3] These can be owner, groups, QoS parameter, execution time, etc.

The first parameter assures that the greater the resource power, the greater the pheromone evaporation speed, increasing the probability of the resource being selected repeatedly. The second parameter is used in the evaporation mechanism to give the scheduler a *time* reference (from a fixed "start time") in order to be able to perform asynchronous pheromone updating. [4] In this paper, the formula used for the evaporation mechanism is:

$$\Delta\phi_{\mathrm{sp}_j} = -(t * \mathrm{cmp}_{\mathrm{pwr}_i}) \tag{5}$$

where $\Delta\phi_{\mathrm{sp}_j}$ represents the updating value of the *j*th pheromone trail[5], *t* is the time (in minutes) elapsed since the last update operation and $\mathrm{cmp}_{\mathrm{pwr}_i}$ is a value related to the computational power of the *i*th resource.

The pseudo-code below summarizes the alienated Ant Algorithm:
alienated Ant - C style pseudo code

```
1  trailInit(pherVector[]);
2
3  void alienatedAnt(pherVector[], TimeEstim){
4   //evaporation step
5   for (i=0;i<pherVectorLenght;i++){
6   pherVector[i]-= evap(elasedTime,resNumber);
7   }
8   //solution creation->resource selection step
9   res = evalPher(pherVector);
10  //pheromone trail update step
11  pherVector[res] = pherUpdate(TimeEstim);
12  return;
13 }
```

Some important aspects of the Alienated Ant Algorithm are its simplicity and its completely distributed nature. Each scheduler takes its decision autonomously based on the information available locally. This makes the decisions related to the scheduling process very fast because, for each job, the scheduler only has to find the lowest value in a vector and use it to identify the underlying related resource to which to forward the job. If each RB and CE scheduler adopts this behavior, it is possible to observe [23] the emergence of a general behavior which is able to minimize the average time spent by each job in the job waiting queue.

[4] Without this time reference, to maintain a consistence among the pheromone trails' values it would be necessary to perform a synchronous update which could be very expensive in terms of computational power.

[5] It should be noted that $\Delta\phi_{\mathrm{sp}}$ value cannot be less than zero. If the $\Delta\phi_{\mathrm{sp}}$, after the updating process, has a negative value, it is rounded to zero.

4.1 The Multi Broker Implementation

Generally, in the AAA each scheduler (RB or CE) makes its decisions based on the information that it has in regard to the status (pheromone) of the underlying resource (CEs or WNs). These pieces of information are estimates of the length of the job waiting queues related to the underlying resources. A scheduler is able to produce these estimates with good precision only if it is the only one that submits jobs to the resources. This condition is valid for the CEs that control the job queue in the relative underlying WNs. Unfortunately, in a multi broker grid environment, the above mentioned condition becomes false for the RBs: the CEs, in fact, receive jobs from multiple RBs. Submitting jobs independently from the others, each RB does not know the number of jobs submitted to each CE by the other RBs and, consequently, looses the correct estimate of the length of the waiting job queue . A simple solution to this problem can be obtained by sharing knowledge about the jobs submission. This can be obtained in two different ways.

The first way to gain knowledge on the overall workload of the grid is based on the message exchange between the RBs; the second one is based on a publisher-subscriber model that involve the CEs. The mechanism based on messages exchange between the RBs foresees that each RB sends, for each submitted job, a notification message to other ones. This message has to contain at least the information about job execution time estimation and the name of the CE to which the job has been submitted. Considering n as the total number of jobs and m as the number of brokers present in the considered grid, the number of exchanged message will be $n(m-1)$. This means that for a VO having more than 10 RBs with an average workload composed by 1000 jobs for each RB, the messages needed to coordinate the scheduling decisions are more than 10,000. The publisher-subscribers model, instead, delegates to the CEs the task to update the information for the RBs. Each time a CE receives a job from a specific RB, it sends the others (RBs) a notification message. Also in this case the number of exchanged messages is $(n(m-1))$ but, in case of simultaneous submissions, the same message can be used to notify more than one job.

Although these mechanisms allow each RB to share the same information, their cost in terms of messages exchanged in both cases $(n(m-1))$ is too high and it makes them impossible to be adopted in order to obtain acceptable performance. Discarding the idea of having a precise knowledge of the "flow of the job submission" of the other RBs, it has been decided to obtain at least an estimate of it. Through an estimate, in fact, each RB could adjust at runtime its own representation of the length of the job waiting queue of the underlying CEs. Each time a job is completed, a notification report containing some information about its status (possible errors, execution time, location of the output files, etc) is sent back to the RB. From this information, the RB can extract the time the job spends waiting in the queue before being executed. This information, stored locally in the RB, is influenced by the jobs submitted globally; therefore this information can be used to make an estimate of the jobs submitted by the others. The estimation process is therefore based

on the concept of feedback. In each RB, the value of the pheromone trail which represents the jth CE, ϕ_{CE_j}, depends on:

- the value of the job execution time estimate for the ith job ($j\mathrm{et}_i$)
- the evaporation factor ($-(t \times \mathrm{cmp}_{\mathrm{pwr}_i})$).

From these parameters, the RB obtains an estimate for the time spent in the job waiting queue ($j\mathrm{wq}_j$)

$$j\mathrm{wq}_j = f(\phi_{\mathrm{CE}_j}) = f(j\mathrm{et}_i, t, \mathrm{cmp}_{\mathrm{pwr}_i}); \tag{6}$$

Unfortunately, as previously stated, the $j\mathrm{wq}_j$ obtained does not provide a value corresponding with the true value, caused by the submissions from the other RBs. In fact, there is an external time-variable factor, referred to as K, the value of which is unknown by the RB. The function thus becomes:

$$j\mathrm{wq}_j = f(K, j\mathrm{et}_i, t, \mathrm{cmp}_{\mathrm{pwr}_i}); \tag{7}$$

Without this knowledge, it is impossible for the RB having the correct information of the job waiting queue's length for each CE. This means that the selection mechanism works on incorrect information and, therefore, it looses its load balancing and minimization capability. Considering $j\hat{wq}_j$ as the true time that the jth job has spent waiting in the queue (obtained by using the notification sent by the WN to the RB), the value of K can be obtained by the difference between $j\hat{wq}_j$ and jwq_j.

$$K = F(j\hat{wq}_j, jwq_j) \tag{8}$$

In order to have a more accurate estimate of other RBs job flows, each RB has to maintain a value of the K factor for each underlying CE.

For this purpose, the authors propose a technique based on a natural-like behavior of each ant that returns back to the anthill after having found the food. Marking the followed path also in the phase of return to the anthill, it is possible to obtain information about the time spent by the jobs in the CEs queue.

In this case, however, instead of marking the path with a quantity of pheromone proportional to job length, the alienated ant updates the pheromone trails with a quantity proportional to the time that it has spent in waiting for the execution.[6]. Assuming that the ant covers the same path for both "trips", the related pheromone is updated two times for each submitted job. The first update is synchronous and "in line" with the job forwarding: the RB chooses the CE, sends the job to it and updates the pheromone following the formula:

$$\Delta\phi_{\mathrm{sp}_j} = j\mathrm{et}_i \tag{9}$$

The second update is asynchronous and "off line" in respect with the job submission. In this case the K value obtained is proportional, as said above, to the time that

[6]This information is available by the RB that has submitted the job through the above mentioned update notification mechanism.

the job has spent in the waiting queue. Let $j\text{wt}_i$ be the value of job waiting time in queue for the ith job, the update value for the trail representing the jth CE is:

$$\Delta\phi_{\text{sp}_j} = j\text{wt}_i \tag{10}$$

The greater the value of $j\text{wt}_i$, the greater the value of K, i.e. many jobs have been sent by other RBs, the greater the value that will be added to the pheromone trails and, as a consequence, the lower the probability that another alienated ant will choose the same path to reach the food.

5 Performance Evaluation

In order to evaluate its performance in a real grid scenario, the Alienated Ant Algorithm has been tested on the COMETA consortium[11] grid infrastructure, where it has been compared with the presently used local scheduling algorithm. The aim of these measurements, performed using the Simgrid toolkit (http://simgrid.gforge.inria.fr/), is to show the advantages, in terms of reduction of Average Queue Waiting Time and of improvement of Load Balancing capability, that can be obtained adopting the proposed scheduling solution.

The average queue waiting time (named μqwt) and the load balancing (named (σlb) have been taken into account in that they represent two fundamental parameters for the evaluation of the scheduling algorithms.

The first one, in fact, measures the average time spent by each job in the scheduler queue before being executed and represents an index for the evaluation of the throughput.

The load balancing capability, instead, represents an estimation of the ability of the algorithm to distribute effectively the jobs among the underlying available resources. In particular, in this chapter we take into account the standard deviation of jobs distribution among the Computing Elements.

The scenario considered in these simulations, shown in Fig. 2, consists of 276 CPUs (4 for each WN) distributed over 7 CEs and managed by 3 different RBs. The jobs are divided in 3 different classes: “short”, with duration included in the range of 1 and 10 min, “medium”, with a duration between 10 and 100 min and “long”, with a duration greater than 100 min.

In this scenario, the performance of the Alienated Ant Algorithm (in the figures called AAA) has been compared with the local scheduling algorithm (in the figures LSA). This LSA distributes the jobs over the available resources basing on the class to which they belong: for each CE, it divides the WNs in three different subsets and assigns each subset of WNs to a specific jobs class. Basing on the observation of the workload of the COMETA grid from November 2006 to November 2007, the resources of each CE have been divided as follow: 50% of WNs has been reserved for the execution of “long” jobs, 40% for the “medium” jobs and 10% for the “short” ones. At high level, i.e at RB level, the jobs are distributed using the standard round robin policy. The performance of both algorithms, evaluated in the

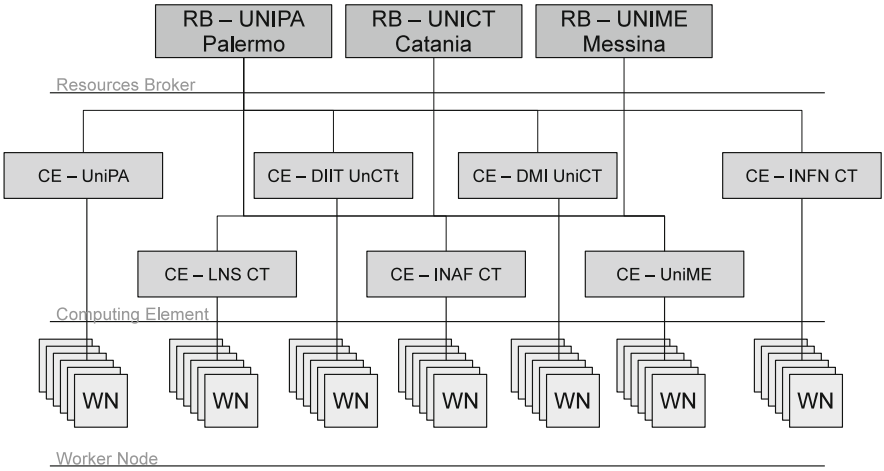


Fig. 2 PI2S2 grid infrastructure

steady state condition, has been compared by changing the job workload both in terms of jobs number and in terms of jobs type (i.e. different proportion among jobs classes).

Fig. 3 and Fig. 4 show the average queue waiting time and the standard deviation of load balancing when the number of jobs submitted to the grid equals, respectively, 3000, 5000, 7500 and 10,000.

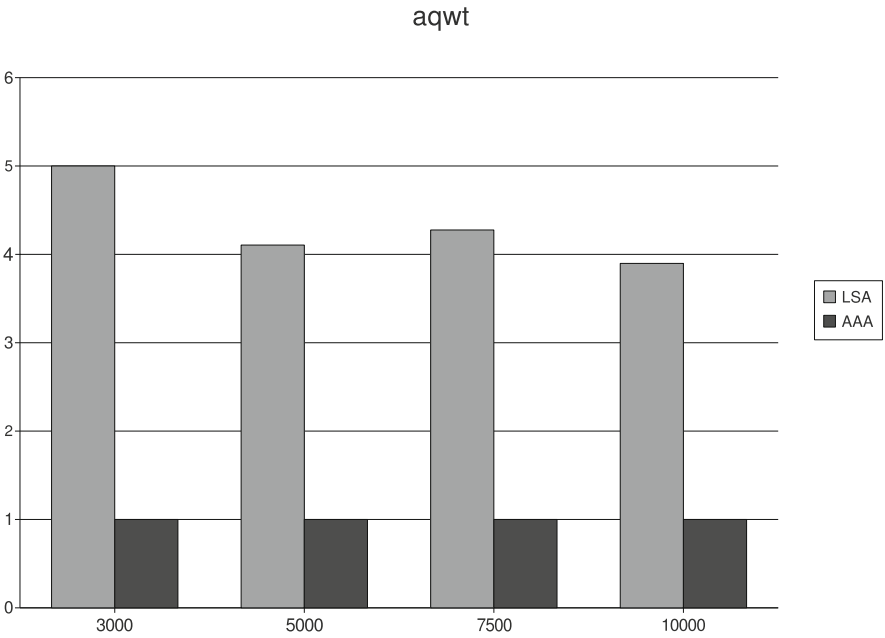


Fig. 3 Average queue waiting time

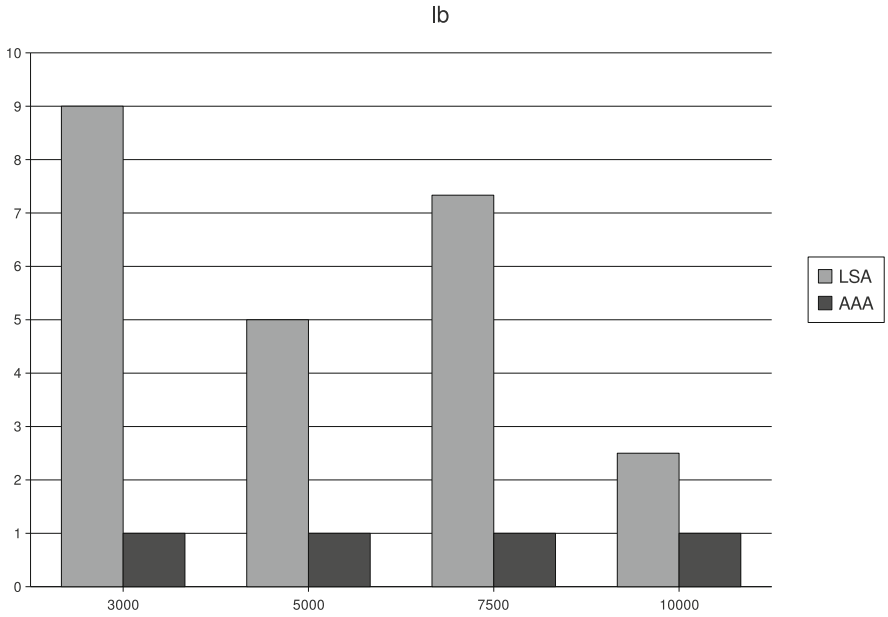


Fig. 4 Load balancing

For the purpose of this chapter, the results have been normalized with respect to the AAA and both the job rate (jobs/s) and the proportion between job classes have been maintained constant (short(10%):medium(75%):long(15%)). The figures show that the performance of the proposed solution is better than the local scheduler for both considered parameters.

Using the Alienated Ant Algorithm, in fact, the time spent in queue by jobs is about 4.3 times (on average) less than the other algorithm. The same consideration can be done for the load balancing capability since the σlb is about 6 times less than the LSA one.

Figs. 5 and 6, instead, show the average queue waiting time and the standard deviation of load balancing when the proportions between short, medium and long jobs are, respectively 55:35:15% (short:medium:long), 45:35:25%, 25:35:45% and 15:35:55%. The number of jobs taken into account for these measurement is equal to 5000.

Also in this case, the results have been normalized with respect to the AAA. These figures confirm the results given above: the performance of Alienated Ant Algorithm, both in terms of the μqwt and in terms of the σlb is, respectively, 3.8 and 7.8 times better that the used scheduling policy.

6 Conclusions

In this chapter the authors propose a nature-inspired algorithm for job scheduling in multi brokers grids.

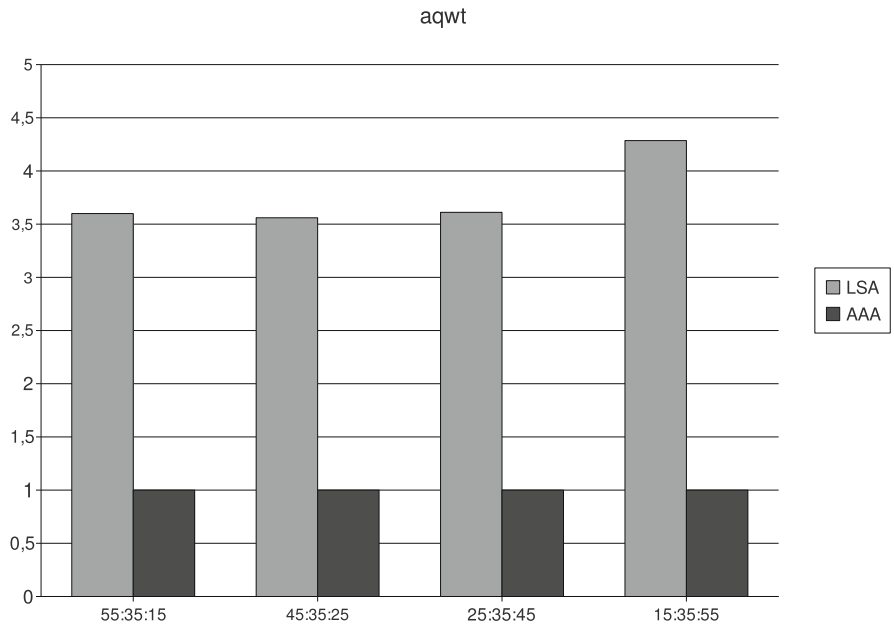


Fig. 5 Average queue waiting time

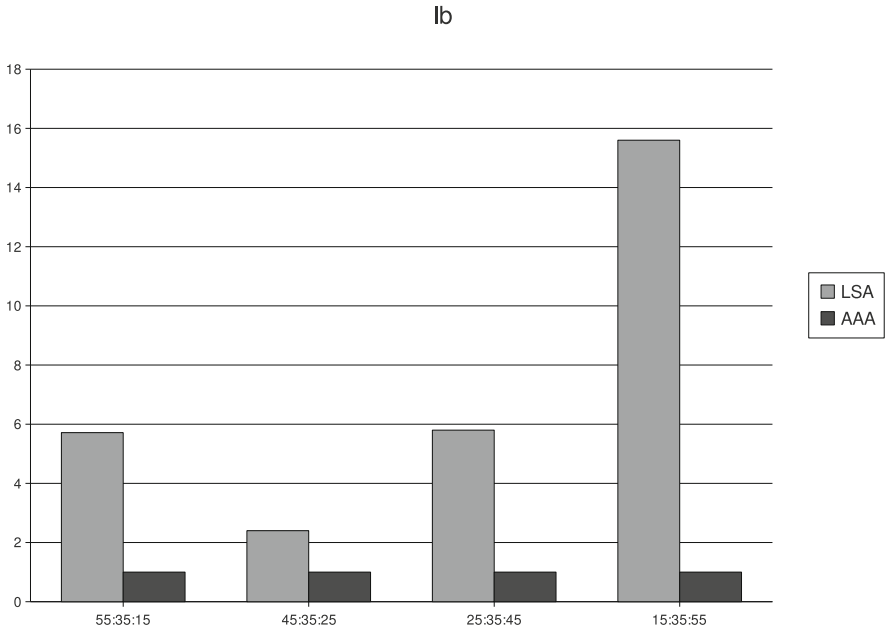


Fig. 6 Load balancing

The proposed solution, called Alienated Ant Algorithm, is a meta-heuristic technique that exploits the self-organization ability of the ants to obtain an effective scheduling policy both in terms of jobs throughput and load balancing. The main characteristic of this algorithm, that also represents the biggest difference with the Ant Colony Optimization approach, consists in the inverse interpretation of the pheromone trails. In fact, while the normal ants choose the path with the highest value of pheromone, the alienated ant chooses the path with the lowest value of pheromone.

This difference changes in depth the behavior of the ant colony in that, instead to converge on a single path (as real ants), the alienated ants tend to spread over all the available paths.

This ”non natural” behavior turns out to be effective in solving distribution problems like the ones related to traffic, flows or load.

The functioning of the proposed algorithm in grids consists of the submission of each job on the resource having, for that time range, the lowest pheromone value, i.e. the least loaded resource. Following this simple strategy, the algorithm is able both to reduce the average queue waiting time for the selected job and to assure, in general, a fair jobs distribution among resources.

The validity of the Alienated Ant Algorithm has been tested applying it to a real grid scenario, the COMETA consortium grid infrastructure, where it has been compared with the currently used local scheduler.

The results of the simulations have demonstrated that the proposed approach turns out to be better both in terms of average queue waiting time and in terms of load balancing capability.

References

1. C. Kesselman, I. Foster, and S. Tuecke. The anatomy of the grid - enabling scalable virtual organizations. International Journal of High Performance Computing Applications, 15(3), 200–222, 2001
2. I. Foster, C. Kesselman, J. Nick, and S. Tuecke. The Physiology of the grid: An open Grid services architecture for distributed systems integration, Open Grid Service Infrastructure WG, Global Grid Forum, 2002
3. I. Foster and C. Kesselman. The grid: blueprint for a new computing infrastructure, Chapter 3. Morgan Kaufmann Publishers, San Francisco, CA ISBN: 1-558660-475-8, 1999
4. Globus toolKit:http://www.globus.org, 2008
5. gLite middleware:http://www.egee.org, 2008
6. Omii-UK middleware:http://www.omii.org
7. M. Dorigo and C. Blum. Ant colony optimization theory: A survey. Journal Theoretical Computer Science, 344 (2-3), 243–278, 2005
8. K. M. Sim and W. H. Sun. Ant colony optimization for routing and load-balancing: Survey and new directions systems, man and cybernetics, Part A, IEEE Transactions, 33, 560–572, 2003
9. D. Merkle, M. Middendorf, and H. Schmeck: Ant colony optimization for resource-constrained project scheduling. IEEE Transaction on Evolutionary Computation, 6(4), 333–346, 2002
10. C. Blum, and M. Sampels. An ant colony optimization algorithm for shop scheduling problems. Journal of Mathematical Modelling and Algorithms, 3(3), 2004

11. COMETA: www.consorzio-cometa.it, 2008
12. C. Blum, and A. Roli. Metaheuristics in combinatorial optimization: overview and conceptual comparison. ACM computing surveys, 35(3) pp., 268–308, 2003
13. T. Stutzle and M. Dorigo. ACO algorithms for the quadratic assignment problem, new ideas in optimization, isbn:0-07-709506-5, McGraw-Hill, London pp. 33–50, 1999
14. M. Dorigo and L.M. Gambardella. Ant colony system: A cooperative learning approach to the traveling salesman problem. IEEE Transaction on Evolutionary Computation 1, 53–66, 1997
15. L.M. Gambardella, M. Dorigo. Ant colony system hybridized with a new local search for the sequential ordering problem. INFORMS Journal Computing 12, 237–255, 2000
16. R. Schoonderwoerd, O. Holland and J. Bruten. Ant-like Agents for Load Balancing in Telecommunications Networks. Proceedings of the first international conference on Autonomous agents, Marina del Rey, CA, 1997
17. M. Reimann, K. Doerner, R.F. Hartl, D-ants: savings based ants divide and conquer the vehicle routing problems. Computer Operation Research 31, 563–591, 2004
18. ML. den Besten, T. Stutzle, and M. Dorigo. Design of iterated local search algorithms: An example application to the single machine total weighted tardiness problem. Proceedings of EvoStim01, Lecture Notes in Computer Science, Springer Berlin, pp. 441–452, 2001
19. T. Stutzle. An ant approach to the flow shop problem. Proceedings of the 6th European Congress on Intelligent Techniques & Soft Computing, Aachen, Germany 1998
20. C. Blum, and M. Sampels. Ant colony optimization for fop shop scheduling: A case study on different pheromone representations. Proceedings of the 2002 Congress on Evolutionary Computing, Honolulu, H1 pp. 1558–1563, 2002
21. S. Fidanova and M. Durchova. Ant Algorithm for grid scheduling problem. LNCS 3743, Sozopol, Bulgaria pp. 405–412, 2006
22. H. Yan, X.-Q. Shen, X. Li, and M.-H. Wu. An Improved Ant Algorithm for Job Scheduling in Grid Computing. Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou 2005
23. M. Bandieramonte, A. Di Stefano, and G. Morana. An ACO inspired strategy to improve jobs scheduling in a grid environment. Proceedings of ICA3PP 2008, Agianapa, Cyprus pp. 30–41, 2008
24. Y. Gao, H. Rong, and J.Z. Huang: Adaptive Grid job scheduling with genetic algorithms. Future Generation Computer Systems 21, 151–161, 2005
25. J. Cao, D.P. Spooner, S.A. Jarvis, and G.R. Nudd. Grid load balancing using intelligent agents. Future Generation Computer Systems 21, 135–149, 2005
26. D.P. Spooner, S.A. Jarvis, J. Cao, S., Saini, and G.R. Nudd. Local Grid scheduling techniques using performance prediction. IEE Proceedings Explore, 150, 87–96, 2003
27. L. Yang, J.M. Schopf, and I. Foster, Conservative scheduling: Using predicted variance to improve scheduling decisions in dynamic environments. Proceedings of the 2003 ACM/IEEE conference on Supercomputing, Phoenix, AZ pp. 31–47, 2003
28. T.L. Casavant and J.G. Kuhl. A taxonomy of scheduling in general-purpose distributed computing systems. IEEE Transactions on Software Engineering, 14, (2), 141–154, February 1988

Part III
Networking

Topology Design of a Service Overlay Network for e-Science Applications

D. Adami, C. Callegari, S. Giordano, G. Nencioni, and M. Pagano

Abstract Nowadays, large-scale numerical simulation, data analysis, remote access to experimental apparatus and cooperative working play a key role in the practice of science and engineering. In this scenario, highly distributed grid environments, including computing, storage and instrument elements, have to be interconnected through high performance networks. In addition, grid applications often have stringent requirements in terms of Quality of Service (QoS), flexibility, network resource provision. Unfortunately, these network requirements can not be easily satisfied, because today's Internet only provides a best-effort packet delivery service. In the past few years, overlay networks have emerged as a profitable way to come through the limitation of the current Internet and to provide value-added network services (QoS, multicasting, security, etc.). This chapter is focused on overlay topology design and, more specifically, on the choice of the best option within a limited set of topology layouts that allows to fulfill QoS network requirements (bandwidth, delay). The problem is formulated as the minimization of a cost function which takes into account traffic demand and latency. Apart from existing overlay topologies, a new traffic demand-aware topology is proposed. Guidelines for the selection of the best topology are provided through extensive simulations.

1 Introduction

In the last years, continuous technology improvements, new collaboration modalities enabled by the ubiquitous Internet, and increasingly complex problems have brought a revolution in the practice of science and engineering. Today's e-Science applications rely on large scale numerical simulations, data analysis and collaboration as much as on individual experimentalists and theorists. Data-intensive applications, simulation-based science, and remote access to experimental apparatus are some of the new modes that increasingly define twenty-first-century science and engineering. More specifically, dramatic improvements in the capability and

D. Adami (✉)
CNIT Research Unit Department of Information Engineering, University of Pisa, Pisa, Italy
e-mail: davide.adami@cnit.it

F. Davoli et al. (eds.), *Remote Instrumentation Services on the e-Infrastructure*,
DOI 10.1007/978-1-4419-5574-6_10,

capacity of sensors, storage systems, computers, and networks are enabling the creation of data archives of enormous size and value. The sources of this large amount of data span a wide spectrum, going from individual, highly specialized and expensive scientific devices located at a single site (e.g. the Large Hadron Collider facility at CERN in Geneva [1]), to high-throughput sensors in fields as varied as environmental and earth observation, astronomy, human health-care monitoring, etc. Moreover, numerical simulation represents another problem-solving methodology, largely used, for instance, in climatology and astrophysics. Computational approaches, requiring high-end ultrafast supercomputers, are increasingly being applied in scientific fields long dominated by experiments (e.g. UK Comb-e-Chem project [2]).

However, the increasing prominence of simulation and data-driven science does not mean that experimental science has become less important. On the contrary, the advance of technology is also producing new revolutionary experimental apparatus and the emergence of high-speed networks allows to integrate them into the scientific problem-solving process in ways not previously conceivable.

Therefore, the scientific community vision is evolving to the more sophisticated and ubiquitous "Virtual Organization" (VO) concept, in which Grid middleware enables "coordinated resource sharing and problem solving in dynamic multi-institutional organization" [3].

In most cases, e-Science applications heavily stress the network interconnecting the Computing Elements (CEs), the Storage Elements (SEs), the Instrument Elements (IEs), and the sensors. In addition, these applications often have stringent requirements in terms of Quality of Service (QoS), intra-domain and inter-domain resource provisioning, but today's Internet is unable to provide such advanced services on end-to-end basis because it still relies on the original TCP/IP architecture.

In recent years, overlay networks have seen wide acceptance as a new approach to satisfy the needs of specific purpose applications (e.g. peer-to-peer applications), to develop new routing paradigms (e.g. content-based routing) or to provide unsupported network services (e.g. application layer multicast, resilience [4], QoS, QoS Routing, etc.). Overlay networks are sometimes built among the end-hosts without any support from the Internet Service Providers (ISPs) [5]. Though highly flexible, this type of overlay networks can not guarantee end-to-end QoS support, because the underlying IP network typically provides only a best-effort packet delivery service.

This paper deals with the deployment of a Backbone Service Overlay Network (B-SON) [6, 7] for e-Science applications. In this case, the overlay network consists of nodes strategically placed around the Internet and managed by a third-party, also known as VO Overlay Service Provider (OSP). According to our vision, the B-SON nodes match with the access nodes of the VO remote sites and overlay paths are created among such nodes. Running an e-Science application means, first of all, selecting a number of CEs, SEs and IEs scattered around the Internet. The set-up of the B-SON is performed by the VO OSP, which has to select the best topology for interconnecting the VO remote sites.

Overlay topology design has been one of the most challenging research areas over the past few years. When considering different topologies, it is necessary

to understand how they affect the overlay routing performance and how to efficiently build overlay topologies connecting all the overlay nodes. Several works have focused on the selection of the best overlay links (e.g. [8]), but other issues, such as binding end systems to overlay access nodes [8], positioning the overlay nodes [9, 10] or choosing the right number of overlay nodes [10], have also been faced. In these studies, the overlay topology is usually represented as a graph and the topology design problem is expressed as an optimization problem. The general approach relies on the use of heuristic algorithms that allow finding a near-optimal solution.

Most works [8–13] analyze the topology design problem from an economic point of view with the aim to minimize the cost for the deployment of the SON. Only a few works [14, 16] deal with the SON topology design problem from a network performance perspective. In [14], the authors aim at finding the overlay topology minimizing a cost function which takes into account the overlay link creation cost and the routing cost. They also highlight how the traffic demand affects the creation of new overlay links. In [15], instead, the authors compare several existing and some new overlay topologies in terms of resilience.

This chapter investigates the overlay topology design problem with the aim to identify, within a restricted set of pre-defined reference models, the best trade-off among QoS performance (bandwidth, delay) and number of accepted overlay sessions. Taking into account the traffic demand among the overlay nodes, the performance and the overhead of some existing overlay topologies are evaluated through extensive simulations. Moreover, a new topology, called K-Shortest Path Tree (KSPT), is introduced and its performance compared with the other topologies.

The chapter is organized as follows. Section 2 introduces the application scenario and formalizes the topology design problem. Next, Section 3 describes the main features of the topologies analyzed in this chapter, whereas Section 4 discusses the results of the simulations carried out to analyze the impact of different overlay topologies on overhead and performance. Finally, Section 5 concludes the chapter with some final remarks.

2 Problem Statement

Let us consider, as an application scenario, the connectivity map for the DORII project [16], recently funded by the European Union within FP7. The main target of this project is the deployment of a grid infrastructure for e-Science applications.

The network infrastructure (see Fig. 1) is organized according to four hierarchical levels:

- Sensor Networks, for data acquisition.
- Local Area Networks, for intra-site connectivity.
- Access Networks, for extra-site connectivity.
- Core Network, for inter-sites connectivity.

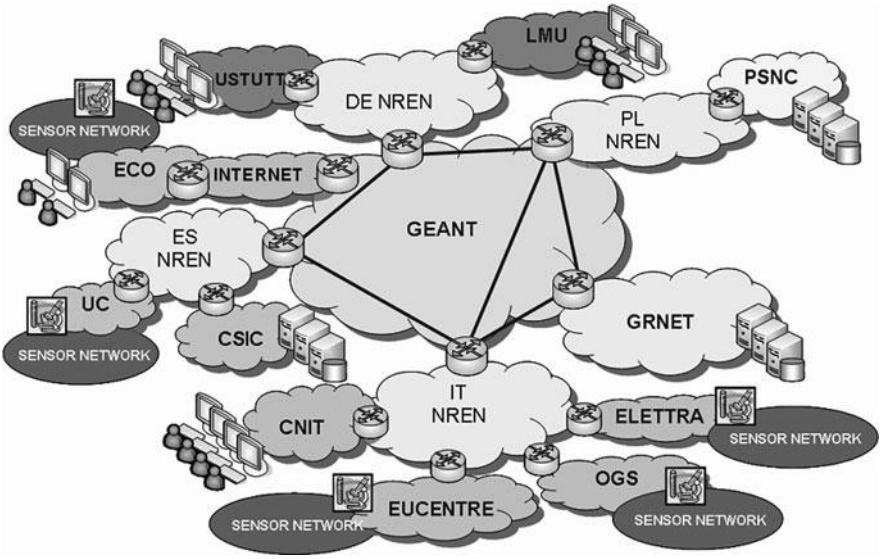


Fig. 1 DORII project: Network connectivity map

- Since each DORII site may either access or share the grid resources (CE, SE, IE, etc) belonging to DORII VO, a B-SON spanning DORII sites can be created.

Two different approaches may be followed for the construction of the overlay network. In both cases, the service overlay nodes are supposed to match with the IP-layer nodes (access routers) that access the LAN at the end-user site. In the first case, the topology is flat and thus completely unaware of the actual network structure, whereas in the second case the topology layout is hierarchical and takes into account the multi-domain organization of the Internet.

In the following, we suppose that:

- The location of the overlay nodes is pre-determined.
- The overlay path between a pair of overlay nodes is selected by using the Dijkstra algorithm. The metric associated to each overlay link is the delay.
- Each overlay path is composed of IP-layer links. At IP layer, the path between a pair of nodes is computed by using the Dijkstra algorithm and the cost of each link is assigned in inverse proportion to the link bandwidth.
- The overlay topology construction problem can be formulated as follows.

Let us consider:

– The IP-layer topology $G_P(V_P,\ E_P)$ where
  – V_P is the set of nodes.
  – E_P is the set of IP layer links.

  - N is the total number of V_P nodes.

- A set of overlay nodes, V_O, which is a subset of V_P.

  - M is the total number of V_O nodes.

- The IP-layer path-link indicator function P_{ij}^{mn}, where $P_{ij}^{mn} = 1$ if the IP-layer path between m and n includes the IP-layer link ij.
- The overlay path-link indicator function Q_{mn}^{xy}, where $Q_{mn}^{xy} = 1$ if there exists an overlay path from x to y that includes the overlay link mn.
- The delay of the IP-layer link ij, D_{ij}.
- The traffic demand between the overlay nodes x and y, T_{xy}.

The goal is to find the topology $G_O(V_O, E_O)$ (i.e., a sub-set of overlay links, E_O) that minimizes the cost function:

$$\sum_{xy} T_{xy} \sum_{mn} Q_{mn}^{xy} \sum_{ij} D_{ij} P_{ij}^{mn}$$

in which the end-to-end delay is weighted by the traffic demand, with the constraints:

- $\sum_{mn} Q_{mn}^{xy} \sum_{ij} D_{ij} P_{ij}^{mn} \leq \delta$, where δ is the overlay paths delay constraint.
- $\sum_{xy} T_{xy} Q_{mn}^{xy} \leq \tau_{mn}$, where τ is the available bandwidth of the overlay link mn.

It is worth highlighting that if popular or greedy nodes exist, the topology layout does not change, but such nodes are located at the centre of the constructed topology, so as to minimize their latency with respect to the other nodes (see [17] for further details).

3 Overlay Network Topologies

The problem formulated in the previous section is NP-hard and therefore it is too complex from a computational point of view. The basic idea behind our approach is to interconnect the overlay nodes using a set of well-known network topologies and to choose the best topology with respect to a set of performance metrics.

This section lists the main features of several candidate topologies that appeared in literature. Moreover, a new traffic demand-aware topology is proposed.

3.1 Full-Mesh (FM)

In the FM overlay network topology, each node is adjacent to all the other nodes at overlay layer. Therefore, every node has the same neighbours number, i.e. the same node degree (d). To retrieve information on the status and performance of the overlay links, every node has to periodically send probing packets to all neighbours at

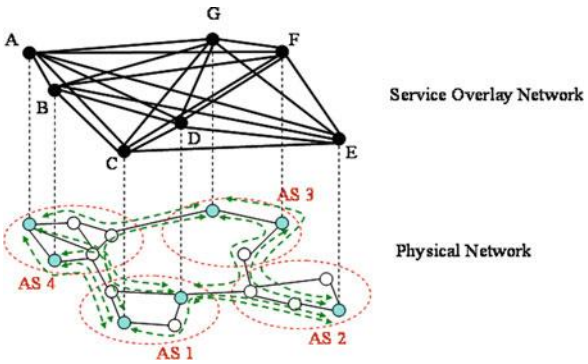


Fig. 2 Full-mesh overlay topology

overlay layer. The FM topology is therefore characterized by the highest overhead, due to overlay monitoring traffic. Figure 2 shows an example of FM topology.

3.2 K-Minimum Spanning Tree (KMST)

A minimum spanning tree (MST) is the lowest cost tree among all the candidate trees connecting a given set of nodes. A KMST overlay topology is composed of K MSTs, where the k-th MST of the composite graph is the MST of the initial graph excluding the edges of previously computed MSTs. Therefore, the overlay links of the K trees are not overlapped. The K value can be chosen as a trade-off among cost, performance and node degree constraints.

This approach has been proposed in [18] so as to minimize the overhead due to the traffic generated for link monitoring.

Figure 3 shows a 2MST, where the two trees are depicted with dashed and solid lines, respectively.

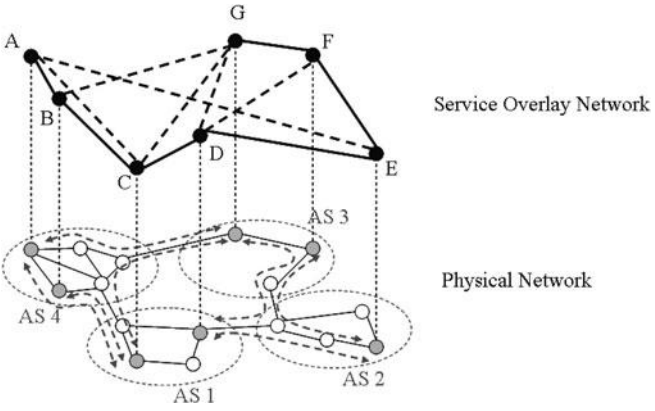


Fig. 3 Two-minimum-spanning-tree overlay topology

3.3 Mesh-Tree (MT)

This topology (called Mesh-Tree in [15]) is obtained by joining the MST connecting all the overlay nodes with the set of overlay links connecting those overlay nodes that have a grandchild-grandparent or uncle-nephew relationships in the MST. Figure 4 shows an MT overlay topology: MSTs are represented with solid lines, whereas links joining nodes with a grandchild-grandparent or uncle-nephew relationship are dashed.

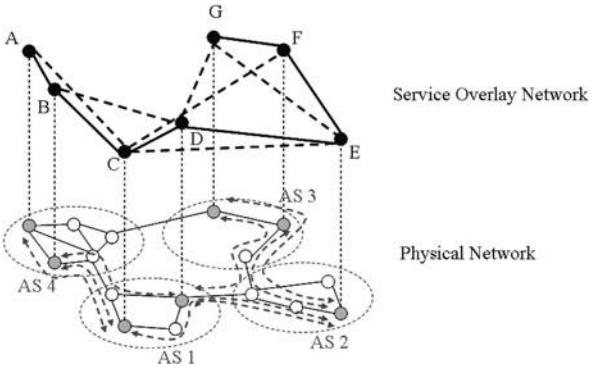


Fig. 4 Mesh-tree overlay topology

3.4 Adjacent-Connection (AC)

The Adjacent Connection (AC) topology relies on the IP-layer topology knowledge. The construction of the AC topology, proposed in [19] and [20], is based on a unique condition: if no overlay node is directly connected to the nodes belonging to the IP-layer path between any pair of overlay nodes, an overlay link connecting this pair of overlay nodes is created. Figure 5 shows an example of AC overlay topology.

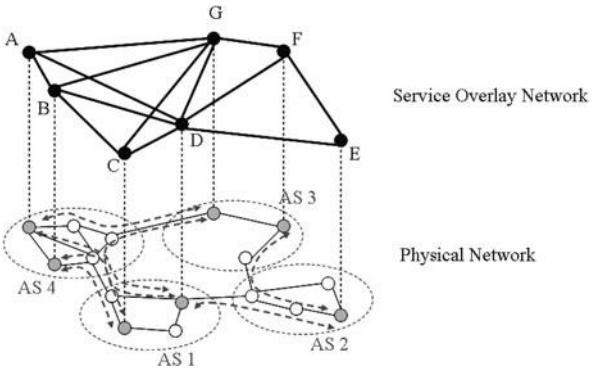


Fig. 5 Adjacent-connection overlay topology

3.5 K-Shortest Path Tree (KSPT)

In this paper, we introduce a new topology, called K-Shortest Path Tree (KSPT), which is constructed taking into account the traffic demand among the overlay nodes. An SPT is made up of minimum cost paths from a node (root) towards all the other nodes. In our case, the metric is the overlay link delay. The choice of the K value, like in KMST, results from a trade-off among cost, performance and node degree constraints.

A KSPT topology can be constructed as follows:

1. Initialize K_{Th}=ceiling[M/3], K = 0
2. Create an SPT for each overlay node.
3. Select the SPT whose root corresponds to the overlay node with the greatest traffic demand.
4. Add the SPT to the topology layout. Increment the value of K (K=K+1).
5. Calculate the average node degree. If it is greater than the node degree constraint or K_{Th} is reached, end. Otherwise, continue.
6. Select the SPT with minimum overlap with the created topology. Go to step 4.

Figure 6 shows a 2SPT, where the two trees, with roots G and C, are depicted by using dashed and solid lines, respectively.

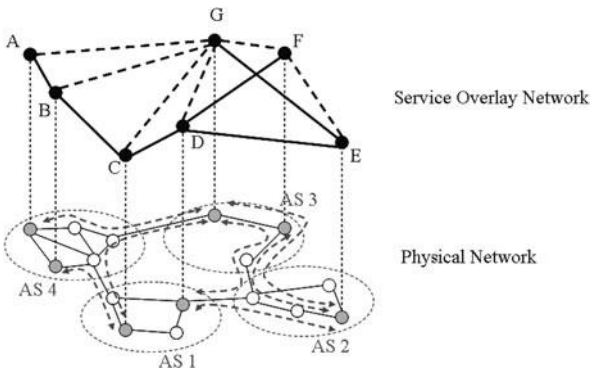


Fig. 6 Two-shortest path tree overlay topology

4 Performance Evaluation

4.1 Simulations Settings

To compare the performance of the topologies described in the previous section, we considered flat and hierarchical IP-layer topologies, generated by means of BRITE [21].

In the flat topology, the generation model for nodes interconnection is based on the Waxman's probability:

$$P(u, v) = \alpha e^{\frac{-d}{\beta L}}$$

where P (u, v) is the probability that a link between the nodes u and v is created, $\alpha(0 < \alpha \leq 1)$ and $\beta(0 < \beta \leq 1)$ are Waxman specific parameters (in our simulations α=0.03, β=0.03), d is the Euclidean distance between the nodes, and L is the maximum distance between any two nodes. The link bandwidth has been assigned according to a uniform distribution in the range [10, 1000] Mb/s. It is worth highlighting that this value does not represent the IP-layer link capacity, but the available bandwidth, since non-overlay traffic may pass through the same links. The link delay has been assigned in proportion to the link length.

To create a hierarchical random topology, it has been adopted a top-down approach that consists of the following three steps:

1. Generate an AS-level topology according to the Waxman's model (for redundancy reasons, we set α=1).
2. For each AS, generate a router-level topology by using the Waxman's model with the same parameters as for the flat topology.
3. Finally, use the random method to interconnect ASs as dictated by step 1: if (i, j) is a link in the AS-level topology, then two nodes (u, v), randomly chosen from ASs (i,j), respectively are interconnected at 512 Mb/s.

Simulations have been carried out randomly associating overlay nodes with the generated IP-layer nodes and interconnecting them through one of the topologies described in Section 3.

For a fair comparison of the candidate topologies and to also meet the node degree constraint d, other research works modify the construction of the topology as follows: if an overlay node degree exceeds d in the resulting overlay topology, only the d closest neighbours overlay nodes are maintained, while the others are pruned. Since this approach dramatically affects the intrinsic nature of each topology, our choice was to consider the overlay node degree as a primary performance metric without changing the resulting overlay topology.

Moreover, the traffic demand is generated according to a uniform distribution normalized with respect to M, so that the overall traffic demand is statistically the same when the number of overlay nodes changes.

A customized simulation environment has been developed by using C programming language.

Two simulation scenarios have been considered:

1. The IP-layer topology is flat (N=300) and the number of overlay nodes changes [M=(10, 20, 40, 60, 80%)*N]
2. The IP-layer topology is hierarchical (N=300, Number of ASs=4, Number of nodes per-AS=75) and the number of overlay nodes changes [M=(10, 20, 40, 60, 80%)*N]

The average value and the 99% confidence interval of each performance metric have been evaluated in the two scenarios by performing a set of 50 independent simulations for each pair (*N, M*).

4.2 Performance Metrics

To compare the performance of the different overlay topologies, we introduce four parameters, each of them representing a specific topology feature.

- **Bandwidth Rejection Ratio (BRR)**
  This performance parameter represents the probability that an overlay path with bandwidth and delay requirements can not be created due to bandwidth unavailability:

$$\text{BRR} = \frac{\text{Number of bandwidth rejected end - to - end overlay paths}}{\text{Total number of end - to - end overlay paths}}$$

- **Delay Rejection Ratio (DRR)**
  This performance parameter represents the probability that an end-to-end overlay path with bandwidth and delay requirements can not be created due to delay limitations although the bandwidth constraint could be satisfied:

$$\text{DRR} = \frac{\text{Number of delay rejected end - to - end overlay paths}}{\text{Number of bandwidth accepted overlay paths}}$$

- **Average Node degree (AND)**
  This performance parameter provides information on the topology overhead. If d_i is the neighbours number of the *i-th* overlay node, AND can be defined as follows:

$$\text{AND} = \frac{\sum_{i=1}^{M} d_i}{M}$$

- **Accepted Traffic Weighted Delay (ATWD)**
  This performance parameter describes the capability of the topology to associate low delays to the highest traffic demands:

$$\text{ATWD} = \frac{\sum_{\text{ij}} \left(\text{Accepted traffic demand between the nodes i,j } \times \text{Delay}_{\text{ij}} \right)}{\text{Total accepted traffic}}$$

4.3 Simulations Results

A different graph has been worked out for each performance metric in case of flat as well as hierarchical topologies.

4.3.1 Flat IP-layer Network Model

Figure 7 shows the AND in a flat IP-layer network when M ranges from 0.10∗N to 0.80∗N. It is worth highlighting that the behaviour of this metric is not affected by the size of the IP-layer network, so these results may also be extended to IP-layer networks with a different number of nodes.

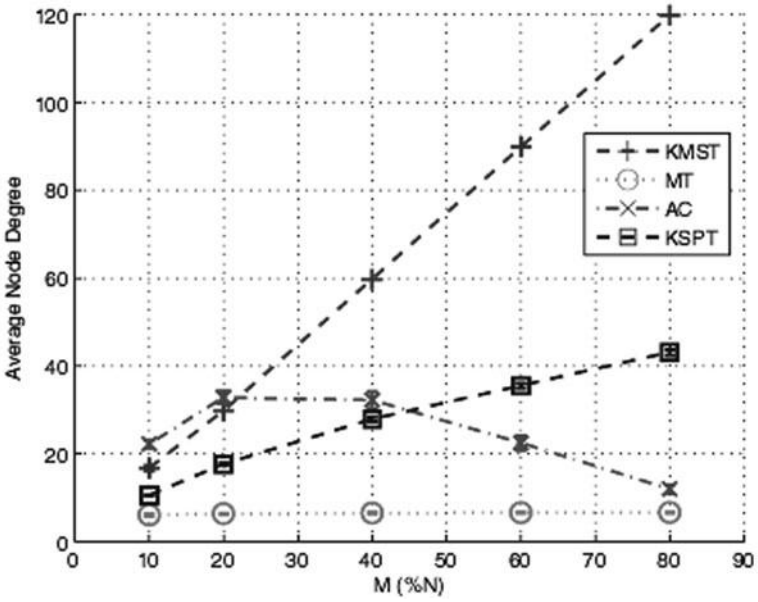


Fig. 7 AND for a flat IP-layer network

The graph outlines that MT always has the minimum node degree that remains constant independently of M.

In contrast, the AND of KMST and KSPT increases almost linearly with M, but KSPT outperforms KMST due to the different way K is selected and the mechanism used to limit the value of K when the AND is too high.

Concerning the AC overlay topology, at the beginning, the AND is the highest one and increases up to reaching a maximum value when M=0.20∗N. Afterwards, it starts decreasing and when M $\geq$ 0.60∗N, AC outperforms not only KMST, but also KSPT. Indeed, when M increases, it is more likely that there is another overlay node along the path between two overlay nodes and therefore the number of overlay links decreases. The AND of FM is always equal to M−1.

Figure 8 reports the BRR average value and confidence interval. As outlined in the graph, FM, KMST, AC and KSPT show a similar behaviour. MT, instead, always has the highest BRR. This result is in accordance with the AND trend: since MT has the lowest AND, also the available bandwidth is the lowest.

Figure 9 shows the DRR trend. Also in this case, FM, KMST, KSPT and AC show a similar behaviour that decreases with M. Instead, MT has the worst performance,

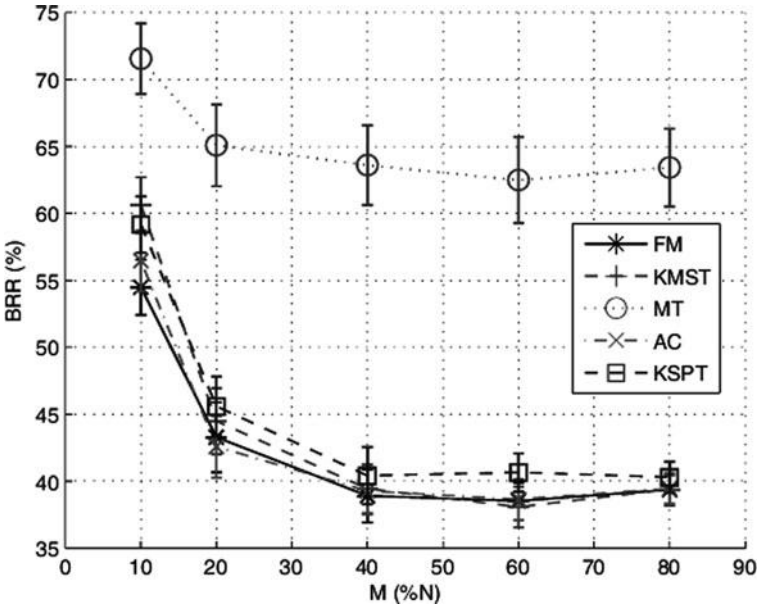


Fig. 8 BRR for a flat IP-layer network

since the values of the DRR for MT always are significantly higher than the ones

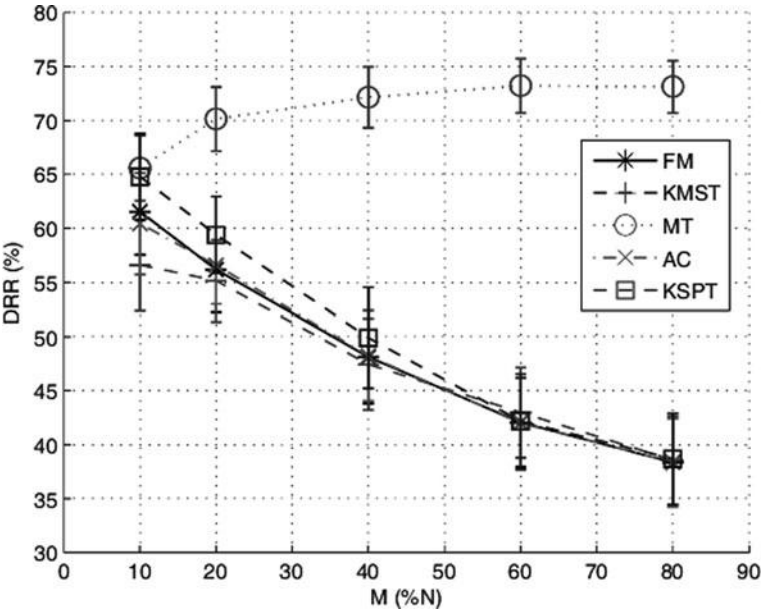


Fig. 9 DRR for a flat IP-layer network

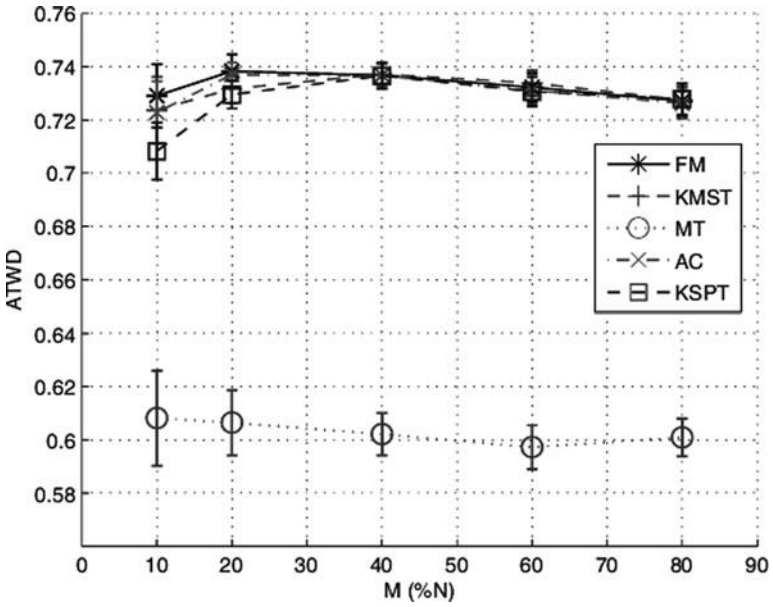


Fig. 10 ATWD for a flat IP-layer network

obtained for the other topologies when M $\geq$ 0.20*N. This is due to the lower AND of the MT topology.

Finally, let us consider the ATWD parameter. By definition, this metric has low values when the overlay topology favours the creation of overlay paths with the lowest delay among the overlay nodes that exchange the largest amount of traffic. As shown in Fig. 10, MT highly outperforms FM, KMST, KSPT and AC.

From the simulation results, we can infer that MT performs worse than the other overlay topologies because, although it is characterized by the lowest AND and ATWD metrics, most of the traffic demand is rejected.

In large size flat IP-layer networks, the overlay topology should be chosen based on the number of overlay nodes. If the number of overlay nodes is large, AC is the best overlay topology, because it performs in the same way as the other overlay topologies, but its AND is the lowest. Instead, when the number of overlay nodes is small, KSPT is the best overlay topology, since in this case it has the lowest AND.

4.3.2 Hierarchical IP-layer Network Model

In a hierarchical IP-layer network, the AND behaviour (see Fig. 11) is similar to that obtained for a flat IP-layer network. Nevertheless, the graph outlines that AC has the same AND as KMST when M=0.10*N and a lower AND even though M=0.20*N.

Figures 12 and 13 show the BRR and DRR behaviours, respectively. Also in this case, the trends are similar to those obtained for flat IP-layer topologies and MT performs worse than all the other approaches. However, it is relevant to highlight that, apart from MT, KSPT has the highest BRR and DRR and the performance

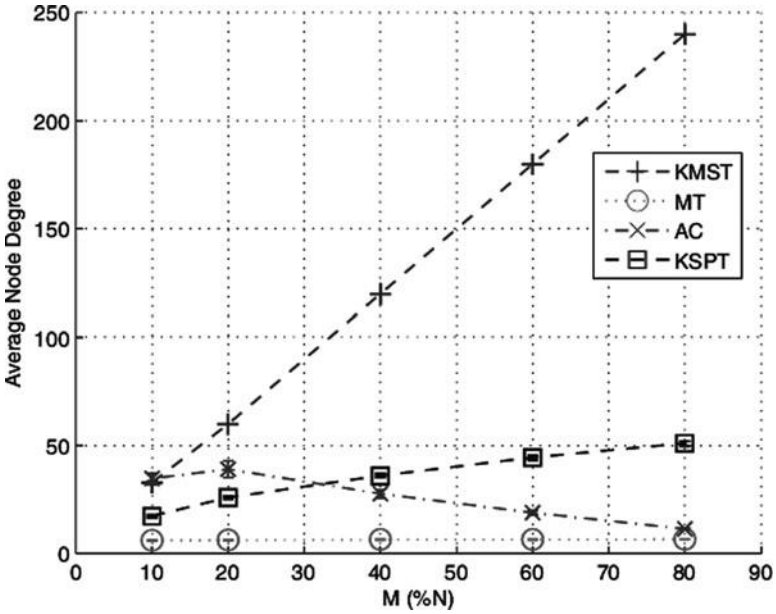


Fig. 11 AND for a hierarchical IP-layer network

difference among KSPT and FM, KMST, AC is larger for small and middle size overlay networks.

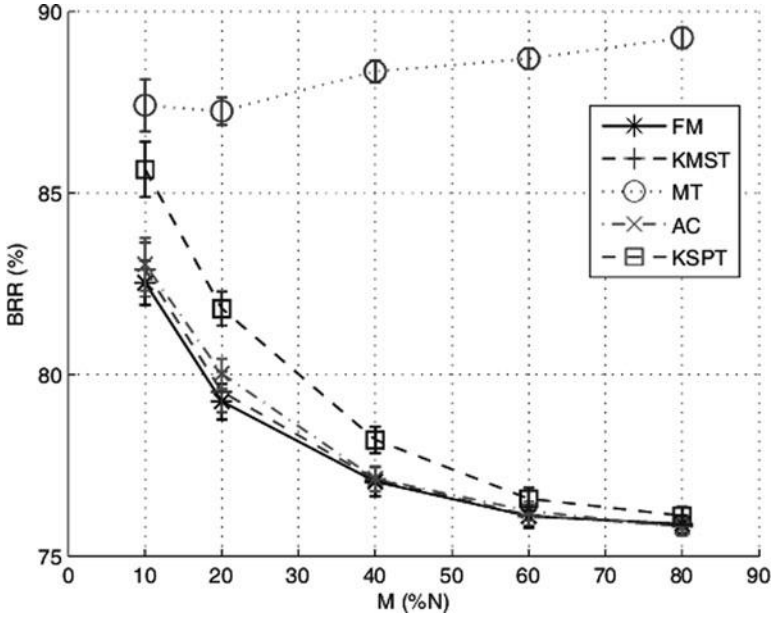


Fig. 12 BRR for a hierarchical IP-layer network

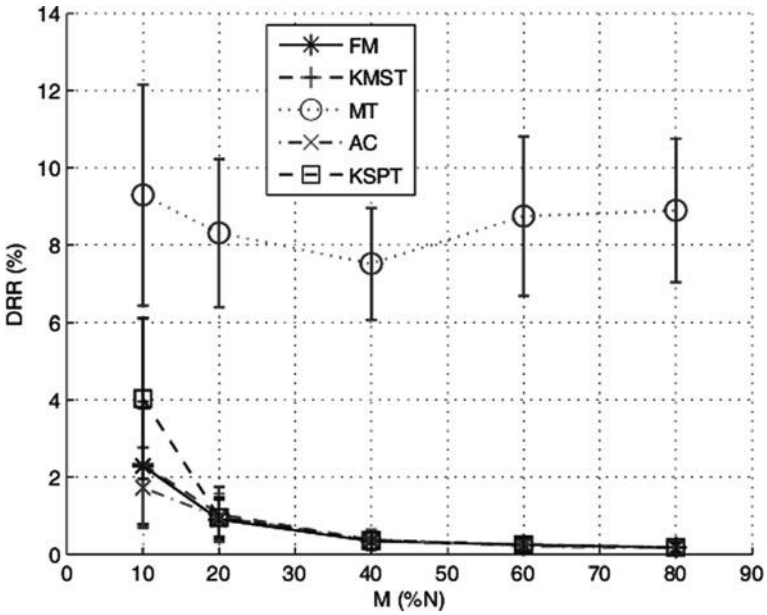


Fig. 13 DRR for a hierarchical IP-layer network

Figure 14 highlights that the ATWD parameter behaviour in hierarchical IP-layer networks shows some differences with respect to the flat topology. Indeed, although the number of IP-layer nodes is increased, MT and KSPT do not exhibit any performance improvement as it occurred in case of a flat topology.

In summary, also in a hierarchical IP-layer network, MT underperforms the other overlay topologies for the same reasons as in case of a flat network.

As far as the remaining topologies are concerned, AC outperforms all the other topologies, due to the "knowledge" of the underlay topology which allows AC to reduce the AND and, as a result, the overhead. Moreover, the results of the simulations outline that KSPT is not a suitable topology for hierarchical IP networks, since it always performs worse than the other overlay topologies (except MT).

5 Conclusions

This chapter dealt with the topology design of a B-SON for e-Science applications taking into account performance constraints, such as traffic demand and overhead. Since finding the optimum topology is NP-hard, the approach proposed in this chapter aims at selecting the best option within a restricted set of well-known topologies, already introduced in previous research works concerning SONs. Also, a new traffic demand-aware topology, called KSPT has been proposed. Through extensive simulations, the performance of each overlay topology have been evaluated with respect to pre-defined performance metrics in either a flat or hierarchical large-size

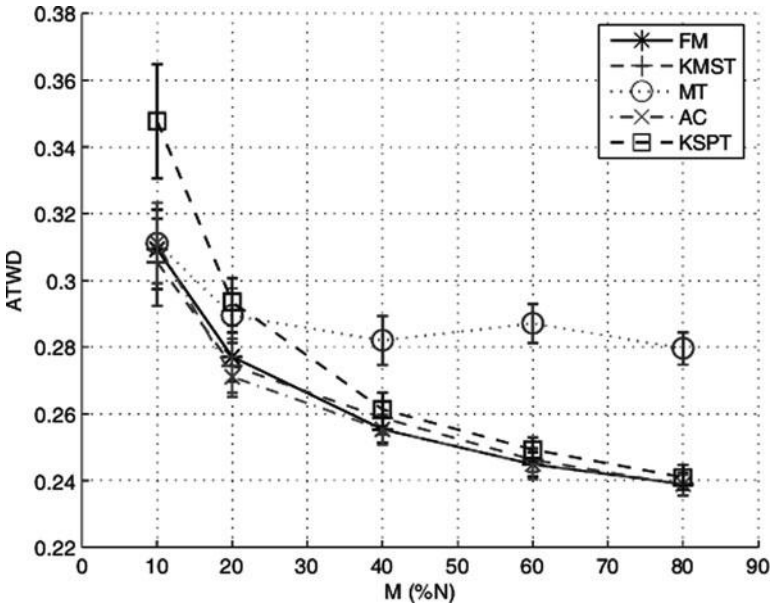


Fig. 14 ATWD for a hierarchical IP-layer network

IP network. The simulation results show that MT is the worse option in every scenarios, because even though its node degree constraint is the lowest, a high number of overlay paths allocation requests is rejected. KSPT is the best option when the topology of the IP network is flat and the number of overlay nodes is small. On the contrary, when the number of overlay nodes is large, the AC topology seems to be the best option. Finally, the simulation results highlight that in case the hierarchical organization of the IP network is also considered, the AC topology is the best option independently of the number of overlay nodes. This result is not surprising, but simply confirms that the knowledge of the underlay IP topology is profitable for the creation of the overlay topology and guarantees best performance.

References

1. LHC Home Page, available at http://www.cern.ch/lhc
2. Comb-e-Chem Project Home Page, available at http://www.combechem.org
3. I. Foster, C. Kesselmann, and S.Tuecke. The anatomy of the Grid: Enabling scalable virtual organizations. International Journal of Supercomputer Applications 15(3), 200–222, 2001
4. D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris. Resilient overlay networks, Proc. of SOSP '01, vol. 35, ACM Press, New York, NY, pp. 131–145, December 2001
5. Z. Duan, Z.-L. Zhang, and YT. Hou. Service overlay networks: Slas, QoS and bandwidth provisioning, in ICNP '02: Proceedings of the 10th IEEE International Conference on Network Protocols, Washington, DC, IEEE Computer Society, pp. 334–343, 2002

6. L. Subramanian, I. Stoica, H. Balakrishnan, and Katz, RH. Overqos: An overlay based architecture for enhancing Internet QoS, Proceedings of NSDI'04, p. 6, USENIX Association, 2004
7. L. Lao, S.S. Gokhale, and J. Cui. Distributed qos routing for backbone overlay networks, Vol. 3976/2006. Lectures notes in Computer Science, Springer, Berlin, pp. 1014–1025, 2006
8. S.L. Vieira, J. Liebeherr. Topology design for service overlay networks with bandwidth guarantees, Proceedings of IEEE IWQoS, pp. 211–220, 2004
9. J. Han, D. Watson, and F. Jahanian. Topology aware overlay networks, Proceedings of INFOCOM 2005, Vol. 4, pp. 2554–2565
10. A. Capone, J. Elias, F. Martignon. Optimal design of service overlay networks, Proceedings of IT-NEWS, pp.1–12, 2008
11. J. Fan, MH. Ammar. Dynamic topology configuration in service overlay networks: A study of reconfiguration policies, Proceedings of INFOCOM '06, 2006
12. H.T. Tran, T. Ziegler. A design framework towards the profitable operation of service overlay networks. Computer Networks, 51(1), 94–113, 2007
13. L. Zhou, A. Sen. Topology design of service overlay network with a generalized cost model, Proceedings of IEEE GLOBECOM' 07, pp. 75–80, 2007
14. M. Kamel, C. Scoglio, T. Easton. Optimal topology design for overlay networks, Vol. 4479/2007, Lectures Notes in Computer Science, Springer, Berlin, pp. 714–725, 2007
15. Z. Li, P. Mohapatra. On investigating overlay service topologies, Computer Networks, 51(1), 54–68, 2007
16. EU DORII Project Home Page, available at http://www.dorii.eu
17. A. Young, J. Chen, Z. Ma, A. Krishnamurthy, L. Peterson, R.Y. Wang. Overlay mesh construction using interleaved spanning trees, Proceedings of INFOCOM '04, pp. 396–407, 2004
18. A. Nakao, L. Peterson, A. Bavier. A routing underlay for overlay networks, Proceedings of ACM SIGCOMM '03, pp. 11–18, 2003
19. Z. Li, P. Mohapatra. Qron: QoS-aware routing in overlay networks. IEEE Journal on Selected Areas in Communications, 22(1), 29–40, 2004
20. Brite Home Page, available at http://www.cs.bu.edu/BRITE/
21. B.D. McBride, C. Scoglio. Characterizing traffic demand aware overlay routing network topologies, Proceedings of IEEE INFOCOM '07, pp. 1–6, 2007

BitTorrent for Storage and File Transfer in Grid Environments

Nikolas Kasioumis, Constantinos Kotsokalis, Pavlos Kranas, and Panayiotis Tsanakas

Abstract Grid Computing is a computing model that treats all resources as a collection of manageable entities with common interfaces to such functionalities as lifetime management, discoverable properties and accessibility via open protocols. It also provides a robust infrastructure for distributed computing, and is widely used in large-scale scientific applications that generate large volumes of data. One of the key factors to those applications is the ability to access very large files in a short -and predictable, if possible– time. This chapter introduces a new architecture, developed to achieve better throughput, more reliable file transfers and better file resilience. The suggested system combines the secure resource sharing paradigm of grid technologies with the qualitative characteristics of peer-to-peer systems for efficiency and resilience. We essentially propose a Grid filesystem and the relevant transport layer, using a slightly modified version of the BitTorrent peer-to-peer protocol, and describe the additional components required to implement it, adhering to de facto Grid Computing best practices. We experimentally find that BitTorrent, compared to GridFTP, is more reliable with regard to completion times and, as expected, more resilient to networking failures.

1 Introduction

Grid Computing has become very popular over the last years, as the need for extended computing resources and data storage has increased especially in scientific experiments and engineering applications. Distributing and providing reliably those resources over a universal network is one of the main scopes of Grid Computing, aiming to ultimately turn the global network of computers into one vast computational resource.

In the Grid Computing paradigm, data may be transparent to the user with regard to its location. It is often considered to be a type of resource itself, therefore we

N. Kasioumis (✉)
National Technical University of Athens, Athens, Greece
e-mail: nikolas@ece.ntua.gr

F. Davoli et al. (eds.), *Remote Instrumentation Services on the e-Infrastructure*,
DOI 10.1007/978-1-4419-5574-6_11,

wish for it to also be virtualized. It may include replicas, be available through *Logical File Names* and thus make its real location irrelevant. As long as a set of data can be referenced through a unique identifier, it is not even important if all the data is concentrated at one place, or scattered over a distributed infrastructure. With the ever growing volume of data produced from Grid-enabled experiments and applications, the question of efficient handling for storage and transfer is very timely.

A *Peer-to-Peer* (P2P) computer network is based on a communications model where each participant is considered to be equal to its peers, providing and consuming computing power, storage and bandwidth to and from the network. The decentralized nature of P2P networks allows for increased robustness in case of failures, by distributing resources and replicating data over peers. BitTorrent, in particular, is a peer-to-peer communications protocol and file distribution system that uses tit-for-tat as a method of seeking Pareto efficiency, thus achiving high levels of resiliency and resource utilization [1]. With BitTorrent, the initial provider of a file collection can avoid incurring the entire costs of hardware, hosting and bandwidth resources. More specifically, one descriptive file containing metadata, known as a *torrent* file, is used for each chunk of data (file collection), including among others the address of the *tracker*, which is a minimal central point of control holding information on the peers distributing the data and coordinating the traffic among them. Information about all the nodes serving the file (*seeds*) and the nodes retrieving it (*leechers*) is provided by the tracker. BitTorrent implements various algorithms aiming to provide reliable, robust and fair data distribution between peers, but unlike Grid-based data transfers, it cannot currently guarantee secure communications.

This chapter aims to describe a conceptual design that provides computing Grids with a filesystem and a transport layer based on BitTorrent, thus exploiting its potential in terms of accessing and transferring data, while also considering some of the Grid's security aspects. Our architecture implements the four essential functions of persistent storage: Create, Read, Update and Delete (CRUD). We do not, however, address issues of concurrency at this stage.

The proposed system ensures all the security mechanisms guaranteed by a Grid network for signalling purposes: authentication, authorization and privacy. Data transfers can be enabled with privacy, by using clients that support the necessary encryption mechanisms. In order to establish the advantages of BitTorrent over the typical point-to-point transfer mechanisms of the current Grid, experiments have been conducted to evaluate reliability and efficiency qualities in relation to GridFTP [2]. The discussion regarding the trade-off between storage capacity utilisation and performance/resilience is not relevant here, as the Grid facilitates replicas of data already [3].

The rest of the chapter is organized as follows: Related work is presented in Section 2; the requirements for Grid Computing are stated in Section 3; evaluation experiments are described and commented in Section 4; in Section 5 our architecture is analyzed and directions for future work are presented in Section 6. We conclude in Section 7.

2 Related Work

Due to BitTorrent's extended usage for the transfer of large files and the increasing share of P2P-related Internet traffic over the last few years, there have been various studies analyzing and monitoring its performance, suggesting ways to improve it by adapting it to the corresponding circumstances. In [4] Bharambe, Herley and Padmanabhan show how simple changes to the tracker and a stricter tit-for-tat policy greatly improve fairness; Gkantsidis and Rodriguez propose a new scheme for content distribution of large files that is based on network coding in [5], while in [6] we read about BitTyrant, a strategy-enabled BitTorrent client that provides a significant performance gain over a typical client; In [7–11] we find thorough studies and analysis concerning BitTorrent's performance.

Studies addressing GridFTP have been conducted as well; in [12] Rizk, Kiddle and Simmonds evaluate a performance boost in GridFTP by splitting TCP connections at one or more points of the network, while in [13] the optimal parameter configuration of GridFTP in terms of the number of TCP connections and the TCP socket buffer size is investigated.

Getting to even more closely related studies, in [14], where an algorithm using a circular queue causing the data transfer tasks to be allocated into the servers in duplication is developed, BitTorrent is examined as a potential download method for a Grid network. In [15], although not specifically in the context of Grid computing, the authors present a peer-to-peer system for content replication, with build-in optimization techniques that enhance the overall system and improve resource utilization. Finally, in [16] the authors present a prototype that uses BitTorrent as a data management subsystem for any Grid Desktop System, based on evaluation experiments comparing BitTorrent to a classic FTP server. Unlike the architecture and evaluation presented in our paper, in [16] the evaluation is carried out only in a high-end LAN cluster using a non-threaded, non-parallel classical FTP server, examining mostly the dominating performance gain of BitTorrent over mass requests for pieces of data. In addition BitTorrent is evaluated as a data management subsystem collaborating with FTP transfer methods and not as a fully functional filesystem for a Grid network. However, some quite interesting and important conclusions can be drawn from the above study. It is clearly demonstrated that in a dedicated LAN environment BitTorrent outperforms an FTP server even in the existence of a small number of peers. BitTorrent can easily scrape through against a flash-crowd effect and cope with an increasing number of peers and large amounts of data. As such, BitTorrent is not only performing better, but also constitutes a more scalable architecture.

3 Requirements for Grid Network Data Transfers

Data transferred across Grid networks may reach sizes of hundreds of gigabytes or even terabytes, products of scientific experiments or other large-scale computations. These resources are requested by members of one or more *Virtual Organisations*

(VOs), which may be on the same LAN or WAN, or on an even more universal network, depending on the territorial attributes of the VO. A Grid network should ensure the data's constant availability, as well as provide some kind of loose efficiency guarantees with regard to data access and transfer. On top of that, privacy should be taken into account, by ensuring authorized and authenticated data access. Apart from these characteristics, some form of indexing facilities should be able to provide information on files, considering their status and attributes, to both network clients and network services, aiming to improve the data's availability.

We assume that we have a variety of *Storage Elements* (SEs) on a Grid network, which are part of a single or different VOs, and they all request access to data files. The data files are stored in any of those storage elements. The actions they are allowed to invoke is to request a file, to store a file, to completely delete a file and to update a file. There is also the need of maintaining the privacy, the integrity, the authentication and the authorisation of every action. Moreover it must always be checked if the storage element is authorised to undertake the desired action according to its status and the VO that it belongs to.

BitTorrent's dominating performance over well-established LAN networks, and over mass requests for data has been clearly demonstrated in related work, as mentioned in Section 2. However, Grid computing has its footing on the hypothesis that resource availability and performance cannot be taken for granted. Thus, we would like to investigate what happens in a less consistent Grid environment, where peers and participants belong to completely different networks connected through the Internet. In this case, a Grid network should provide statistical certainty over some acceptable threshold, with regard to data availability and data transfer performance. We tested the hypothesis that BitTorrent is more suitable than GridFTP in such an environment, and preliminary experimental results are presented in Section 4.

4 Experimental Results

4.1 Setup

For our evaluation experiments, in lack of a more stable testbed platform and with the intention to simulate a real world universal environment, we used the Planetlab global research network [17]. We randomly chose 21 functional nodes[1] (of unknown, random hardware characteristics) from around the world in various universities and institutes: United States of America, Japan, Venezuela, Switzerland, Germany, Poland, Italy, Spain, Puerto Rico, Norway, Austria, Belgium, United Kingdom. Each node was based on a minimal Fedora Core Linux [18] installation which we enriched with the necessary dependencies and prerequisites for our

[1] We simulate the case of large capacity storage elements, thus we need a small number of nodes. Choosing a larger number of nodes would signify an intention to use small capacity peers (in terms of storage) which is unwanted, as we plan to transfer large scale data without splitting it into smaller fragments due to capacity constraints on the storage elements.

end-user utilities: the Azureus BitTorrent client along with its embedded tracker [19] and Globus Toolkit version 4 [20]. The Java Development Kit was used, and the Azureus client and tracker were controlled through console and/or web interface after setting up their corresponding plugins and configurations.

4.2 Procedure

Due to Planetlab's daily bandwidth restrictions, we ran a relatively short series of evaluation experiments, though capable of providing sufficient results to establish our assumptions. In all the experiments a file sized 3 GB (3072 MB) was used, representing, in perspective, large-scale data transfers over Grid networks. The metric measured in all cases was the throughput, from the consumer (downloader) point of view.

The first round of experiments concerns BitTorrent. We gradually spread a file through 20 of our nodes, using our 21st node only as a tracker (in order to avoid Planetlab's bandwidth limitations in our tracker). During that period of time we recorded each node's throughput, and used it as our criteria for selecting the nodes to be our final testing downloaders. We then defined 16 "average-throughput" nodes as our seeds (file providers / uploaders) and the remaining 4 high-throughput nodes as our leechers (file requesters / downloaders). We chose to run the experiment using 16 seeds and 4 leechers in order to have an analogy with GridFTP's 4 parallel connections per download and thus match other similar past experiments [21]. The experiment was run 5 consecutive times.

The second round concerns GridFTP-based transfers. For reasons of maximum fairness towards GridFTP, we selected the same 4 nodes with the best performance as our GridFTP servers. One of the 4 server nodes became unavailable during the time of the experiments, so it is not included in the figures. We then selected 10 nodes having noticeable upload performance as our clients and supplied them with the same file as before, sized 3 GB. Each GridFTP server downloaded the file 2 times from each client using the globus-url-copy command and 4 parallel threads. The experiments took place on the same time of the day as the BitTorrent experiments, assuming similar background traffic characteristics of the intermediate network links.

4.3 Results and Discussion

The outcome of the BitTorrent downloads is shown in Figs. 1 and 2. Figure 1 shows the results for each node and each of the five consecutive runs. Figure 2 shows the average values of all runs for these nodes.

Throughout the experiments we observed that as soon as the first leecher completed its download, thus becoming a seed, the rest of the leechers obviously increased their download speed, due to the existence of one more seed, and the improved sharing of the seeds' upload capabilities. While a lot of seeds suffered

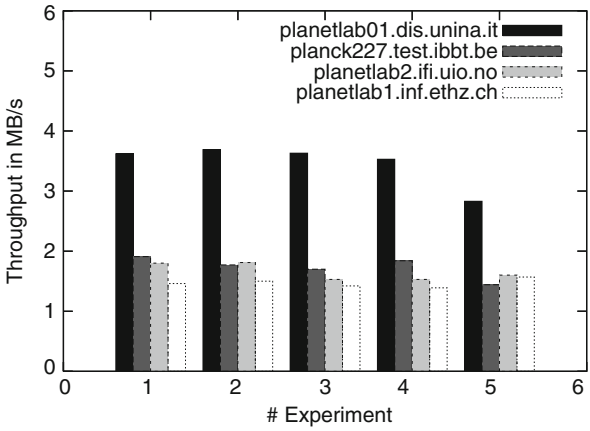


Fig. 1 BitTorrent experiments

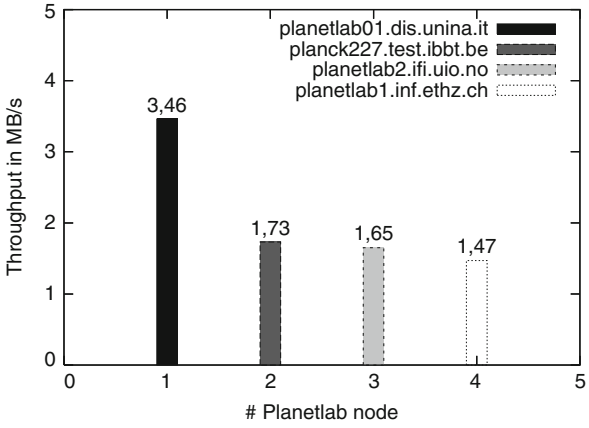


Fig. 2 Avg. transfer rates for all BitTorrent experiments

from bandwidth and performance limitations due to Planetlab's nature and the network's random behavior, the leechers managed to pull through obtaining and maintaining high throughputs to meet our performance expectations for the transfer. The average throughput achieved per transfer in all experiments with BitTorrent has been very consistent, as it is clear from the first figure. The fluctuation was minimal, providing a very good statistical profile.

For the GridFTP transfers, the average throughputs are presented in Figs. 3 and 4.

In every transfer, each server was downloading from one client only, thus exploiting the connection's full available bandwidth. Moreover, the TCP buffer size was manually configured for each transfer, ideally chosen using the Globus Toolkit documentation rules, based on selected values for bandwidth and *Round-Trip Time* (RTT). More specifically, the documentation states:

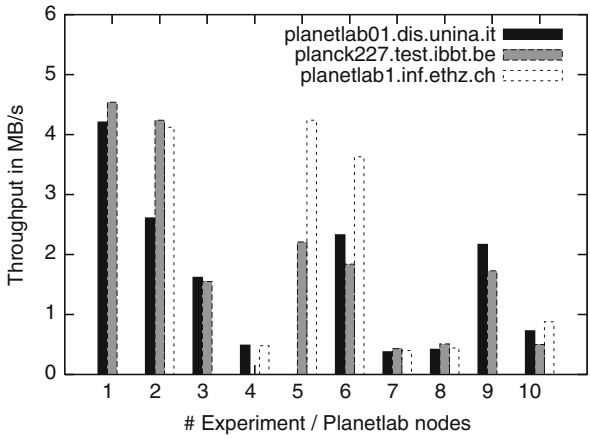


Fig. 3 GridFTP experiments, avg. value of two runs

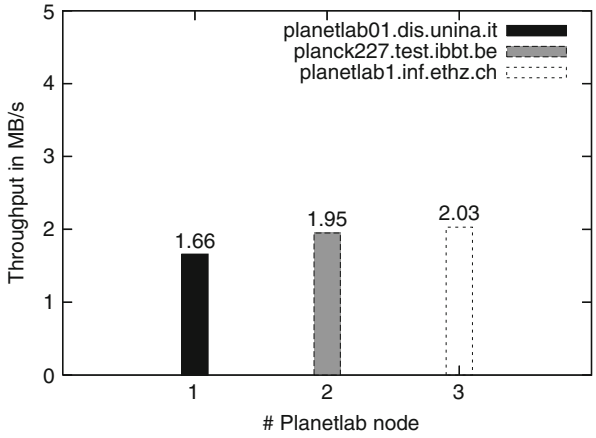


Fig. 4 GridFTP experiments, avg. value per server node

> Any point in the network can be the bottleneck, so you either need to talk with your network engineers to find out what the bottleneck link is or *just assume that your host is the bottleneck and use the speed of your network interface card (NIC)*.

As such, we chose a bandwidth of 100 Mbps, equal to the host's NIC. For RTT, we selected values between 10 and 350 ms, depending on the values returned by "ping" commands.

We observed that GridFTP performance was fluctuating significantly, depending on each server-client couple used. Thus, the reliability of GridFTP with regard to its average throughput and the corresponding completion time is limited. In addition, due to the random malfunction of one server, the reduced resiliency of GridFTP-based transfers was more evident, unlike a peer-to-peer network like BitTorrent where the case for decentralisation is clearly stated.

These measurements present another aspect of a Grid network, that of a more dispersed one. In such a network we observe that BitTorrent can provide a more reliable performance at all cases, even when coping with less stable peers, and maintain a more predictable and efficient throughput. In contrast, GridFTP is directly affected by the network's local and temporary conditions.

5 System Architecture

As it is indicated by the experiments, the use of the BitTorrent protocol can solve certain problems related to the transportation of large data files over a network in real-life environments. For all the reasons described in the previous section, we propose a design for the extension of a Grid ecosystem so as to use the BitTorrent protocol for data transfers, instead of the commonly used GridFTP. We describe an architecture that serves as a file system on a Grid network and utilizes BitTorrent clients whenever the need for data access occurs.

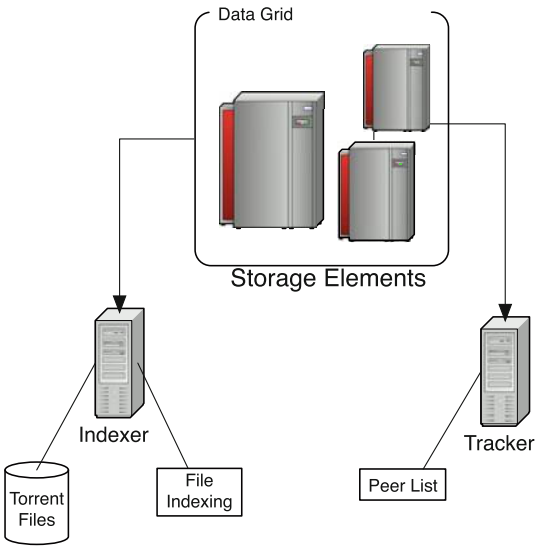
We described the requirements of the scenario we are looking into, in Section 3. There is the need for a central or distributed service that invokes the necessary actions for the use of BitTorrent according to the requests from the storage elements. There is also the need for a component serving as the tracker for the implementation of the BitTorrent protocol.

Our proposed system consists of a main indexer service, a tracker service and several storage elements. The SEs may be anything from a high-end storage resource to a simple personal computer. The indexer service keeps track of all the files stored on the system and manages information on who stored the files, which VOs are authorised to access them and which actions are allowed to be invoked. It also holds all the corresponding torrent files for anything stored in the system. When a SE has to request, store, delete or update a file, it sends a message to the indexer and waits for a response. The indexer verifies whether the SE is authorized for the desired action, and responds accordingly. It then activates the corresponding process. The overall architecture is illustrated in Fig. 5. The processes for those four actions are described in detail below.

Get action: (Fig. 6) A peer requests a file from the indexer. The indexer examines whether the peer is authorized, and if so, it grants the request and sends back the corresponding torrent file to the peer. The peer starts the file sharing procedure as specified by the BitTorrent protocol. After finishing, it notifies the indexer. The indexer is updated on the altered status for this dataset.

Put action: (Fig. 7) A peer requests permission from the indexer to add a new file to the network. If authorized, the peer creates a corresponding torrent file and sends it over to the indexer upon granted permission. The peer starts seeding the file as specified by the BitTorrent protocol. The indexer randomly chooses peers to share the file and sends them copies of the corresponding torrent file as well as a signal to start requesting the file.

Fig. 5 System components



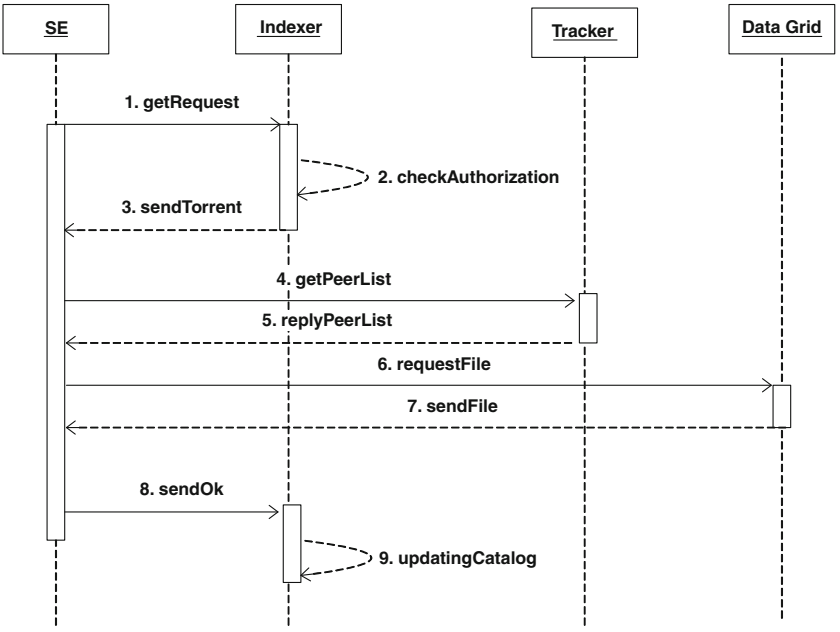


Fig. 6 " Get" action

Delete action: (Fig. 8) A peer requests the deletion of a file. The indexer checks if the peer is authorized for this procedure. The peer stops seeding the file upon successful authorization. The indexer informs the rest of the file's seeds to stop seeding too and delete the file .

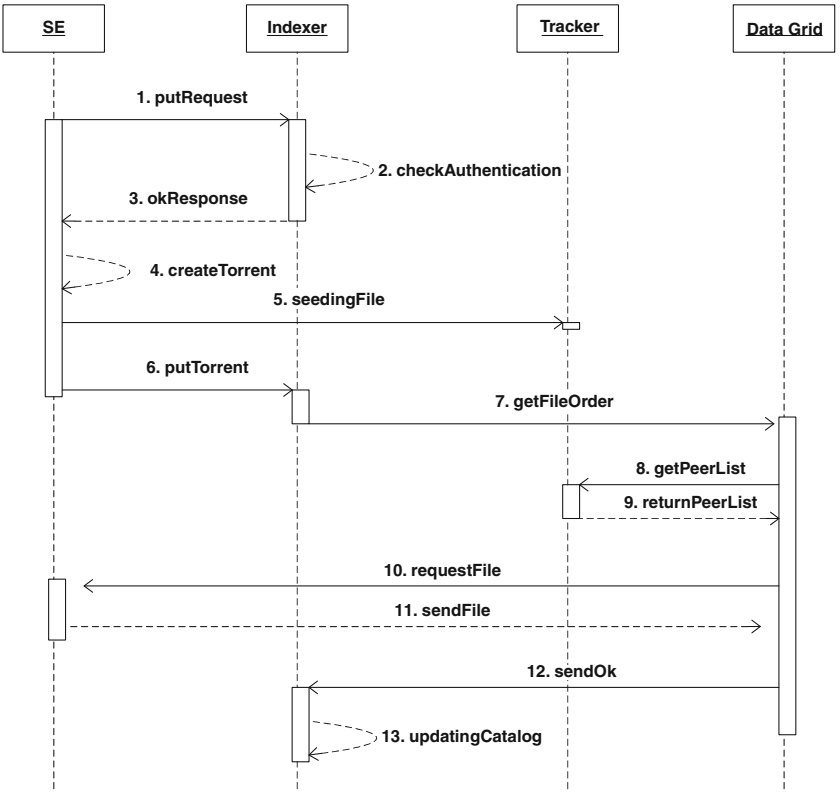


Fig. 7 " Put" action

Update action: (Fig. 9) The update action can be analysed as the sequential combination of a delete and a put action.

The indexer maintains a list of all the files stored in the system, and the nodes that have obtained them. The nodes are categorized into "complete" and "pending". When a peer starts downloading a file, a new record is added to the list and the peer is characterized as "pending". When it sends a confirmation signal to the indexer, its status is changed to "complete". Every time a delete action occures, the indexer sends delete signals to all nodes that have stored it, whether they have a pending or a complete status.

It needs to be mentioned that every SE has an edge-Web Service that sends and receives messages from the indexer, and acts according to indexer-originating signals. Moreover, there is a BitTorrent client in every such device which can be controlled by the classes of our system via well-defined interfaces. It must also be underlined that every communication between a SE and the indexer is based on message-level security, while the transfer of the torrent files and datasets is based on

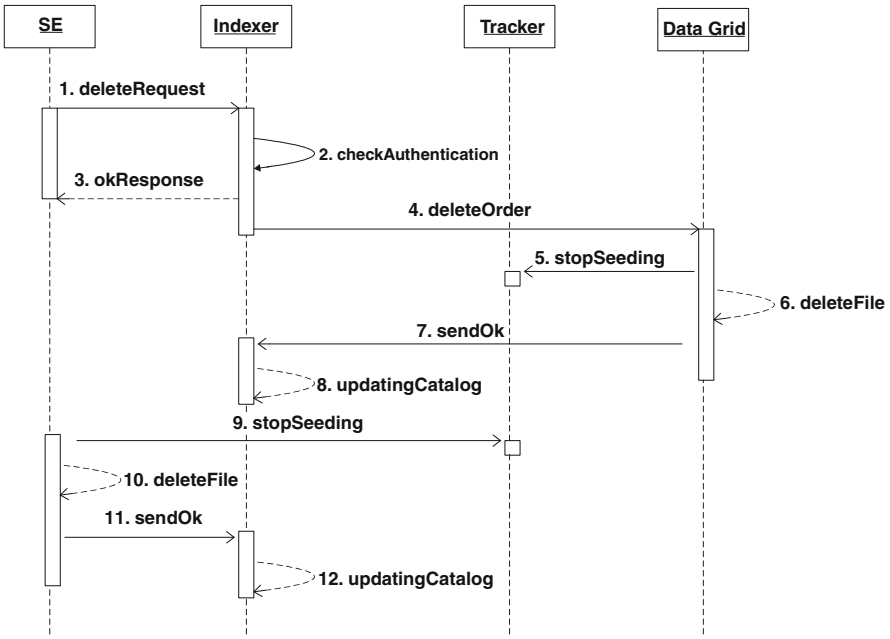


Fig. 8 " Delete" action

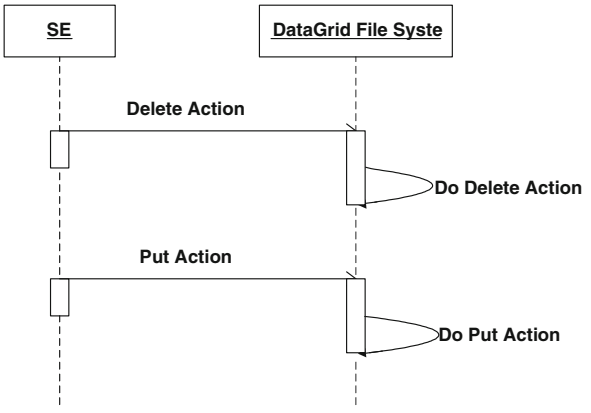


Fig. 9 " Update" action

transport-level security. Both message-level and transport-level security in GSI are based on public-key cryptography and, therefore, can guarantee privacy, integrity and authentication.

This middleware can be put to best use when there is a requirement of data replicas and there are no significant storage space constraints, although as mentioned earlier, data set copies are already foreseen and allowed in the Grid. Replicas ensure data recovery and moreover, they contribute to the proper use of the network. We

suggest the use of a popularity variable, which indicates how many times a single file is being accessed. The files are categorized, according to their popularity. The number of the replicas needed to be stored on the network is based on the corresponding popularity. At every *get* request, the indexer creates a record with the name of the requested file and a time stamp. Moreover, there is a process running as a daemon, which examines the timestamps and categorises every file according to its last access time, then changes its popularity value. In cases of high popularity, the daemon informs the indexer which sends get orders to more nodes to download the file. This increases the number of replicas stored on the network. As it has been proven, the BitTorrent protocol works better and achieves better performance with more seeds [22]. More specifically, when the number of the participating seeds is increased, the searching algorithms that are implemented by the BitTorrent protocol always pick the most appropriate ones so as to achieve even better performance [9]. In cases of low popularity, the daemon informs also the indexer which, in turn, sends delete orders to nodes. The number of replicas is decreased and more storage space is made available on the network. After the completion of a get/put/delete/update request, the corresponding node announces its available data resources to the indexer. The indexer stores this information in its catalog. It is possible for the indexer to make use of additional performance information related to the network, from services such as the *Network Weather Service* [23]. Information of this kind can be very useful for reasons of matchmaking and optimal peer selection by the indexer, taking into account those peers' networking capacity and performance.

6 Future Work

We have described a system which serves adequately as a new file system for data grid networks and benefits from the underlying BitTorrent protocol. As it is confirmed from the figures in previous sections, this system achieves higher and more consistent throughput at the transportation of large files between nodes that exist in remote locations. It is possible to achieve even better performance, taking into account two crucial parameters such as the popularity of every file and the network conditions. Furthermore, it is evident that reliability of data transfers with regard to the estimated or expected time-to-finish is much higher when using distributed file transfers like the one proposed in this paper.

The description of this system is being based on a central index service and a single tracker. As a result there are two single points of failure. The problem can be solved using distributed index services alongside more trackers. The use of the BitTorrent protocol with multiple trackers has been described in [24]. Distributed indexers achieve even better performance based on the fact that nodes communicate with the most suitable candidate. The exchange of replicated data between indexers is not a problem, as these files are very small and their transfer requires minimal resources from the network. As such, these transfers do not affect the measurements.

Another topic which arises from the analysis of a system like this, which effectively describes a distributed file system, is the matter of concurrency. Relevant

issues will be addressed in future work, possibly employing a two-phase-commit protocol or database-like transactions.

A more extensive set of experimental measurements that will enhance our conclusions on the advantages of utilizing the Bit Torrent protocol is required. As of now, we have measured the download times among a constant set of seeds and leechers. In real life conditions, nodes may decide to delete a file due to capacity problems or following a delete signal from the indexer, making file availability change dynamically. Hence, the amount of seeds and leechers participating in a file transfer must be experimentally altered during this process. Finally, an extension of the current prototype towards a complete implementation of the proposed system will follow.

7 Conclusions

In this chapter we presented the subject of data distribution in a Grid network using peer-to-peer technologies, taking into consideration the relevant security aspects. We analyzed the requirements of a new file system that manages the data stored across the network. We showed experimentally that using BitTorrent for file transport is more efficient on average, while at the same time it provides greater reliability with regard to file transfer time-to-finish. We then suggested a new architecture for accessing large scale files on a Grid network, based on the BitTorrent protocol. Four basic actions were described in order to implement the CRUD principles for file systems. Remaining gaps were then stated, providing directions and grounds for future work.

References

1. B. Cohen. Incentives build robustness in BitTorrent. Workshop on Economics of Peer-to-Peer Systems, 2003
2. Globus GridFTP: http://www.globus.org/grid_software/data/gridftp.php, accessed December 2008
3. A.L. Chervenak, N. Palavalli, S. Bharathi, C. Kesselman, R. Schwartzkopf. Performance and scalability of a replica location service. 13th IEEE International Symposium on High Performance Distributed Computing, June 4–6, Honolulu, Hawaii, USA, 182–191, (2004)
4. A. Bharambe, C. Herley, V. Padmanabhan. Analyzing and improving bittorrent performance. Microsoft research technical report msr-tr-2005-03, Carnegie Mellon University and Microsoft Research, 2005
5. C. Gkantsidis, P.R.: Rodriguez. Network coding for large scale content distribution. INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE 4, 2235–2245, 2005
6. M. Piatek, T. Isdal, T. Anderson, A. Krishnamurthy, A. Venkataramani. Do incentives build robustness in BitTorrent? Technical report, University of Washington Computer Science & Engineering, 2006
7. D. Arthur, R. Panigrahy. Analyzing BitTorrent and related peer-to-peer networks. SODA '06: Proceedings of the 17 annual ACM-SIAM symposium on Discrete algorithm, New York, NY, ACM, 961–969, (2006)

8. M.R. Ceballos, J.L. Gorricho. P2P file sharing analysis for a better performance. ICSE '06: Proceeding of the 28th International Conference on Software Engineering, New York, NY, ACM 941–944, (2006)
9. D. Qiu, R. Srikant. Modeling and performance analysis of BitTorrent-like peer-to-peer networks. SIGCOMM '04: Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, ACM New York, NY, 367–378, (2004)
10. A.A. Hamra, P.A. Felber. Design choices for content distribution in P2P networks. SIGCOMM Computer Communication Review, 35(5), 29–40, (2005)
11. J.A. Pouwelse, P. Garbacki, D.H.J. Epema, H.J. Sips. The Bittorrent P2P file-sharing system: Measurements and analysis. 4th International Workshop on Peer-to-Peer Systems (IPTPS), February 24–25, Ithaca, NY, USA, 2005
12. P. Rizk, C. Kiddle, R. Simmonds. Improving gridFTP performance with split TCP connections. E-SCIENCE '05: Proceedings of the 1st International Conference on e-Science and Grid Computing, Washington, DC, IEEE Computer Society 263–270, (2005)
13. T. Ito. Understanding and optimizing gridFTP: Leveraging parallel data transfer for long-fat networks. Master's thesis, Graduate School of Information Science and Technology, Osaka University, 2006
14. R.S. Bhuvaneswaran, Y. Katayama, N. Takahashi. Redundant parallel data transfer schemes for the grid environment. ACSW Frontiers '06: Proceedings of the 2006 Australasian Workshops on Grid Computing and e-Research, Darlinghurst, NSW, Australian Computer Society, 71–78, (2006)
15. L. Caviglione, C. Cervellera. Design of a peer-to-peer system for optimized content replication. Computer Communications, 30(16), 3107–3116 Special Issue: Advances in Communication Networking, (2007)
16. B. Wei, G. Fedak, F. Cappello. Collaborative data distribution with BitTorrent for computational desktop grids. ISPDC '05: Proceedings of the The 4th International Symposium on Parallel and Distributed Computing (ISPDC'05), Washington, DC, IEEE Computer Society, 250–257, (2005)
17. Planet Lab: An Open Platform for Developing, Deploying and Accessing Planetary-Scale Services: https://www.planet-lab.org/, accessed December 2008
18. Fedora Core Linux: http://fedoraproject.org/, accessed December 2008
19. Azureus BitTorrent client: http://azureus.sourceforge.net/, accessed December 2008
20. Globus Toolkit: http://www.globus.org/toolkit/, accessed December 2008
21. B. Kónya, O. Smirnova. Performance Evaluation of the GridFTP wihtin the NorduGrid Project. Technical report, NorduGrid, 2001
22. A. Bharambe, C. Herley, V. Padmanabhan. Analyzing and Improving a BitTorrent network's performance mechanisms. IEEE Conference on Computer Communications (INFOCOM), April 23–29, Barcelona, Catalonia, Spain, 2006
23. R. Wolski, N.T. Spring, J. Hayes. The network weather service: A distributed resource performance forecasting service for metacomputing. Future Generation Computer Systems, 15(5–6), 757–768, (1999)
24. G. Neglia, G. Reina, H. Zhang, D. Towsley, A. Venkataramani, J. Danaher. Availability in BitTorrent systems. IEEE Conference on Computer Communications (INFOCOM), May 6–12, Anchorage, Alaska, USA 2007

Context Awareness in Autonomic Heterogeneous Environments

A. Zafeiropoulos and A. Liakopoulos

Abstract The concept of self-managing of autonomic networks is a paradigm shift from today's management models, aiming at enabling networked nodes to self manage their behaviour within the constrains of the operator's policies and objectives. The incorporation of the autonomicity concept to communications systems makes necessary the development of extensible context models. These enable the efficient representation of available information, needed for handling and distributing it. In this chapter, we focus on increasing context awareness towards the reduction of complexity in the management of multiple mechanisms realised in autonomic networks, giving special attention in managing, maintaining and exploiting autonomic entities in a unified way. An ontology is proposed that complies with the principles of a recently introduced Reference Model for autonomic network engineering/self-management within node and network architectures dubbed: the Generic Autonomic Network Architecture (GANA), which aims to identify autonomic behaviours realised via hierarchical control loops among self-managing elements.

1 Introduction

It is now common knowledge that IP networks are increasingly becoming larger and more complex to monitor or fully manage in a cost-effective way. The vision of the Future Internet, is of an autonomic network whose nodes are engineered in such a way that all the traditionally so-called network management functions defined by the FCAPS (Fault, Configuration, Accounting, Performance, Security) management framework [1], as well as the fundamental network functions such as routing, forwarding and monitoring, are designed to automatically feed each other with information and effect feedback processes among the diverse functions. The feedback processes enable reactions in functions of the network and of individual

A. Zafeiropoulos (✉)
Greek Research and Technology Network, Av. Mesogion 56, 11527, Athens, Greece
e-mail: tzafeir@grnet.gr

F. Davoli et al. (eds.), *Remote Instrumentation Services on the e-Infrastructure*,
DOI 10.1007/978-1-4419-5574-6_12,

nodes, in order to achieve and strive to maintain some well defined goals of the network.

Autonomic principles are required in order to accomplish management of dynamic, heterogeneous and complex networks, where each network entity needs to be able to take network optimisation decisions. Predefining the monitoring scheme can be inefficient in heterogeneous environments, taking into consideration the constant changes in the network topology and the diversity of the interconnected systems. Therefore, the monitoring components within an autonomic network should be continuously adapted, in a flexible manner, to an ever changing network infrastructure (self-configuration). Flexibility or adaptability is considered as one of the most important design principles that demands different kinds of awareness both in the ends and in every node in the service supplying chain, while targeting users' satisfaction as the final goal of any management process [2].

Thus, it is clear the need for the inclusion of capabilities such as self-awareness, self-network knowledge and self-service knowledge to facilitate adaptation to unpredictable conditions and recovery from failures [3]. These capabilities can be introduced through the increase in the context-awareness level of an autonomic network. By increasing the context-awareness of monitoring data exchanged by autonomic nodes it is possible to efficiently sense network conditions and the level of provided services and proceed to corrective actions (self-healing) in case of identified problems. Also, context-aware information is easily used in order to take decisions according to the specified administrators' policies (self-optimization).

A Future Internet capable of embracing this concept and delivering context aware services to users and artefacts elevates this to a pervasive sensing and acting knowledge network. This would be a network able to make decisions, actuate environmental objects and assist users [4]. But, the incorporation of autonomicity to communication systems requires the efficient representation of the available information. Extensible context models have been developed in order to handle and distribute this information properly. Furthermore, ontology engineering [5] has been proposed as a formal mechanism for (i) reducing the complexity of managing the information needed in network management and autonomic systems and (ii) for increasing the portability of the services across homogeneous and heterogeneous networks. Learning and reasoning techniques have to be used to support intelligent interactions among the specified entities. Ontologies will function as core components for automated analysis of enterprise-wide event data, will be based on user-defined rules, trigger corrective actions for healing the system, deal with policy based goals on a higher abstraction level and provide new levels of functionality.

In this chapter we present the approach that is designed in the framework of the EFIPSANS-IP-Project [6] for implementing context-aware autonomic networking functionalities and we describe in detail the deployed ontology. The EFIPSANS-IP-Project that started in January 2008, aims at exposing the features in IPv6 protocols that can be exploited or extended for the purposes of designing or building autonomic networks and services. The proposed approach is based on the recently proposed Generic Autonomic Network Architecture (GANA) [7]. GANA sets the principles and guidelines that need to be followed according to our vision of the

Future Internet design. In contrast to any other of today's best known approaches, including clean-slate approaches, such as 4D [8], CONMan [9], a Knowledge Plane for the Internet [10], FOCALE [11], a Situatedness-based Knowledge Plane for autonomic networking [12], GANA captures in a holistic way, the generic principles required for a generic autonomic network architecture. Information about the current network state is being collected according to the GANA information model and the use of a domain-specific ontology add relationships between context data and the behaviours of the autonomic nodes, thus allowing them to reason about the modelled information and interact in a way that accurately reflects the overall network behaviour.

The chapter is organised as follows; section 2 presents definitions of the basic concepts of context awareness and the related work in autonomic networks; section 3 describes in brief the fundamental concepts of the GANA architecture; section 4 details the approach for providing context awareness in EFIPSANS and the designed GANA ontology and finally section 5 concludes the chapter with a discussion of current challenges and future work.

2 Related Work

2.1 Context and Context Awareness

Context information is complex, dynamic and heterogeneous. Context comprises of contextual information which may be retrieved from heterogeneous sources and is defined as any information that characterizes the situation of an entity (place, person, object). Users as well as devices and network entities can have different forms of context, and support multiple "profiles" that express their needs, presence, and other contextual information [13, 14]. Combination of context data from one or more sources may denote the existence of a specific situation. According to [15] situation is "the state of a context at a certain point (or region) in space at a certain point (or interval) in time, identified by a name". Consequently we can consider that situation can be derived from aggregating and refining types of context information. By recognizing specific events and situations in a networking environment, knowledge about the current conditions can be extracted, leading to context awareness.

Context awareness refers to the capability of an application, service or even an artefact being aware of its physical environment or situation and responding proactively and intelligently based on such awareness. Context-aware applications, context-aware artefacts or context aware systems are aware of their environment and circumstances and can respond intelligently [4]. The notion of context awareness is very important in autonomic environments where the networking entities have to sense the changes in the network conditions and proceed to specific actions in order to support autonomic functionalities (self-adaptation, self-optimisation, self-configuration). The autonomic network has also to adapt the services and resources that is offering to the changing demands of the user, as well as adapt to changing environmental conditions, thus helping to manage business, system, and behavioural

complexity. Especially in heterogeneous environments, the efficient handling and distributing of context data and knowledge to support context-awareness is an important functionality [13].

Context has firstly to be acquired from the lowest level networking entities, then be aggregated to more capable entities and finally be available and disseminated to the autonomic network. Context dissemination is the propagation of context to other entities. Special functionalities are being designed for efficient discovery and exchange of context data among different networking entities. Since we are referring mainly to heterogeneous environments, query and dissemination mechanisms have to be used in order to make context available to other entities. Context has to follow specific format and be mapped to context models in order to have special meaning. These models are sophisticated data structures, which are populated by abstracting and representing contextual information for further processing and observation of certain events and situations [14].

In addition to proper context modelling, collaboration of different entities is necessary to provide information on remote events, associate network conditions and make conclusions for specific situations in the network. The correlation of observations at multiple observation points in the network is essential to get a network wide view that provides the basis to decide, based on the current state of the network [16].

Finally, reasoning mechanisms are applied in higher level for recognizing specific situations in the network environment according to the representation followed by the applied context model. Reasoning is also used to check the consistency of context and context models [4]. Several information models and ontologies have been defined for addressing this issue in autonomic networks, as it is described in detail in the following section.

2.2 Information Models and Ontologies for Autonomic Networks

Representation of context data can be realised through various formats from databases and XML files to more advanced formats such as information models or ontologies. The outcome of each representation scheme in autonomic networking is the extraction of knowledge through the efficient representation of data and support of self-monitoring, self-diagnosing, and self- adapting functionalities. Thus, it is crucial to select the proper representation scheme according to the requirements imposed by the network and the applications scope.

According to [17], an information model is a representation of concepts, relationships, constraints, rules, and operations to specify data semantics for a chosen domain of discourse. The advantage of using an information model is that it can provide sharable, stable, and organized structure of information requirements for the domain context. Different mappings (data models) can be derived from the same information model. But, data models are usually specific to the application(s) for which they have been used or created and are not intended a priori nor considered to be shared by other applications [18].

In communication networks, the following information models are being used for representing the relationships and capturing the behaviours of the network entities: (a) the Common Information Model (CIM) [19] that is a common definition of management information for systems, networks, applications and services, (b) the Shared Information and Data Model (SID) [20] that provides business and system view definitions for designing and managing a telecommunications network, and (c) the Directory Enabled Network-new generation (DEN-ng) policy model [13] that provides an abstraction and representation of the entities in a managed environment and describes the business, system, implementation and runtime aspects of managed entities and their relationships in a simple but comprehensive form.

In addition to information models several ontologies have been designed for providing context-awareness in autonomic networks. Knowledge extracted from information/data models forms facts, while knowledge extracted from ontologies is used to augment the facts, so that they can be reasoned about [3]. It is important to note that, current research indicates, that ontologies are the most expressive context representation models [21, 22].

Ontologies provide a powerful paradigm for context modelling offering rich expressiveness and supporting the dynamic aspects of context awareness [23]. An ontology is an agreement about a shared, formal, explicit and partial conceptualization that contains the vocabulary and the definition of the concepts and their relationships for a given domain, including the instances of the domain as well as domain rules that are implied by the intended meanings of the concepts [18]. An ontology must be explicit, formal and open. Explicit means that the entities and relationships used are precisely and unambiguously defined in a declarative language suitable for knowledge representation, formal refers to the fact that the ontology should be representable in a formal grammar and open means that all users of an ontology will represent a concept using the same or equivalent set of entities and relationships [5]. Furthermore, ontologies, unlike specific and implementation-oriented data models, should be as generic and task-independent as possible [18].

Several research projects have recently proposed the use of context-aware information and ontologies for improving monitoring mechanisms. In FOCALE [11], ontologies contribute towards the realisation of dynamic control loops and the specification of different policies in cases of changes to the network. MOMENT [24] has proposed an ontology for network monitoring tools so as to have a consistent and ordered way to produce information. In AutoI [25], the AutoI Information Model (AIM) with the corresponding ontology is designed that represent a virtual communication resource overlay with autonomic characteristics that is used to adapt the services and resources offered to meet changing user needs, business goals, and environmental conditions. In OASIS [18], ontology-based management and modelling techniques are used and referenced in the framework of a distributed context handling and delivery system in next generation autonomic networks. OASIS uses a global domain ontology (OSM) that captures the knowledge around context information and includes a vocabulary of terms with a precise and formal specification of

their associated meanings. In CONTEXT [5], the synergy obtained between context-awareness, ontologies and autonomic networking promotes the definition of a new, extensible, and scalable knowledge platform for the integration of context information and services support. Finally, in EMANICS [26] a new concept for context models is presented following the policy based management paradigm, suitable for representing information and managing services in cross-layered environments.

The integration of ontologies is an important task in the ontology development area and can speed up the development of new extensible and powerful ontologies in the future [5]. In EFIPSANS we are taking in account existing ontologies in the field of autonomic networks and we specify our context representation models that are based on the Generic Autonomic Network Architecture (GANA), while incorporating parts of ontologies from related projects, as we describe in detail in Section 4.

3 A Brief Introduction To The Generic Autonomic Network Architecture: GANA

In contrast to existing approaches to autonomic networking, an appropriate model for an autonomic networked system is needed to be introduced that would allow us to reason and think of a particular autonomic function that implements the concept of a control loop at some particular level of abstraction. This means, we should then be able to talk about autonomic networking functions such as autonomic routing, autonomic forwarding, autonomic monitoring in the sense that the elements driving the control loops, use information learnt from its required information suppliers to control the behaviour of the functionality that is then considered to be autonomic.

Figure 1 shows a model of an autonomic networked system and its associated control loop that we have developed in EFIPSANS. It is derived and extended from the IBM-MAPE model [27] specifically for autonomic networking and is the basis for the derivation of the GANA architecture. The model is generic and illustrates the possibility of the distributed nature of the information suppliers that supply a so called Decision Element (DE) and the diversity of the information that can be used to manage the associated managed resources and elements. It can be applied to different levels of functionality and abstraction within node architectures and network architectures.

We define a Decision Element (DE) as a concept that is associated with some concrete resources or functional entities (e.g. protocols) that the Decision Element manages and drives its control loop. Information or views are being continuously exposed by its managed resources or functional entities, together with information coming from other required or potential information suppliers of the Decision Element. We also adopt the concept of a Managed Entity (ME) to denote a managed resource or an automated task in general, instead of a managed element. As illustrated in Fig. 1, the actions taken by the Decision Element do not all necessarily have to do with triggering some behaviour or enforcing a policy on the Managed

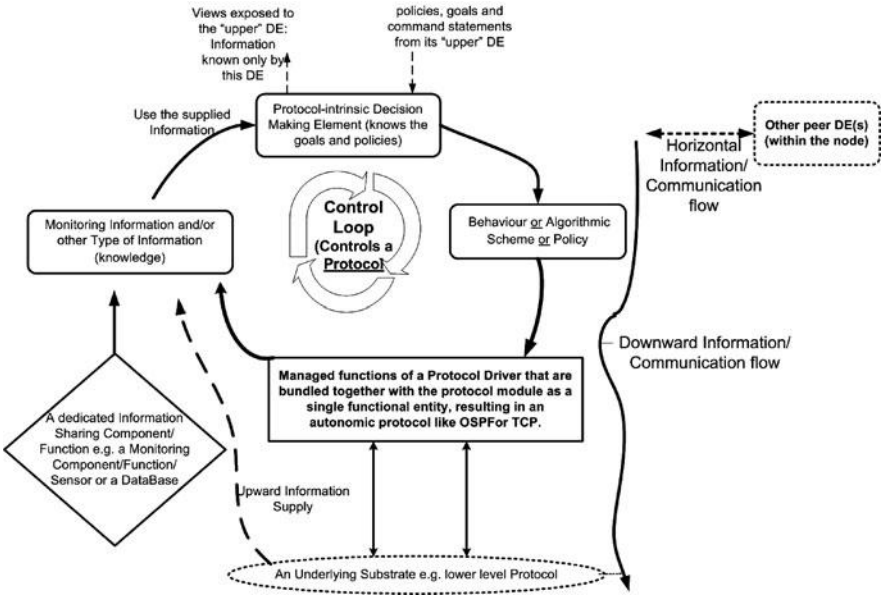


Fig. 1 A generic model for an abstract autonomic networked system

Entities but that, some of the actions executed by the Decision Element may have to do with communication between the Decision Elements and other entities. This is indicated by the extended span of the arrows: "Downward Information flow" and the "Horizontal Information flow", as well as the fact that a Decision Element also exposes views such as events to its "upper" Decision Element and receives policies, goals and command statements from its "upper" Decision Element.

3.1 Types of Control Loops in GANA

GANA defines a Hierarchical Control Loops (HCLs) framework that reflects on the need for control loops operating at different levels of abstractions and complexity. Four levels of abstractions for which Decision Elements, Managed Entities and associated control loops can be designed are described below.

Protocol Level [level 1]: The concepts of a control loop, Decision Element, Managed Entity, as well as the related self-manageability issues may be associated with some implementation of a single network protocol or an automated task (in general). For example, protocols like OSPF and TCP are known to have the above concepts intrinsically implemented within the behaviour of the protocol. This level is the lowest level of abstraction of functionality that can be associated with the manifestation of control loops.

Function Level [level 2]: On a higher level of abstraction than low level protocols and protocol stacks, concepts of a control loop, Decision Element, Managed Entities

may be addressed on the level of abstracted network functions, such as routing, forwarding, mobility management, QoS management, etc. At such level of abstraction, what is collectively managed by an assigned Decision Element are a group of protocols and mechanisms (its Managed Entities) that are collectively abstracted by a particular networking function.

Node Level [level 3]: On a higher level of autonomic networking functionality, the concepts of a control loop, Decision Elements, Managed Entities, as well as the related self-manageability issues may be associated with a system or node as a whole. The node's main Decision Element has access to the "views" exposed by the lower level Decision Elements and uses its knowledge of the higher level to influence the lower level Decision Elements to take certain desired decisions, which may in turn further influence or enforce desired behaviours on their associated Managed Entities, down to the lowest level of individual protocols.

Network Level [level 4]: The highest level of control loops is the network level. There may exist a logically centralized Decision Element (DE) or DEs such as the ones proposed in the 4D network architecture [8] that knows the objectives, goals or policies to be enforced by the whole network. The objectives, goals or policies may actually require that the nodes' main Decision Elements of the network export "views" such as events and state information to the logically centralized Decision Element(s), in order for the network-level DEs to influence or enforce the Decision Elements of the nodes to take certain desired decisions. A distributed network level control loop is implemented following the above set-up, while another would involve the main Decision Elements of nodes working co-operatively to self-organize and manage the network without the presence of a logically centralized Decision Element(s).

The hierarchies of Decision Elements in GANA, which allow for some decisions to be taken autonomously at different levels of control and complexity, can have the following forms: Hierarchical relationships between Decision Elements where lower level Decision Elements are managed by their upper level Decision Element, Peering relationships between Decision Elements, which facilitate communication between Decision Elements for exchanging information and Sibling relationships between Decision Elements whereby the entities are created or managed by the same upper level Decision Element.

3.2 The Functional Planes of GANA

In GANA [7], we first take the position that the functional planes known in today's world of networking can be compressed – with merging and re-factoring some of the planes – into four functional planes that are still called: the Decision Plane, the Dissemination Plane, the Discovery plane and the Data plane, which we adopt from the 4D architecture [8], but we re-define them for GANA because we introduce the concept of an autonomic Decision Element into the architecture of every network node/device, as opposed to the 4D, which assumes that network nodes/devices need not have Decision Elements within them.

It is important to note that, according to GANA, even if a protocol or mechanism is considered applicable to more than one plane, it still should be considered as being assigned to a single Decision Element that manages it, with other entities possibly only using its services and not managing it directly. Any requests to change or influence its behaviour from other entities other than its associated Decision Element should be achieved via negotiations with its Decision Element, which may then allow another entity to directly access the considered Managed Entity.

4 Context Awareness In EFIPSANS

GANA is supporting context-awareness through combining information models with a set of ontologies. Since information and data models are not capable of representing the detailed semantics required to reason about behaviour, we augment our use of knowledge extracted from information and data models with ontologies. Information about the current state is collected according to the information model and – through the use of a domain-specific ontology – relationships are added between context data and the behaviours of the autonomic nodes. Consequently, the nodes are able to reason about the modelled information and interact in a way that accurately reflects the overall network behaviour.

As mentioned in the previous section, the approach that we have followed in EFIPSANS is to design our information model and ontology, taking in account the existing models and ontologies on the field and re-using or extending parts of them that are related with the GANA architecture. Thus, we use parts of the DEN-ng policy model [13], the corresponding DENON-ng ontology [13] and the MOMENT [24] ontology. The DEN-ng policy model is currently adapted, embedding entities and characteristics described in the GANA meta-model [6] that is the information model designed in EFIPSANS. The GANA meta-model, defines via a semantically precise meta-model the relationships among the identified elements. It deals with and formalises each and every aspect of GANA up to details that are needed in order to be able to analyse, verify, build and evaluate autonomic behaviours. In addition to the GANA meta-model, the GANA ontology is designed in order to enrich this model with semantics, enable information gathered from the network to be analyzed and ensure that the model accurately reflect the current operational status. The ontology is described using the Ontology Web Language (OWL) as a representation language. Part of the ontology is built upon the MOMENT one, which provides a classification of the network monitoring tools, the measurements that can be performed and the different kind of networks. We extend the MOMENT ontology by adding descriptions for the supported services in an autonomic network, the characteristics and the available interfaces of the GANA defined elements and the possible interactions among them.

The GANA ontology is designed according to the requirements that were described in Section 2.2. It is explicit since all the entities are defined using separate classes and all the relationships are defined through object, datatype and annotation properties. It is formal since it follows formal relationships defined in OWL and all

the descriptions and restrictions that are stated in the GANA meta-model and open since all users can use the same set of entities and relationships in order to increase context-awareness in their scenarios. Furthermore, it is generic enough in order to be applicable to diverse kind of applications and be importable to other ontologies, easily extensible to support new artefacts that can be defined as GANA architecture is evolving and supports policy concepts that enable self-functionalities in an autonomic network.

Context awareness Managed Entities are defined within GANA, that implement mechanisms such as the proper mapping of the acquired data to the information model and the ontology based learning and reasoning. These entities are able to extract semantically rich information and make it available to other function level Decision Elements. In network level, these Managed Entities are cooperating and exchanging context-aware information, facilitating thus the extraction of meaningful events in the autonomic network.

4.1 The GANA Ontology

The GANA Ontology adds semantics to the autonomic elements that are defined in the GANA architecture. It describes the autonomic entities and the relationships among them, through the definition of a number of basic classes and properties. It is described using OWL [28], deployed on Protégé 3.4.1 and graphs are produced using the Ontoviz and Jambalaya modules.

In Fig. 2, are shown the basic classes (and part of their sub-classes) of the GANA ontology that describe general concepts that are present in an autonomic network. We present the relationships among the different classes, the restrictions that are imposed and their interactions with other classes and sub-classes. The following basic classes are defined (in alphabetical order): *AutonomicBehaviour*, *AutonomicNode*, *Capabilities*, *CommunicationInterface*, *ControlLoop*, *Element*, *Event*, *GanaPlane*, *Goal*, *Mechanism*, *MonitoringData*, *Policy*, *Profile* and *Protocol*. All these classes are subclasses of the *Owl:Thing* (abstract class) that is the root class and is automatically created in Protégé 3.4.1.

The *AutonomicNode* is the basic class that defines the concept of an autonomic node, its interfaces and its characteristics. An autonomic node in GANA consists of *Elements* that can be *DecisionElements*, *ManagedEntities* or *InformationSuppliers* (see also Fig. 1). An autonomic node has *Capabilities* (special interfaces or resources with specific scope and lifetime) and is administered by the *MainNodeDE*. The Main Node Decision Element is responsible for all the functionalities of the autonomic node and is unique for each node. Several types of Decision Elements are defined, each one of them being responsible for a specific functionality, protocol or mechanism. The most important of them are the *FaultManagementDE*, *ForwardingManagementDE*, *MobilityManagementDE*, *MonitoringDE*, *QoSManagementDE*, *RoutingManagementDE* and *SecurityManagementDE*.

These Decision Elements are managing Managed Entities through the establishment of control loops in Network, Function, Node or Protocol level. The

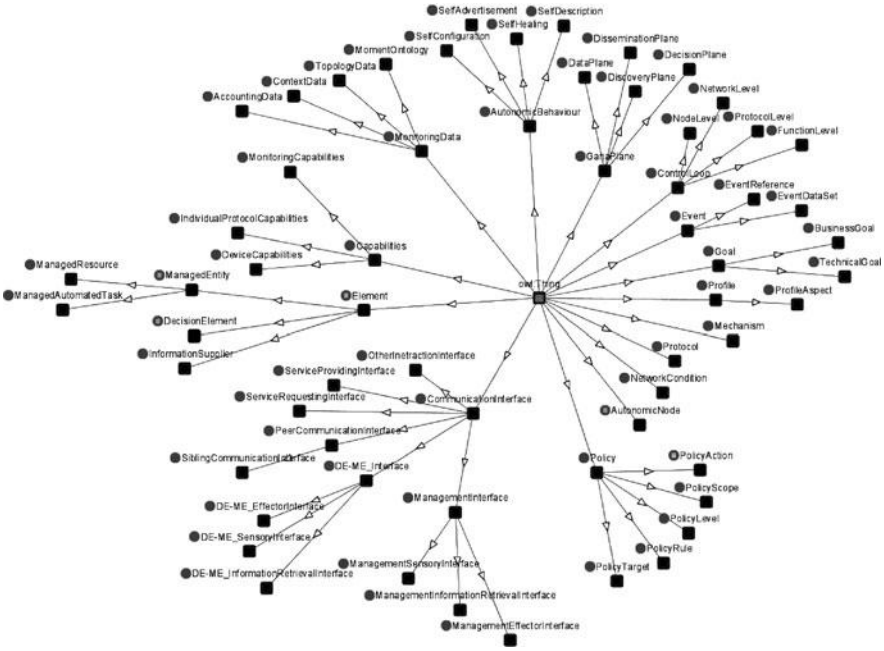


Fig. 2 Classes and sub-classes of the GANA Ontology

classes *ControlLoop* and *GanaPlane* are used for representing these concepts. It is important to note that in our ontology, we have defined a datatype property called *DecisionElementPriority* that is used for providing priorities to the different types of Decision Elements that may act upon the same Managed Entities or have conflicting policies. A Decision Element is responsible to trigger a specific *AutonomicBehaviour* or/and follow a specific *Policy*, according to information that receives in the established control loop from its Managed Entities or Information Suppliers. Each *Element* belongs to a GANA plane and specifically each Decision Element belongs to the Decision Plane while each Managed Entity belongs to the Discovery, Dissemination or Data Plane. Furthermore, the Managed Entities are responsible for the implementation of networking *Protocols* or *Mechanisms* in the autonomic network. These relationships are shown in Fig. 3.

Communication is realised among elements through the specification of communication interfaces in the class *CommunicationInterface*. Since GANA has specified Hierarchical, Peering, and Sibling Relations among elements, the same communication interfaces are created and expanded as shown in Fig. 4. *DE_ME interfaces* (sensory, effector and information retrieval) are used from the Decision Element in order to communicate with the *ManagementInterfaces* of its Managed Entities. The Decision Elements have also their own *ManagementInterfaces* for establishing communication in case they are also managed from upper level Decision Elements (e.g. from the Main Node DE). *ServiceProviding* and *ServiceRequesting*

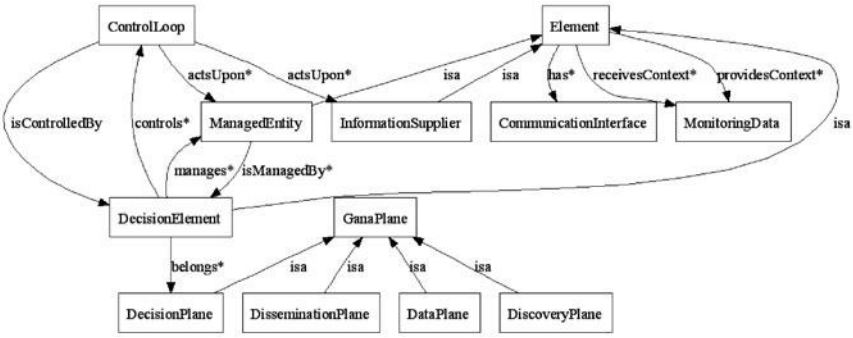


Fig. 3 Relationships among *ControlLoop*, *Element* and *GanaPlane* classes

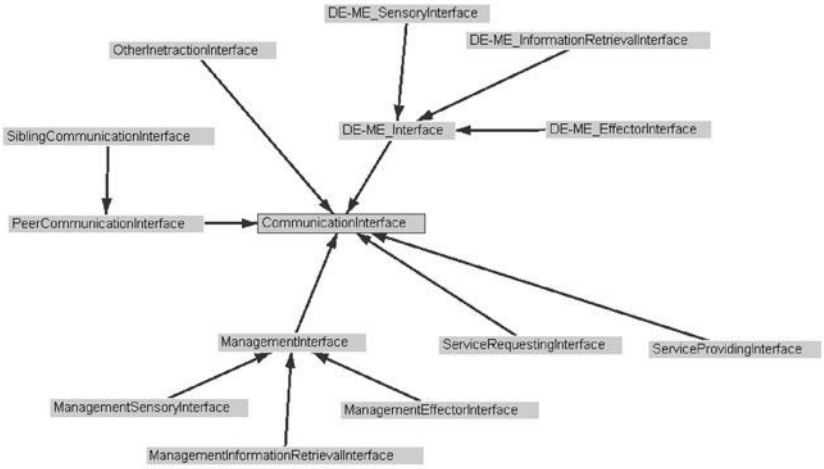


Fig. 4 Communication interface

interfaces are used from a Managed Entity while providing or requesting a service. *Peer* interfaces are used for communication among peering elements and *OtherInteraction* interfaces are used for communication with Information Suppliers.

The *MonitoringData* class describes the entities that are important for the collection of context-aware data in an autonomic network and the concepts that have to be represented for describing monitoring functionalities. In this class we use concepts defined in the MOMENT ontology and in addition we define concepts related with monitoring in autonomic networks. The MOMENT ontology specifies in detail, among others, abstract concepts (such as errors, events), active and passive measurements metrics and tools (such as delay, active bandwidth, iperf), physical

objects (such as filters, flows, network objects), network services and general concepts (such as physical location, logical location). In addition to the entities used from MOMENT, we incorporate entities from the ONIX – Overlay Network for Information eXchange – framework [6] that is the proposed solution for scalable and fault-tolerant resource discovery in large-scale IPv6 network in EFIPSANS. In ONIX system resources descriptors will be structured using XML and then will be mapped to the ontology as context-aware data. Furthermore, we are going to specify in detail entities that are responsible for providing context-aware topology and accounting data. Since the MOMENT ontology and the ONIX framework are under specification, the *MonitoringData* class will be updated accordingly in the future. *ContextData* are used for the description of *Events*, providing the capability to represent specific events in the network, associate them with an *EventReference* and through reasoning mechanisms recognize them and proceed to appropriate actions.

The classes *Policy*, *Profile* and *Goal* are used for the descriptions of policies in an autonomic network and their relationships with the available profiles that can be adopted and the goals that have to be succeeded (see Fig. 5). As stated earlier, for the description of the policies, concepts are used from the DENON-ng ontology [13]. The *Policy* class has the following sub-classes: *PolicyAction* for the description of specific actions, *PolicyLevel* for the definition of the level (e.g. network, node) where the policy can be applicable, *PolicyRule* for the description of rules and *PolicyScope* for the specification of special scopes (e.g. security). The class *Profile* defines the user and network profiles that can be adopted. Each profile is related to one or more policies and is described through combination of context data. The class *Goal* is divided in *BusinessGoals* and *TechnicalGoals*. A *Technical Goal* is followed by a policy and a *BusinessGoal* is composed of *Technical Goals*.

Thus, we have described the basic concepts that are present in the GANA architecture and their relationships. Since the ontology is designed in order to be as much generic as possible and be importable and extendable, each class can be divided in further sub-classes in accordance with the specific requirements and restrictions that each networking environment can impose.

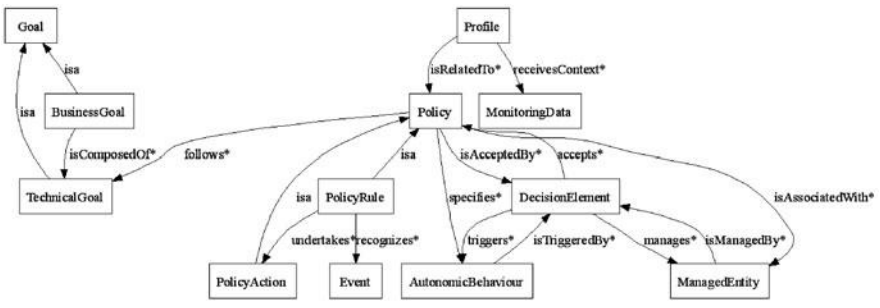


Fig. 5 Relationships among *Policy*, *Profile* and *Goal* classes

5 Conclusions – Future Work

In this chapter we presented an approach for providing context awareness in an autonomic network, using the Generic Autonomic Network Architecture (GANA) as the basis for producing autonomic behaviour specifications for selected diverse networking environments. We proposed an ontology for representing the contextual information and take advantage of the formal language description of context for increasing the extracted knowledge in autonomic heterogeneous environments.

Current challenges include the automatic validation and correctness of the deployed models and ontologies through learning and reasoning mechanisms, the provision of mechanisms for unified information modelling and storage as a support to context building and the examination of the impact that may be imposed by the processing of data.

As GANA is going to evolve, we plan to further extend the ontology by incorporating any new concepts and validate the proper design of the ontology through its application to several scenarios. We will focus on the advantages that the proper processing of context aware information can offer. Finally, a scheme will be proposed for efficient dissemination and discovery of data and cooperation among the context aware Managed Entities in wired and wireless environments.

Acknowledgements This publication is based on work carried out and funded under the framework of the European Commission ICT/FP7 project EFIPSANS (www.efipsans.org). The authors wish to thank all the collaborators in EFIPSANS for their efforts and support for completing this work.

References

1. ITU-T. TMN Management Functions, Telecommunication Standardization sector of ITU, [Online], Available: http://www.itu.int/rec/T-REC-M.3400-200002-I/en, M.3400, 2000, last visited August, 2009
2. F. Liberal, J. Fajardo, and H. Koumaras. QoE and ∗-awareness in the Future Internet, Towards the future internet – a European research perspective, IOS Press Books Online, ISBN 978-1-60750-007-0, DOI: 10.3233/978-1-60750-007-0-293, 2009
3. A. Galis et al. Management architecture and systems for future internet networks, Towards the future internet – A European Research Perspective, IOS Press Books Online, ISBN 978-1-60750-007-0, DOI: 10.3233/978-1-60750-007-0-112, 2009
4. N. Baker et al. Context-aware systems and implications for future internet, Towards the future internet – A European Research Perspective, IOS Press Books Online, ISBN 978-1-60750-007-0, DOI: 10.3233/978-1-60750-007-0-335, 2009
5. J.M. Serrano, J. Serrat, J. Strassner. Ontology-based reasoning for supporting context-aware services on autonomic networks. IEEE International Conference on Communications 2007, 2097–2102, 24–28 June 2007, DOI= http://dx.doi.org/10.1109/ICC.2007.347
6. The EFIPSANS project, [Online], Available: http://www.efipsans.org/, last visited September, 2010
7. R. Chaparadza. Requirements for a generic autonomic network architecture (GANA), suitable for standardizable autonomic behavior specifications of decision-making-elements for diverse

networking environments. International Engineering Consortium (IEC) Annual Review in Communications, 61, December 2008
8. A. Greenberg, G. Hjalmtysson, D.A. Maltz, A. Myers, J. Rexford, G. Xie, H. Yan, J. Zhan, and H. Zhang. A clean slate 4D approach to network control and management, ACM SIGCOMM Computer Communication Review, 35(5), 41–54, 2005
9. H. Ballani and P. Francis. CONman: A step towards network manageability, ACM SIGCOMM Computer Communication Review, 37(4), 205–216, 2007
10. D.D. Clark, C. Partridge, J.C. Ramming, and J.T. Worclawski. A knowledge plane for the internet, In Proceedings of the 2003 Conference on Applications, Technologies, Architectures and Protocols for Computer Communications, pp. 3–10, 2003
11. B. Jennings, S. van der Mer, S. Balasubramanium, D. Botvich, M.O. Foghlú, and W. Donnelly. Towards autonomic management of communications networks, IEEE Communications Magazine, 45(10), 112–121, 2007
12. T. Bullot, et al. A situatedness-based knowledge plane for autonomic networking, International Journal of Network Management, 18(2), 171–193, 2008
13. J.M. Serrano et al. Policy-based context integration & ontologies in autonomic applications to facilitate the information interoperability in NGN, In hot topics in autonomic computing on hot topics in autonomic computing (Jacksonville, FL). USENIX Association, Berkeley, CA, 6–6, 2007
14. C.B. Anagnostopoulos, A. Tsounis, and S. Hadjiefthymiades. Context awareness in mobile computing environments. Wireless Personal Communications, 42(3), 445–464, 2007
15. A. Zimmermann et al. Personalization and context-management. User modeling and user adaptive interaction (UMUAI), Special Issue on User Modeling in Ubiquitous Computing, (15), 275–302, 2005
16. T. Zseby et al. Towards a future internet: Node collaboration for autonomic communication, Towards the Future Internet – A European Research Perspective, IOS Press Books Online, ISBN 978-1-60750-007-0, DOI: 10.3233/978-1-60750-007-0-123, 2009
17. Y.T. Lee. Information modeling – From design to implementation, Proceedings of the 2nd World Manufacturing Congress, Universities of Durham, Durham, U.K., pp 315–321, ICSC, Canada/Switzerland, September 27–30, 1999
18. J.M. Serrano et al. Ontology-based management for context integration in pervasive services operations, In Proceedings of the 1st international Conference on Autonomous infrastructure, Management and Security: inter-Domain Management (Oslo, Norway, June 21–22, 2007). A.K. Bandara and M. Burgess. (eds) Lecture Notes In Computer Science, vol. 4543. Springer, Berlin, Heidelberg, pp. 35–48
19. DMTF, Common Information Model (CIM) Standards home page, [Online], Available: http://www.dmtf.org/standards/cim/, last visited September 2010
20. TMF, Shared Information/Data (SID) Model – Business View Concepts, Principles, and Domains. GB922, Ed. NGOSS R6.1, Document Version 6.1, November, 2005
21. R.C.A. da Rocha and M. Endler. Evolutionary and efficient context management in heterogeneous environments, In MPAC '05: Proceedings of the 3rd international workshop on Middleware for pervasive and ad-hoc computing, ,New York, NY, 2005. ACM Press, pp. 1–7
22. T. Strang, and C. Linnhoff-Popien. A context modelling survey, In Proceedings of the Workshop on Advanced Context Modelling, Reasoning and Management associated with the 6th International Conference on Ubiquitous Computing (UbiComp), Nottingham, 2004
23. R. Schmohl and U. Baumgarten. A generalized context-aware architecture in heterogeneous mobile computing environments, In Proceedings of the 4th international Conference on Wireless and Mobile Communications, July 27 – August 01, 2008, Washington, DC, 118–124. DOI= http://dx.doi.org/10.1109/ICWMC.2008.59
24. A. Salvador et al. Ontology design and implementation for IP networks monitoring, First International Workshop on Web & Semantic Technology, Wrexham, 2009

25. The AutoI project, [Online], Available: http://ist-autoi.eu/autoi/, last visited September, 2010
26. The EMANICS project, [Online], Available: http://emanics.org/, last visited September 2010
27. IBM, Understanding the Autonomic Manager Concept, [Online], Available: http://www.ibm.com/developerworks/library/ac-amconcept/index.html, last visited September, 2010
28. OWL Web Ontology Language Reference, [Online], Available: http://www.w3.org/TR/owl-ref/, last visited September, 2010

Enabling e-Infrastructures in Italy Through the GARR Network

Mario Reale and Ugo Monaco

Abstract In this chapter we present GARR, the Italian Research and Academic Network, its current and forthcoming infrastructures, and the major projects involving it. We underline the role of GARR as an e-Infrastructure enabler, by reporting about the current and future GARR activities concerning this field.

1 Introduction

In the last few years the term *e-Infrastructure* has been progressively emerging in the scientific domain to indicate a distributed computing and storage infrastructure, and more in general, geographically distributed services, generally spanning different administrative domains and local area networks, integrated to provide affordable functionality to users in the accomplishment of various tasks. Web Services in an open Service Oriented Architecture emerged as the main corresponding reference standard protocol and architecture. Computing, data Grids and Clouds are a typical example of e-Infrastructures. The key enabling element for e-Infrastructures is the underlying network, whose performances and characteristics dramatically impact on the functionality exploited by users.

In this chapter we introduce the Italian Academic and Research Network, GARR [1], and describe its support to the provisioning of large e-Infrastructures in Italy and Worldwide.

The rest of the chapter is organized as follows. In Section 2 we introduce GARR, both as an organization and a network, describing its current and future network architectures and operating principles. In Section 3 we report about the major Grid e-Infrastructures and research projects supported by the GARR network, and, in some cases, we also describe the direct contribution of GARR in them. In Section 4 we sketch future developments of the GARR network and how the implied improvements will positively impact on e-Infrastructures. We then draw a short set of conclusions.

M. Reale (✉)
Consortium GARR – The Italian Research and Academic Network, Roma, Italy
e-mail: mario.reale@garr.it

F. Davoli et al. (eds.), *Remote Instrumentation Services on the e-Infrastructure*,
DOI 10.1007/978-1-4419-5574-6_13,

2 GARR And The Current GARR Network

Consortium GARR is a no-profit organization settled under the aegis of the Ministry of Education and Scientific Research (MIUR) and founded by CNR (National Research Council), ENEA (Organization for the New Technologies, the Energy and the Environment), Fondazione CRUI (Conference of Chancellors of the Italian Universities) and INFN (National Institute for Nuclear Physics).

The mission of the Consortium GARR, for the sake of simplicity in the following only referred to as GARR, is to fulfil the networking needs of the Italian academic and research community, i.e., the GARR user community, and to facilitate cooperation in the field of research through the exploitation of leading-edge e-Infrastructures. To this aim, GARR designs, engineers and operates the Italian high-speed data network for University, Scientific Research and Education, referred to as the GARR network. Furthermore, GARR provides advanced network and application services and support, it disseminates about network infrastructures and it stimulates the exchange of technical know-how, through the contribution and the organization of specific events involving the whole GARR community.

About 450 user sites (overall more than 2 Million end users), including Research Organizations, Universities, Observatories, Laboratories, Institute for Research in Health Care (IRCCS), Music Conservatories and Academies of Art, Libraries, Schools, Museums and other Scientific and Educational Facilities (of national and international relevance) are currently connected to the current implementation of the GARR network which is named GARR-Gigabit [2] (GARR-G).

Being the National Research and Education Network (NREN), GARR is committed to serve the community of their users and network administrators in an efficient, proactive manner. To this aim, customized network reports are produced, dedicated VPNs and links are put into operation, and specific analyses of requirements are undertaken.

Work for the GARR-G network started in 2001 with the set up of a testing facility and then in 2003 with the implementation of the production network. The design principles of GARR-G took into account technical and administrative constraints, such as costs, availability of infrastructures and features on commercial equipment, provisioning time. They were aimed at fulfilling user and network management and operation requirements: network and application service portfolio, capacity, scalability, performance, reliability, Quality of Service, efficient management/control and operation, and flexibility.

The GARR-G backbone topology is shown in Fig. 1.

GARR-G is basically an IP nationwide network based on about 100 network nodes (routers, switches, D/CWDM equipment) located in 45 Points of Presence (PoPs) distributed over the whole national territory and directly operated by local Access Port Managers (APMs) and GARR technical staff. Network backbone and access links are based on different technologies, as C/DWDM, SDH, MPLS, Ethernet, ATM, xDSL, and they are provided by GARR or by user infrastructure if available, or, in most cases, by national and international operators. User traffic aggregation and transport is mainly carried out by IP routers directly managed by

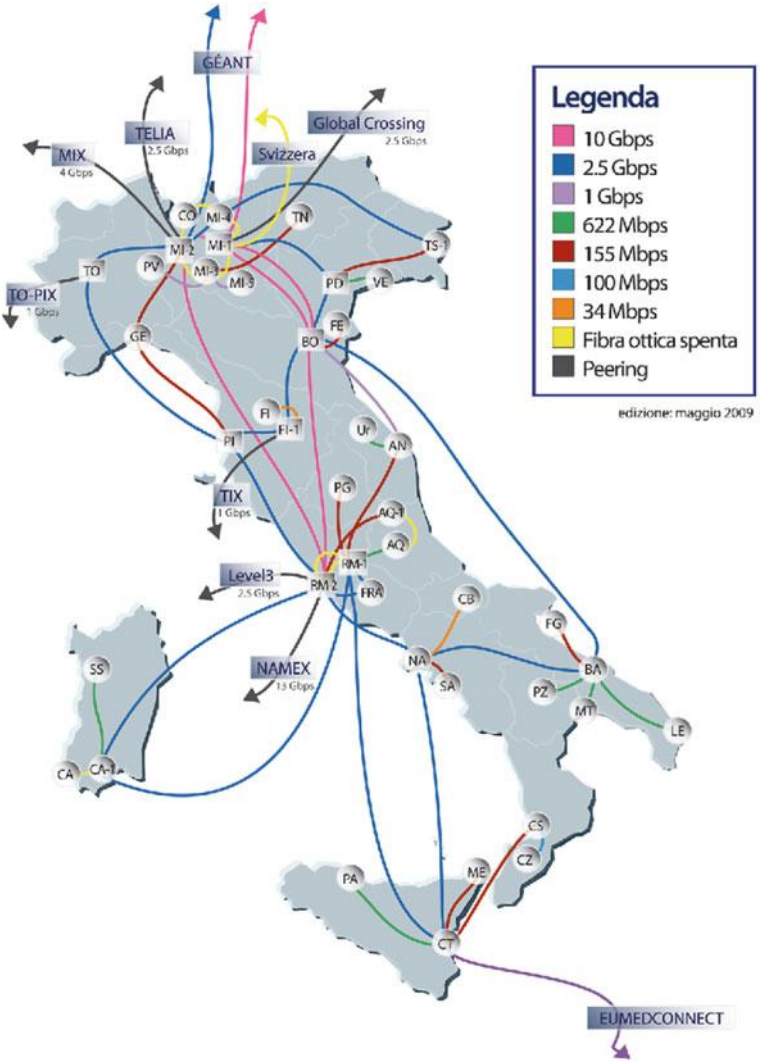


Fig. 1 GARR-G backbone network

the GARR Network Operation Center (NOC). The backbone link capacity has been designed in order to account possible network faults and it ranges from 10 Gbps to 2.5 Gbps in the core and from 1 Gbps to 34 Mbps in the edge, while access link capacity spreads from 10 Gbps to 2 Mbps. Aggregated backbone and access capacities are more than 100 Gbps and 50 Gbps respectively.

Moreover, the GARR-G network is interconnected with the rest of worldwide Internet by peering links with overall capacity of about 80 Gbps. Peering is established with General Internet upstream providers, national and international Internet Service Providers (ISP) and Content Providers (CP) and with the European and

International research community, through local peering with neighboring National Research and Education Networks (NRENs) and globally with the GÉANT [3] network, the Pan-European Research and Education Network.

The interconnection among GARR and GÉANT or other NRENs is not just limited to physical network peering, but it is strongly based also on close relationships at the technical staff level. Common involvements in service operations and research activities are permanently ongoing, as those established within the framework of international projects and initiatives for innovative applications and infrastructure, as GN2/GN3, FEDERICA [4], LHCOPN [5], DEISA [6], EVLBI [7], EGEE III [8], EGI [9] and EUMEDCONNECT II [10].

These relationships represent a great advantage for the GARR community since they are a unique and value-added opportunity to enable the provisioning of high performance and customized network services, at the European and Worldwide scope, for the support of different applications including Grids, distributed computing, telemedicine, e-learning and multimedia, high energy physics, radio astronomy, earth observation, supercomputing.

3 E-Infrastructure Supported By The GARR Network

GARR is acting both as a network provider and a supplier of research personnel and resources in relevant national and international activities around the network.

To mention some of them, GARR is providing resources and personnel to the European projects GÉANT 3 (European Research and Education backbone network), EUMEDCONNECT II (Eu-Mediterranean Research Network), EGEE III and EUMEDGRID-Support (International Grid projects), IGI [11] (The Italian National Grid Initiative), GriSu [12], INFNGRID [13], ENEAGRID [14] (national Grid projects), IDEM [15] (National Federated Authentication and Authorization Infrastructure based on Shibboleth).

3.1 EGEE III and the W-LCG

EGEE represents the major EU supported effort on international grid projects. It is currently running in its third mandate. It started in 2004 and currently provides a stable e-Infrastructure for the provisioning of e-Science to a large community of European scientists, hosting within its infrastructure the European wing of the W-LCG [16] (Worldwide-LHC Computing Grid), which is the Grid dedicated to data analysis of the LHC particle collider at CERN, Geneva, Switzerland.

The EGEE Production Service infrastructure is a large multi-science Grid infrastructure, federating around 260 resource centres world-wide, providing some 40,000 CPUs (around 80,000 cores) and several Petabytes of storage.

It spans over 45 different countries. This infrastructure is used on a daily basis by several thousands of scientists federated in over 200 Virtual Organizations (VOs) and 20 k simultaneous jobs running. It is a stable, well-supported infrastructure, running the latest released versions of the gLite middleware.

The W-LCG grid is a hierarchical world wide e-Science infrastructure, whose main centres are organized in Tiers. The Tier-0 centre is the most important of all, being the initial centre where all data produced by the LHC machine are acquired and initially stored, namely CERN in Geneva, Switzerland. Data are then distributed to 11 Tier-1 centres in many EU countries and in the USA using the LHCOPN (LHC Optical Private Network). The LHC OPN topology is shown in Fig. 2.

In Italy the Tier-1 centre is hosted at CNAF in Bologna. GARR is providing connectivity to CNAF by means of a redundant 10 Gbps link to both CERN through GÉANT and to Karlsruhe, Germany (the German Tier-1) by means of a dedicated cross-border fibre through Switzerland. Various Tier-2 centres (currently 10) are represented by relevant data and computing centres for the major High Energy Physics Experiments are connected through the GARR network both to the national Tier-1 at CNAF and to the Tier-0 at CERN.

3.1.1 Enabling IPv6 Within the gLite Middleware (EGEE SA2 IPv6 Task)

Within the Service Activity 2 – Network Support (SA2) of EGEE III, a task in which GARR is highly involved is the analysis and pursuing of the IPv6 compliance of the gLite middleware (Task TSA2.3.3, coordinated by GARR).

Among the activities carried out by this task [17] there is:

- the overall assessment of the degree of IPv6 compliance in the various gLite middleware components
- a general analysis of the IPv6 compliance of the external packages used by the gLite stack
- support to the gLite developers' community about IPv6 compliant programming and how to write Address Family independent applications. The support includes the organization of specific tutorials to the EGEE developers' and specific studies on both selected gLite and external components and the posting of well defined, specific bugs about IPv6 related to evident or presumed non-compliance of the gLite code
- the development and maintenance of automated tools aimed at pursuing the analysis of the IPv6 compliance of the gLite code (static and dynamic code checkers)

This task has provided already a relevant outcome about IPv6 for the gLite community and has kept high the interest around IPv6 within EGEE and fostered a gathering of efforts and synergies among different Grid projects in Europe and in the United States. IPv6 sessions have been organized at the 25th Open Grid Forum in March 2009, Catania, at the yearly EGEE conferences and User Forums.

3.2 Infngrid

The INFN grid is the INFN national grid infrastructure devoted to providing storage and computing resources to the High Energy Physics community. It is made up of around 35 sites spread over the whole country and providing an overall Computing Power of around several Millions SpecInt2000 and a distributed Storage capacity of several Petabytes.

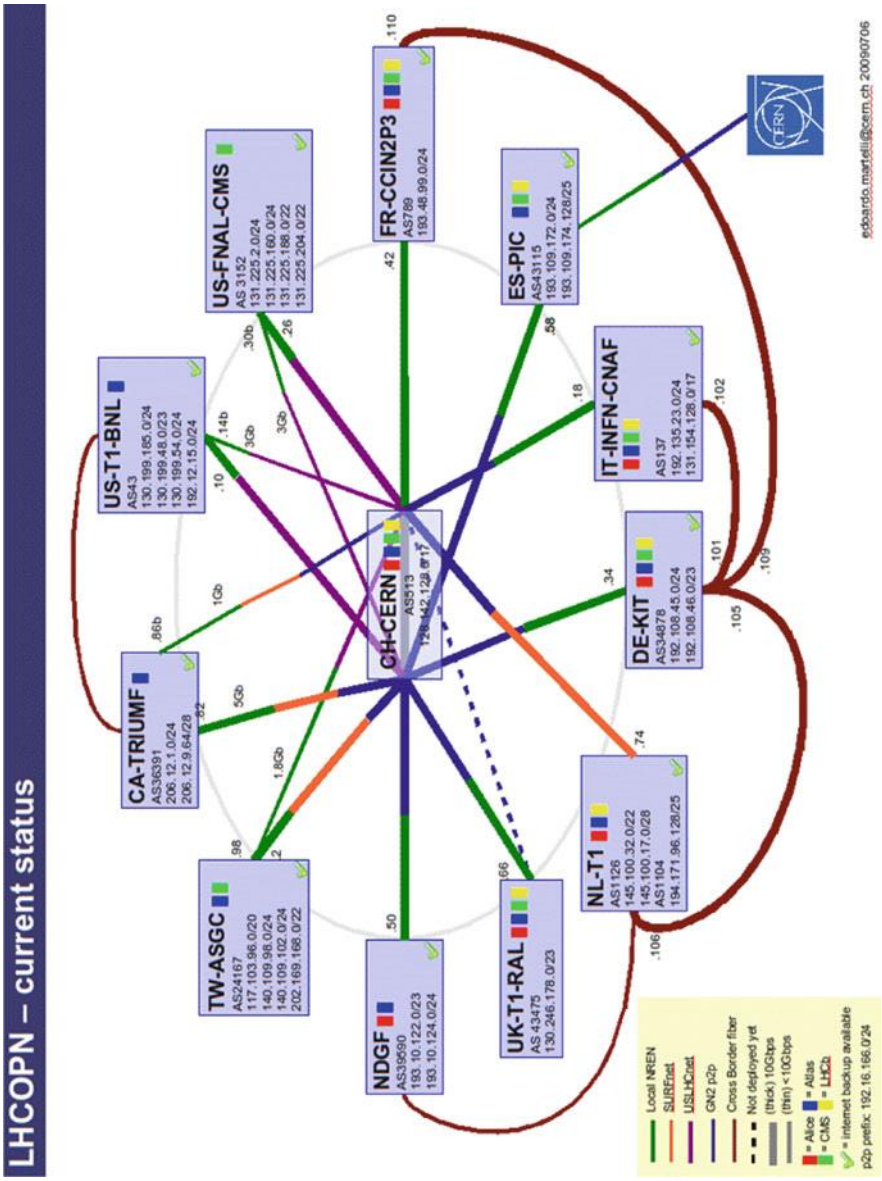


Fig. 2 LHC optical private network topology

INFNGRID acts as the Italian arm of the World-Wide LCG (W-LCG) computing Grid, which is practically a federation of national and regional grids.

The most important computing and data centre for INFNGRID is the Tier-1 centre at CNAF in Bologna. CNAF provides an overall computing capacity of 6.5 Mega SpecInt2000, corresponding to around 2600 core processors, a tape storage capacity of 4 PetaBytes, provided through 20 tape servers, and a disk storage capacity of several PetaBytes.

All these relevant resources are interconnected to INFNGRID, W-LCG and the world-wide scientific community by means of 3 10 Gbps link provided by GARR and GEANT. The Tier-1 at CNAF takes part to the LHC Optical Private Network (LHCOPN), interconnecting by means of dedicated lambdas on optical fibres the Tier-0 Centre at CERN (Geneva) and the 11 Tier 1 world-wide. In particular, CNAF is linked through a cross border fibre through Switzerland to the German Tier-1 centre in Karlsruhe.

3.3 ENEA Grid

ENEA is the National Agency for New Technologies, Energies and the Environment. It includes 12 major research sites all over Italy, and its major computing and storage centres are interconnected by the ENEA GRID infrastructure, providing ENEA researchers with easy access to the multi-platform computing environment. GRID functionalities (authentication, authorization, resource discovery and management) are provided by mature multiplatform components, the ENEA GRID middleware. In particular the infrastructure relies on:

- Open AFS/Kerberos 5 (integrated) as a Distributed File System
- LSF Multicluster as a WAN resources manager
- Java, Citrix & Freenx as a User Interface

ENEA focused on pursuing interoperability with other Grid infrastructures: a gateway implementation method (Shared Proxy Approach for Grid Objects) has been developed and deployed to achieve interoperability with gLite-based grid infrastructures.

The major resource centre is the CRESCO [18] computing centre in Portici (Naples). CRESCO is an HPC system (rank n. 125 in the Top 500 list) , capable of 17.1 Teraflops, made by 300 hosts, 2720 cores. Some other 600 CPUs represent the remaining part of the ENEA GRID infrastructure.

GARR provides the network to the ENEA GRID infrastructure, by means of 9 PoP with various access link capacities (ranging from 18 to 2000 Mbps).

3.4 GRISU': The Southern Grid Infrastructure

Five major national e-Infrastructural project hosted in Southern Italy have gathered their resources in the National Italian Southern Grid infrastructure, encompassing

relevant computing and storage resources for Grid and HPC computing. It has been established in September 2007. The five federated centres are:

1. SPACI / University of Naples (Naples) [19]
2. CRESCO (Portici, Naples)
3. PI2S2 / COMETA (3 major centres in Sicily) [20]
4. CyberSAR/ CosmoLab (Cagliari, Sardinia) [21]
5. SCOPE (Lecce) [22]

These five federated projects represent an e-Infrastructure overall providing around 8000 CPU cores and 0.5 PetaBytes of Storage, devoted to research and innovation.

The GriSu e-Infrastructure is based on the underlying GARR network (GARR-G) to interconnect its resources and to connect them to the internet and to the research networks world wide.

GARR-X [23], the forthcoming GARR network, will be able to put in place high throughput dedicated end-to-end circuits, completely devoted to the GriSu infrastructure, and to reserve individual lambdas for their traffic.

3.5 IGI: The Italian National Grid Initiative

IGI is the Italian Grid Infrastructure project, representing the Italian National Grid Initiative both at the national and international levels. It encompasses all major research institutions in Italy: it currently represents 16 institutions in different domains (INFN, ENEA, GARR, CNR, INAF, INGV, University of Napoli Federico II, University of Calabria, COMETA, CASPUR, SISSA, COSMOLAB, ELETTRA, SPACI, University of Piemonte Orientale, University of Perugia).

It is currently at its first stages of life, still waiting to be officially recognized as an independent legal entity by the Italian Government. However it started already to manage its infrastructure, having been running in the temporary of a Joint Research Unit already for almost 2 years now.

Its infrastructure includes 50 sites in Italy, among which the major resources are included in the CRESCO computing center, the CYBERSAR project, the INFNGRID project, the LIBI international laboratory for Bioinformatics, and the southern Italian projects PI2S2 and SCoPE.

The reference VOs and applications are in the fields of High Energy Physics, Astronomy, Plasma and Fusion Physics, Bioinformatics, Material Science, Remote Instrumentation and Control, Theoretical Physics, Engineering, Economics, and others.

IGI is expected to completely take off and to be fully operational in 2010 with the start of the EGI project, for which it represents Italy.

The whole IGI e-Infrastructure relies on the GARR network to implement its services, being the reference research network for all sites of IGI, which are interconnected and connected to the GARR backbone (and through it, to the world

wide system of research networks, starting from GÉANT) at different access link throughputs (typically ranging from few Mbps to 1 Gbps – with the exception of several 10 Gpbs links for CNAF).

3.6 The FEDERICA Project

FEDERICA (Federated e-infrastructure dedicated to European Researchers innovating in computing network architectures) is a FP7 European Commission co-founded project started in 2008. The project is coordinated by GARR and it has 20 partners, 9 NRENs, TERENA [24], DANTE [25], universities, research institutions and Juniper Networks. The goal of the project is the creation of a pan-European e-infrastructure for testing and research in future Internet.

The FEDERICA e-infrastructure is built over existing NREN networks and leverages GÉANT end-to-end connection service GÉANT+. It consists on 12 PoPs interconnected by 18 1 Gpbs Ethernet links. Each PoP hosts one FEDERICA L2/L3 switch (Juniper MX480 or EX3200) and several computing elements, i.e., servers (Sun X2200 M2). Dedicated, agnostic and transparent network environments named *slices* are provided simultaneously to different FEDERICA users, e.g., research groups or teams involved in other national or international projects.

Slices are created by leveraging virtualization capabilities enabled in switches and servers and they consist in a set of virtual network and computing resources configured according to user's request. Control and configuration capabilities within a slice are provided to the user down to the lowest possible network layer. Slices can be used in order to carry out "disruptive" research (generally not allowed on wide-area production networks), to test novel tools, protocols and paradigms and to enable the graceful implementation of new single/multi-domain service layers. Slices may also communicate with the General Internet. Moreover, the FEDERICA infrastructure is open to be interconnected or federated with other e-Infrastructures and also it is open to host researcher hardware and applications in compliance with the policies provided the User Policy Board.

In the framework of the FEDERICA project GARR operates and supports the local FEDERICA e-infrastructure, e.g., resources setup, installation and housing, interconnection with GÉANT PoP and IP peering with the GARR network.

4 GARR-X And Future Developments

In order to improve performances, to better support the existing and future e-Infrastructures, and to be able to meet possible future – currently unknown – user requests, GARR is working on two parallel fronts. On one side GARR is implementing a next-generation multi-service data network based on leading-edge technologies and functionalities. On the other, GARR participates and contributes in research projects and operational activities which are expected to bring:

- new experiences and initial solutions for provisioning / control / management of e-infrastructures in multi-domain, multi-vendor, and virtualized environments
- improvements in grid middleware products paradigms
- input with real test cases and results to standardization bodies

In 2007 GARR started an internal project aimed to design the evolution of the GARR network for the next 5 years. The new network infrastructure which is named GARR-X, will overcome some limitations of GARR-G and will be able to provide advanced services and bandwidth regardless of the user geographic location, while reducing access costs, in particular, GARR-X will result in:

- more capacity, expected 40 times at the end of the project with respect to the current one and double capacity from the first year
- better performance, e.g., low delay and jitter
- low layer network services, e.g., optical VPNs, Carrier Ethernet services and SAN extensions
- on-demand lambdas and end-to-end dedicated L1/L2 circuits
- more flexibility
- faster provisioning times

Enabling technologies and factors are:

- dark fibre hiring and direct management of the transport infrastructure (in GARR-G mainly commissioned to leased-lines operators)
- DWDM long haul transport equipment
- efficient and reliable user traffic aggregation and transport
- integration with Metropolitan and Regional Area Networks in order to optimize the use of the existing resources, to contain costs and to identify likely synergies among different users

The GARR-X dark fibre topology is shown in Fig. 3. More than 10,000 km of backbone and access dark fibre and about 250 new carrier grade pieces of equipment (routers, switches, transport and amplification nodes) will be tendered.

Moreover, during the project time, peering with the rest of the General Internet will be improved, either by capacity enhancements of existing links, or by establishing new peering with neighbouring NREN networks. These new connections will mainly be based on cross border fibres (CBFs) and will boost the role of Italy towards the Mediterranean area. The GARR-X project has been organized in subsequent phases; the first one, named Phase0, has already started and it is expected to light up by the beginning of 2010 more than 4,000 km of the backbone and access dark fibres foreseen in the whole project. The GARR-X Phase0 backbone dark fibre topology is reported in Fig. 4.

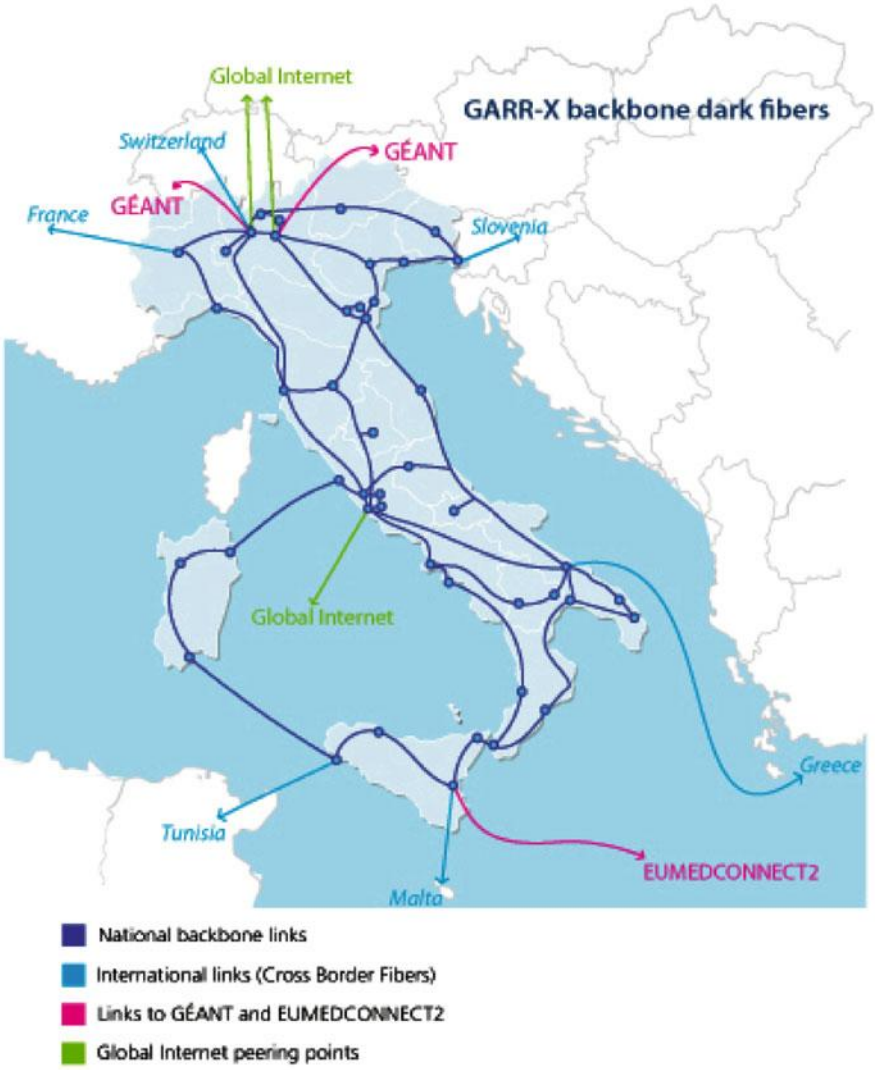


Fig. 3 GARR-X backbone dark fibres

5 Conclusions

GARR, the Italian Research and Academic Network is largely involved in the provisioning of e-Infrastructures in Italy and, through the interconnection with GÉANT, it also supports Worldwide scale ones.

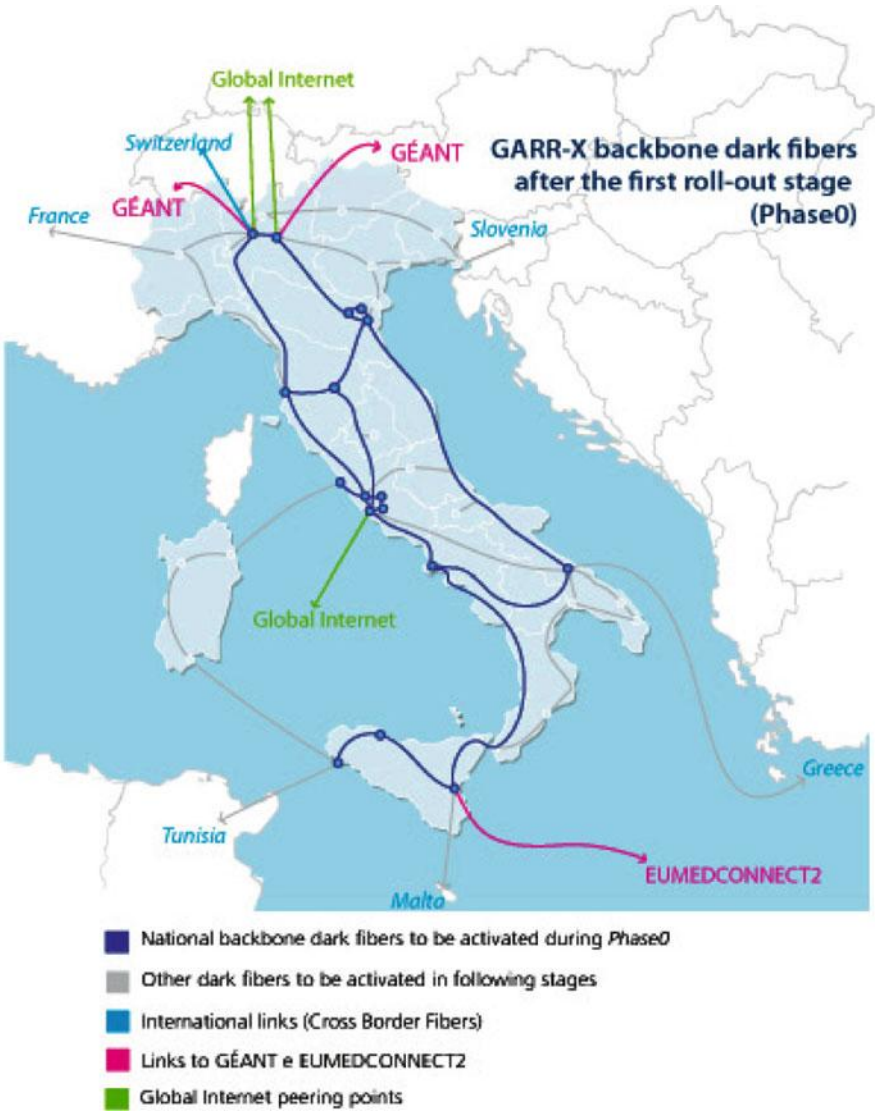


Fig. 4 GARR-X Phase0 backbone dark fibres

GÉANT, the National R&E Networks and the Campus Networks provide a federal model for the connectivity of European Researchers and Scientists.

The network layers play a crucial role in determining the final performances experienced by large community of users.

Grids and e-Infrastructures in general represent major demanding users for the network and will be exploiting all their potential with respect to capacity and performances.

GARR-X is the Italian contribution to the development of research networks in the next years and is the answer to the increasing demand for highly performing e-Infrastructures in future.

In future it will further extend the possibility of meeting specific users' needs and requirements, by – for instance – allowing setting up dedicated, reserved bandwidths by means of the optical networking devices and reducing the circuit provisioning time for newly connected sites.

It will make end-to-end fully customized services reality for *every* network user, by also increasing the overall performances and reducing links and circuits provisioning time.

The research and infrastructure projects mentioned in this chapter are expected to provide relevant outcomes, know-how and facilities. They represent the challenge for fostering the introduction of highly performing networks and communication protocols for tomorrow's scientific Europe.

Acknowledgments The authors of this chapter would like to thank their colleagues for having provided suggestions and material included here. In particolar, we thank Marco Marletta, Mauro Campanella, Alessandro Inzerilli, Federica Tanlongo, Claudia Battista and Massimo Carboni.

References

1. GARR – The Italian Academic and Research Network – http://www.garr.it
2. GARR-G – GARR-Gigabit – http://www.garr.it/stampaGARR/materiali/leaflet_RETE_GARR.pdf
3. GÉANT – The Pan-European Research and Education Network – www.geant.net
4. FEDERICA – Federated E-infrastructure Dedicated to European Researchers Innovating in Computing network Architectures – http://www.fp7-federica.eu
5. LHCOPN – Large Hadron Collider Optical Private Network – https://twiki.cern.ch/twiki/bin/view/LHCOPN
6. DEISA – Distributed European Infrastructure for Supercomputing Applications – http://www.deisa.eu
7. EVLBI – European Very Long Baseline Interferometry – http://www.evlbi.org
8. EGEE III – Enabling Grids for E-sciencE III – www.eu-egee.org
9. EGI – European Grid Initiative – www.eu-egi.org
10. EUMEDCONNECT II http://www.eumedconnect2.net/
11. IGI – Italian Grid Infrastructure – http://igi.cnaf.infn.it
12. GriSù – riglia del Sud – http://www.grisu-org.it
13. INFNGRID https://grid.infn.it
14. ENEA-GRID www.eneagrid.enea.it
15. IDEM – IDEntity Management per l′accesso federato – http://www.idem.garr.it
16. W-LCG – Worldwide LHC Computing Grid – http://lcg.web.cern.ch/LCG
17. EGEE III SA2 IPv6 Follow Up Wiki pages https://twiki.cern.ch/twiki/bin/view/EGEE/IPv6FollowUp
18. CRESCO – Centro computazionale di RicErca sui Sistemi COmplessi ENEA GRID – http://www.cresco.enea.it
19. SPACI – Southern Partnership for Advanced Computational Infrastructures – www.spaci.it
20. COMETA – COnsorzio Multi Ente per la promozione e l′adozione di Tecnologie di calcolo Avanzato – http://www.consorzio-cometa.it
21. CyberSAR – Cyberinfrastructure per la ricerca scientifica e tecnologica in Sardegna – http://www.cybersar.com

22. SCOPE – Performance, Cooperative and distributed System for scientific Elaboration – http://www.scope.unina.it
23. GARR-X http://www.garr.it/reteGARR/GARR-X.php
24. TERENA – Trans-European Research and Education Networking Association – http://www.terena.org
25. DANTE – Delivery of Advanced Network Technology to Europe – http://www.dante.net

Academic MANs and PIONIER – Polish Road to e-Infrastructure for e-Science

Artur Binczewski, Stanisław Starzak, and Maciej Stroiński

Abstract The chapter present the history of e-infrastructure for researchers in Poland. It includes the history of Metropolitan Area Networks and of the National Research and Education Network (NREN) development and examples of use of the NREN in different research disciplines. It describes the integrated service platforms concept and cases of implementation of five services in the Polish NREN.

1 Introduction

Building a dedicated information & communication infrastructure for science in Poland boasts a long, 17-year history. Its beginning was associated with the necessity of building metropolitan academic optical networks that would connect and integrate common services for the universities and scientific institutions acting in a given city. One of the most important common services was the environmental access to High Performance Computing as that time brought the foundation of five such HPC Centers. An additional motivation to build metropolitan networks was the fact that the units were highly dispersed as well as the lack of proper telecommunication infrastructure of the main national operator. It was assumed that the networks must outdistance the solutions available on the market since the scientific community expects to have access to such infrastructure in order to create innovations. Consequently, the scientific community was forced to make use of such innovative technologies as FDDI in the environment of metropolitan networks. Innovative was also the acceptance of the concept of a metropolitan network as "all – optical". This attitude was then adapted and developed while creating the concept of the national PIONIER Network. In this case an innovative counterpart was the Ethernet over DWDM technology in the wide area network. The implementation

A. Binczewski (✉)
Poznan Supercomputing and Networking Center, Poznan, Poland
e-mail: artur@man.poznan.pl

F. Davoli et al. (eds.), *Remote Instrumentation Services on the e-Infrastructure*,
DOI 10.1007/978-1-4419-5574-6_14,

of innovative technologies came along with a significant economic effect of the adapted solutions. For example, the implementation of the PIONIER Network in 2003 resulted in the 20-fold increase of bandwidth in the backbone of the national network, and its maintenance cost was 50% lower. It was the consequence of switching from the channels leased from the telecommunication operators to our own infrastructure.

The chapter depicts the Polish path to building an advanced infrastructure for e-Science, describes academic MANs, the PIONIER Network, and its significance for science. The chapter presents the commenced development project of the PIONIER Network in the form of an integrated service platform for science named PLATON. Advanced services for the whole scientific community, including HD videoconferencing, EDUROAM, campus computations, archiving, and the scientific interactive HD television, are being realized within the project.

2 MANs and Common Network Initiatives

The process of building the e-Infrastructure for e-Science in Poland commenced in 1993 when 21 academic cities saw the construction of scientific, fiber-optic Metropolitan Area Networks (MANs). At the same time, five High Performance Computing Centers were founded. The networks were connected via channels up to 2 Mbps. In 1997, several academic MANs formed a consortium to build POL-34, a national broadband academic network based on digital channels leased from telecommunication operators. Built in the ATM 34 Mbps technology and then upgraded to 155 Mbps and 622 Mbps, the network operated till 2003. The same year witnessed the launching of the national optical network called PIONIER – Polish Optical Internet, built since 2001. The network uses the concept of its own optical fiber, the DWDM technology, and 10 Gbps Ethernet. In 2006, the scientific community outlined the strategy of the PIONIER development named PIONIER2; the currently realized project of constructing a service platform for science called PLATON is carried out within the PIONIER2.

The presented development path of the Polish e-Infrastructure does differ from the solutions in other European countries. Unlike most European NRENs, the Polish NREN does not provide direct support for scientific institutions (end users) and constitutes a backbone network between/connecting 21 MANs and 5 HPC Centers.

From the organizational point of view, academic MANs are affiliated to selected universities or other scientific institutions in a given region and substantially supervised by the elected Boards of Users. A MAN is obliged to render services for the local scientific community and is entitled to apply for subsidy from the Ministry of Science. All MANs are also legal operators of data transmission and Internet access services as well as members of the PIONIER Consortium. MAN's activity is "non-profit", and the costs are covered from the subsidy from the Ministry of Science and financial support from the end users.

The MAN infrastructure is based on its own optical network connecting buildings and campuses of the local scientific community. The teletransmission system is built directly on "dark fibers"; now it is mostly 1 Gbps or 10 Gbps Ethernet offering transport services in VLAN. Access to "dark fibers" made it possible to carry out tests and implement new transmission technologies without the need to discontinue rendering services to users. Over the last 15 years, this policy facilitated the implementation and migration to the following new technologies: originally FDDI, then ATM and 1 Gbps Ethernet, and now 10 Gbps Ethernet/MPLS. The MAN infrastructure usually covers all urban areas and enables cooperation not only with scientific institutions but also the municipal authorities, schools, hospitals and public institutions (libraries, museums, etc.). MANs offer such basic services as the scalable Internet access service, virtual networks services, and resource access services (e-mail accounts, FTP services, www, etc.).

The PIONIER Network is operated and owned by the Poznań Supercomputing and Networking Center affiliated to the Polish Academy of Sciences. It connects over 700 universities and R&D institutions via MANs and renders services to more than 2 million scientists and students.

3 The PIONIER Network

The concept of the program named PIONIER – "Polish Optical Internet – Advanced Applications, Services and Technologies for the Information Society" was developed in the end of 1999, and a year later, it was accepted by the State Committee for Scientific Research as an official program of developing the e-Infrastructure for e-Science. The program envisaged building a national optical network as a base for the development of services and applications for science. A model of the network was constructed in 2000, and presented at the exhibition accompanying the ISThmus2000 Conference. The experiments were described in [1, 2], and the construction works commenced in 2001. The first stage included building 2300 km of the network's own optical links and finished in 2003 when the network was launched in the DWDM and 10 Gbps Ethernet technologies.

The successive years saw the expansion of the network which now covers 5400 km (Fig. 1). It connects all MANs, and by the end of this year all units will have been connected in the closed loops topology. About 400 km of optical network are currently being constructed.

The network is being built in cooperation with the telecommunication operators selected to be the partners of the scientific community in the public tender. The cooperation basically comes down to two models: the first one is the shared cable and separate fibers, and the second is common telecommunication ducts and separate cables and pipes. Such model of building the scientific network has significantly lowered the construction costs.

A very important element of the concept of building the PIONIER Network was the realization of optical links to the Polish borders in order to connect

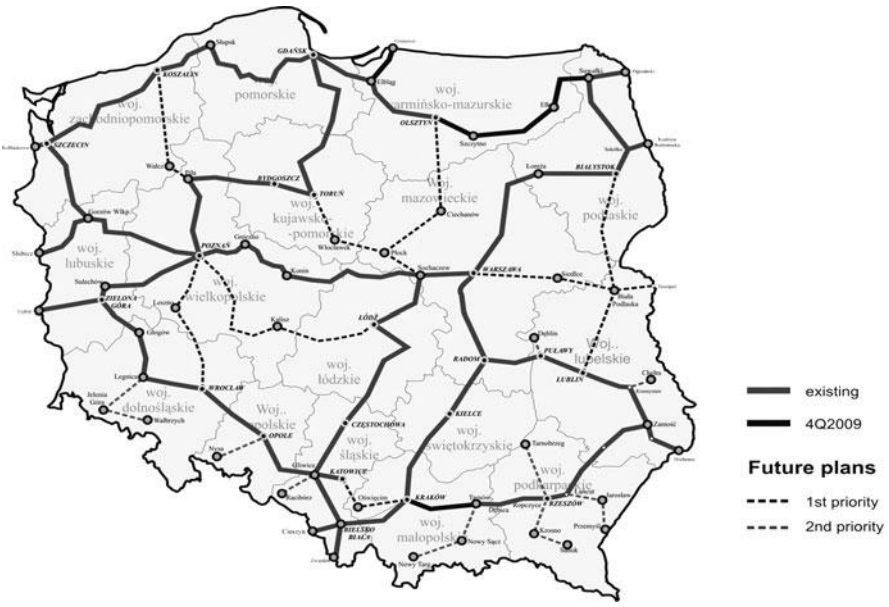


Fig. 1 Topology of fiber infrastructure in PIONIER network

to the neighboring NRENs. The idea was presented in [2, 3]. Presently, the PIONIER network is directly connected to Germany (3 optical links), the Czech Republic, Slovakia, Ukraine, Belarus, and Lithuania. The connection to the Russian Kaliningrad Region is being constructed.

In order to establish cooperation with all the countries of the Baltic region, PIONIER has a direct optical link to Hamburg, where it connects to the networks of the Scandinavian states acting within the NORDUnet Organization as well as the Dutch network named SURFnet. The network of cross-border connections is depicted in Fig. 2.

The PIONIER Network has been designed in the DWDM and 10 Gbps Ethernet/MPLS technologies. In principle MANs are connected to the PIONIER Network from at least two independent directions. The DWDM system installed in all relations between MANs has been equipped with two 10 Gbps lambdas. In every MAN there is a 10 Gbps Ethernet/MPLS switch. MANs are connected with 10 GE interfaces or numerous 1 GE interfaces. The system makes it possible to setup scalable point2point networks in the MPLS technology or point-multipoint in the VPLS technology. The structure of the network is shown in Fig. 3.

The PIONIER Network is managed from two Network Operation Centers (NOCs) in Poznań and Łódź. The Poznań NOC is responsible for the DWDM layer and Ethernet/MPLS, while the NOC in Łodź is in charge of the IP layer. Both Centers work in the 24/7 mode. Each node of the PIONIER Network is equipped with out-of-band management system realized as a dedicated VPN established in

Fig. 2 Network of cross-border connections

Fig. 3 The structure of the PIONIER network

the GSM/UMTS network. It enables the monitoring of the condition of the network and network equipment, but also the environmental parameters, such as power, air conditioning, and security of the node. Additionally, the main network nodes host measuring servers with the network synchronization via GPS. Thus it is possible to take such measurements as one-way delay, reordering or jitter in the operation network.

What deserves special attention is the organizational structure of the PIONIER Consortium. It includes 22 academic MANs and the HPC center, and is managed by the PIONIER Board seating 22 members, each having one vote. The PIONIER Executive composed of four members is responsible for the operational management of the Consortium. A "non-profit" organization, the PIONIER Network is financed from members' fees. The decision concerning the algorithm of cost distribution is taken annually by the PIONIER Board. PSNC has been chosen by the Board to be the operator of the PIONIER Network and owner of the national infrastructure. The Center also represents the Polish scientific community in international relations concerning networks.

The Consortium realizes the federation model. Within the federation other consortia constituting subsea of the PIONIER Consortium are founded and dedicated to carry out concrete projects, e.g. five units have gathered in the PL-GRID Consortium to realize the project of building a grid infrastructure.

4 PIONIER for Science

The Polish e-Infrastructure constitutes an integral part of the European Research Area (ERA) and is composed of three layers: network, grid, and scientific data.

The European network infrastructure consists of the GÉANT Network and the national scientific networks connected to it, including the PIONIER Network. PSNC hosts the node of the GÉANT Network. The PIONIER Network uses two main transmission services: the 10 Gbps link to the GÉANT IP Network and 1 GE dedicated channels, in order to setup the VPN networks to support scientific projects. The dedicated channel services are rendered via 10 GE interfaces. Moreover, links to the Commodity Internet are terminated in PSNC – currently a total of 20 Gbps capacity.

Via the PIONIER Network, the Polish scientific community may have broadband access to the Commodity Internet. Thanks to the VPN networks, Polish scientists have an opportunity to actively participate in major European projects. Good examples are radioastronomy and high energy physics.

Connecting the biggest Polish radiotelescope in the town of Piwnice near Toruń enabled Polish radioastronomers to participate in the building of a pilot configuration of e-VLBI in 2005; it was also feasible to take part in the EXPRESS project [3–5]. The project envisaged an attempt to utilize the grid infrastructure related to the network in order to construct a software correlator with no faults of a hardware correlator [6, 7]. The system encompasses a network of file servers associated/connected with radiotelescopes. Divided into packets, the gathered data are sent to the nodes of the dispersed correlator and processed in real time.

In this case, using the network infrastructure has improved the quality of the research. In the near future the research development will require the throughput of 10 Gbps for one radiotelescope. Another major research project named LOFAR originally required 3 Gbps channels, but soon it will also need 10 Gbps. The

PIONIER Network is preparing to support the Polish fragment of the LOFAR infrastructure (three nodes are planned in Poland).

Another example of using the PIONIER Network in research is the high energy physics, including the research associated with the LHC experiment. Poland's participation in preparing LHC is substantial. The PIONIER Network offers physicists access to the VPN network which gives an opportunity to connect to the Tier1 node in Karlsruhe via direct link with the German scientific network named DFN (by means of CBF), thus enabling the scientists to take part in experiments carried out by CERN.

The scientific community gathered around the PIONIER Network actively participates in the research related to the new generation networks. It concerns coordination of two European projects: PHOSPHORUS and PORTA OPTICA STUDY.

A significant achievement in the R&D domain was defining a major European project called PHOSPHORUS (coordinated by PSNC), which cooperates with similar projects in the USA (Enlightened Computing) and Japan (G-Lambda), and refers to integrated management of network and grid resources [8, 9]. Satisfactory results were achieved by designing a new G^2MPLS protocol that makes it possible to recognize and reserve network and grid resources in one stage [10, 11]. The solution is being standardized within the Open Grid Forum.

The concept of development and implementation of the optical scientific networks in Eastern Europe has been developed within the PORTA OPTICA STUDY project managed by PSNC. The achieved results have been widely published and discussed in Europe and the USA [12, 13]. The initial assumptions of the project base on the concept of cross-border optical links formulated in PSNC [2].

PSNC participates in numerous other projects such as ATRIUM [14], GN/GN2/GN3 [15], SEQUIN [16], 6NET [17], MUPBED [18], EMANICS [19] and FEDERICA [20]. Worth mentioning is the Center's participation in projects defining new measurements methods in computer networks and developing standards in this domain, e.g. the perfSONAR system for network monitoring [21–25], the CNIS system for discovery of network topology [26, 27] or the AutoBahn offering bandwidth on demand and deployed in NRENs [28–30]. All mentioned products have been developed within the GN2 project. Moreover, the FEDERICA project defined and implemented the research infrastructure consisting of virtual resources: slices in computing nodes and dedicated channels connecting the nodes. The infrastructure enables researchers to create virtual research infrastructures of European scope with a guarantee of the delivered resources.

Polish scientists are very successful in the domain of grids, and hence become coordinators of research projects on the European level, e.g. GRIDLAB [31], CROSSGRID [32], CHEMOMENTUM [33], RINGRID [34] and DORII [35]. They also take part in numerous grid-related European projects, such as EGEE I/II/III [36], PRACE [37], EUFORIA [38], HPC EUROPE I/II [39], BALTICGRID [40], INT.EU.GRID [41], INTELIGRID [42], OMII_EUROPE [43], G-ECLIPSE [44], QOSCOS GRID [45], ACGT [46], VIROLAB [47] and EUROGRID [48].

The GRIDLAB project was one of the major projects related to the grid environment in Europe and focused on optimization of the processing model. The project brought numerous tools to support users in application development, e.g. Grid Application Toolkit (GAT) and the Grid Resource Management System (GRMS) offering dynamic resource allocation to applications [49–54].

The RINGRID and DORII projects resulted in developing the concept of virtual laboratories by defining their universal architecture which takes into account such features of the equipment characteristic of various domains of science as heterogeneity, specificity, diverse access, and the level of automatization of the task realization. Hence it was possible for scientists in these laboratories to use such research apparatus as: radiotelescopes [55], magnetic resonance spectrometers (NMR) [56, 57] and Remote Oceanographic Instrumentation [58].

As for the infrastructure of the scientific data layer, special attention should be drawn to the cooperation of Polish scientists within the following four European projects: EUROPEANALOCAL [59], ENRICH [60], CACAO [61] and DRIVER/DRIVER2 [62]. The EUROPEANALOCAL project brings the integration of Polish digital resources stored in the Federation of Digital Libraries acting within the PIONIER Network [63, 64].

The Federation includes 42 regional digital libraries that store the resources of several hundreds of such institutions all over Poland (Fig. 4).

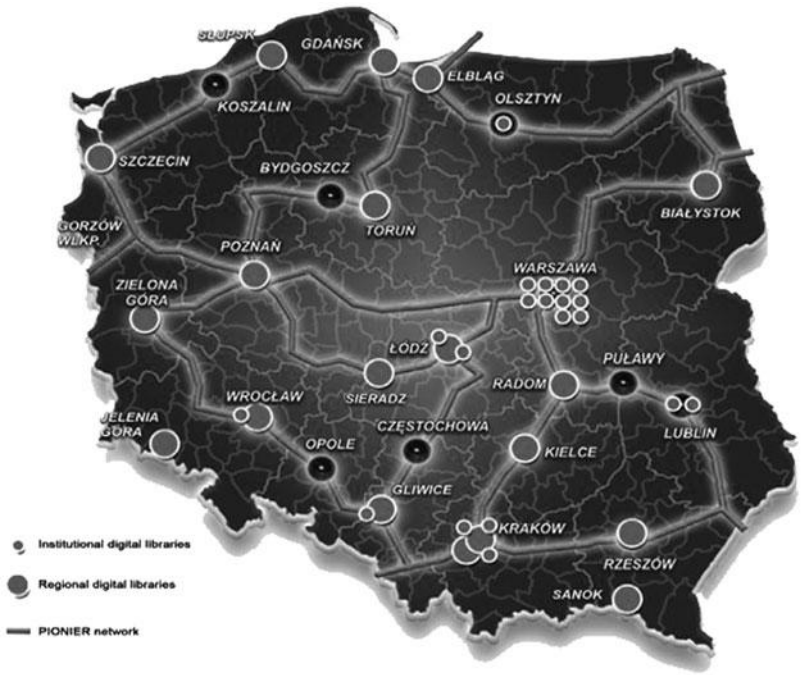


Fig. 4 Installation of digital libraries in Poland

5 Development of e-Infrastructure in Poland

The development of the scientific network infrastructure in Poland has been defined in the PIONIER2 development strategy with the following three fundamental objectives:

– development of integrated service platforms,
– development of the MAN infrastructure,
– construction of a hierarchical optical infrastructure in a backbone network.

The concept also resulted in defining a service platform for science (the PLATON project) that is presently being realized. The platform encompasses 5 services dedicated to the scientific community: HD videoconferencing, EDUROAM, campus computations, archiving and the scientific interactive HD television.

The HD videoconferencing service consists in building an infrastructure enabling audiovisual HD connectivity between MANs and HPC Centers. To this end, videoconferencing rooms equipped with HD terminals will be prepared and arranged in all units of the PIONIER Consortium. A pool of videoconferencing software for PCs will also be available. Additionally, two HD videoconferences bridges as well as servers for transmission archiving system will be installed in the PIONIER Network.

The EDUROAM service aims at building an infrastructure and implementing access to wireless Internet for the benefit of the Polish science and education. The service also offers roaming and user verification by means of certificates. It is planned to purchase and install 21 wireless kits (a kit will consist of 50 radio access points), 22 regional eduroam servers and 2 national eduroam servers.

The campus computations service consists in building an all-Poland computing-service infrastructure providing applications on users' demand. The service will offer flexible and scalable access to applications in both Windows and Linux systems. It will be available in 20 units of the PIONIER Consortium, and each unit will be equipped with a cluster system connected via dedicated channels in the PIONIER Network with the remaining systems in other cities.

The common archiving service provides the scientific community all over Poland with the possibility of data backup and remote archives. The service will offer data encryption and the opportunity to save a copy of the data in two geographically independent archiving centers. It will be rendered on the basis of a geographically distributed data storage infrastructure composed of 10 purchased and installed archiving nodes. The nodes will consist of 120 TB disc matrixes, 2.5 PB tape libraries, and service servers along with specialist software.

The scientific interactive HD television service will consist in developing and launching an integrated service to present and popularize research results. The service will include production, management, broadcast and content storage tools, a dispersed system of the HD content distribution and "live"/"on demand" transmission, as well as an environment of service provision in the "Application on Demand"

mode. It is planned to purchase and launch six production studios, 15 recording studios and a mobile studio (an outside broadcast van) which will be located in MANs and HPC Centers. The infrastructure will make it possible to produce audiovisual materials in the form of programs and live transmissions, and place them in two content repositories. The content will be made available via HD Content Delivery Network to be installed in the PIONIER Network.

6 Conclusions

The Polish e-Infrastructure for e-Science, and particularly the optical PIONIER Network (its main element), acts for the benefit of the whole scientific community by supporting the realization of scientific research in numerous domains, such as radioastronomy, high energy physics, biology, computing chemistry, medicine, technical sciences, etc.

The infrastructure owes its innovative nature to the needs of the scientific community. It is one of the first networks in the world to realize the concept of "all optical networks." Innovative was also the CBDF concept opening new opportunities for scientific cooperation between the neighboring countries, as well as the realization of the pan-European network.

Innovative as it is, the PIONIER Network plays a significant role in the realization of prototype services and application systems for the information society in Poland. An excellent example is the public television content distribution system [65–69] realized in the national iTVP project with the use of the Living Lab model.

It should be emphasized that the building of the infrastructure is a long-standing process that would have never been successful without implementing proper verification mechanisms and the organizational help of the Ministry of Science.

References

1. A. Binczewski, N. Meyer, J. Nabrzyski, S. Starzak, M. Stroiński, and J. Węglarz. First experiences with the Polish optical internet. Computer Networks, 37(6), 747–760, December 2001
2. J. Nabrzyski, S. Starzak, M. Stroiński, and J. Węglarz. Concept of the academic and research European multicolour network, Conference Materials of ISThmus2000, 11–13th April, Poznań, pp. 31–38, 2000
3. N. Kruithof. E-VLBI using a software correlator, Grid enabled remote instrumentation, signals and communication technology. Springer, New York, NY, ISBN 978-0-387-09662-9, pp. 537–544, 2009
4. D. Stokłosa. Workflow management in remote instrumentation infrastructure – based on e-VLBI experiences, INGRID 2008 Conference Materials, Ischia, Italy, 9–11 April, Springer, 2009
5. Ł. Dolata, M. Okoń, D. Stokłosa, Sz. Trocha, N. Meyer, M. Stroiński, D. Kaliszan, T. Rajtar, and M. Lawenda. The user interaction and workflow management in grid enabled e-VLBI experiments. Computational Methods in Science and Technology, 15(1), 2009

6. D. Stoklosa, M. Lawenda, N. Meyer, M. Stroiński, D. Kaliszan, T. Rajtar, and M. Okoń. Workflow management in remote instrumentation infrastructures – e-VLBI experiences, EGEE'07 Conference Budapeszt, 1–5 October, 2007, conference materials
7. M. Okon, D. Stoklosa, D. Kaliszan, M. Lawenda, T. Rajtar, N. Meyer, M. Stroinski, R. Oerlemans, and H.J. van Langevelde. Grid integration of future arrays of broadband radio-telescopes – moving towards e-VLBI. INGRID 2007 Conference, S. Margherita Ligure Portofino, Italy, 16–18 April, Springer, 2007, conference material
8. N. Ciulli, G. Carrozzo, G. Giorgi, G. Zervas, E. Escalona, Y. Qin, R. Nejabati, D. Simeonidou, F. Callegati, A. Campi, W. Cerroni, B. Belter, A. Binczewski, M. Stroiński, A. Tzanakaki, and G. Markidis. Architectural approaches for the integration of the service plane and control plane in optical networks. A Computer Networks Journal, Optical Switching and Networking, 5(2+3), 94–106, ISSN: 1573–4277, 2008
9. B. Belter, A. Binczewski, G. Carrozzo, N. Ciulli, E. Escalona, G. Markidis, R. Nejabati, D. Simeonidou, M. Stroiński, A. Tzanakaki, and G. Zervas. Between GRIDs and networks: GRID-enabled network control planes. Campus-Wide Information Systems, 25(5), 273–286, ISSN: 1065–0741, 2008
10. G. Zervas, E. Escalona, R. Nejabati, D. Simeonidou, G. Carrozzo, N. Ciulli, B. Belter, A. Binczewski, M. Stroinski, A. Tzanakaki, and G. Markidis. Phosphorus grid-enabled GMPLS control plane (G2MPLS): Architectures, services and interfaces. IEEE Communications Magazine, special issue on Multi-Domain Optical Networks: Issues and Challenges, 46(6), 128–137, ISSN: 0163–6804, June 2008
11. E. Escalona, G. Zervas, R. Nejabati, D. Simeonidou, G. Markidis, A. Tzanakaki, G. Carrozzo, N. Ciulli, B. Belter, and A. Binczewski. Deployment and interoperability of the phosphorus grid enabled GMPLS (G2MPLS) control plane, 8th IEEE International Symposium on Cluster Computing and the Grid, 2008. CCGRID apos;08. 19–22 May, pp. 716–721, Digital Object Identifier 10.1109/CCGRID.2008.120, 2008
12. A. Binczewski, M. Przybylski, M. Stroiński, and J. Węglarz. Porta optica distributed optical gateway to GÉANT2. Computational Methods in Science and Technology, 11(2) , ISSN: 1505–0602, 2005
13. A. Binczewski, M. Przywecki, M. Stroiński, W. Śronek, P. Turowicz, and J. Węglarz. Porta optica study – distributed optical gateway to eastern Europe. NATO Science for Peace and Security Series, D: Information and Communication Security, Vol. 20, ISBN 978-1-58603-934-9, 2008
14. ftp://ftp.cordis.europa.eu/pub/ist/docs/rn/atrium.pdf
15. http://www.geant.net/
16. http://stats.dante.org.uk/sequin/
17. http://www.6net.org/
18. http://www.ist-mupbed.org/
19. http://emanics.org/
20. http://www.fp7-federica.eu/
21. J.W. Boote, E.L. Boyd, J. Durand, A. Hanemann, L. Kudarimoti, R. Łapacz, N. Simar, and S. Trocha. Towards multi-domain monitoring for the European research networks, Selected Papers from the TERENA Networking Conference, ISBN 90-77559-04-3, Computational Methods in Science and Technology, 11(2), 91–100, ISSN: 1505-0602, 2005
22. A. Hanemann, J.W. Boote, E.L. Boyd, J. Durand, L. Kudarimoti, R. Łapacz, S. Trocha, M. Swany, and J. Zurawski. PerfSONAR: A service-oriented architecture for multi-domain network monitoring, Proceedings of the 3rd International Conference on Service Oriented Computing, Springer, LNCS 3826, ACM Sigsoft and Sigweb, Amsterdam, The Netherlands, pp. 241–254, December 2005
23. J. Zurawski, J. Boote, E. Boyd, M. Głowiak, A. Hanemann, D.M. Swany, and S. Trocha. Hierachically federated registration and lookup within the perfSONAR framework, in moving from bits to business value, Proceedings of the 2007 Integrated Management Symposium, 2007, IFIP/IEEE, München, Germany, May, 2007

24. A. Hanemann, L. Kudarimoti, S. Kraft, N. Simar, and S. Trocha. Implementing multi-domain monitoring services for European research networks. TERENA Networking Conference 2007, Lyngby, Denmark, 2007 http://tnc2007.terena.org/programme/presentations/show.php?pres_id=23
25. A. Binczewski, M. Lawenda, R. Łapacz, and S. Trocha. Application of perfSONAR architecture in support of GRID monitoring, Proceedings of INGRID 2007: Grid Enabled Remote Instrumentation, Signals and Communication Technology, Springer, USA, ISBN 978-0-387-09662-9, pp. 447–454, 2009
26. M. Łabędzki, C. Mazurek, A. Patil, and M. Wolski. Modeling and interacting with a network using common Network Information Service, Artykuł zaakceptowany na The TERENA Networking Conference 2009, Malaga, 2009
27. G. Cesaroni, M.K. Hamm, M. Labedzki, G. Vuagnin, M. Wolski, and M. Yampolskiy. I-SHARe a process support tool for Multi-Domain Services, The TERENA Networking Conference 2009, Malaga, Spain, 2009
28. M. Campanella, R. Krzywania, V. Reijs, and A. Sevasti. The bandwidth on demand service for the European research and education networks, Herakleion (Greece) 2006, International Conference on Photonics in Switching 2006 http://www.garr.it/documenti/PS2006-GN2-JRA3_final-02.pdf
29. A. Binczewski, L. Grzesiak, E. Kenny, M. Stroiński, R. Szuman, S. Trocha, and J. Weglarz. Monitoring solution for optical grid architectures, INGRID 2007, Grid Enabled Remote Instrumentation – Springer ISBN 978-0-387-09662-9, S. Margherita Ligure, Italy 2007
30. M. Campanella, R. Krzywania, and A. Sevasti. A bandwidth-on-demand system case-study based on GN2 project experiences, TERENA Networking Conference 2007, Lynby, Denmark, 2007, http://www.garr.it/documenti/TNC2007_GN2-JRA3_BoD-case-studies.pdf
31. http://www.gridlab.org/
32. http://www.eu-crossgrid.org/
33. ftp://ftp.cordis.europa.eu/pub/ist/docs/grids/chemomentum_en.pdf
34. http://www.ringrid.eu/
35. http://www.dorii.eu/
36. http://www.eu-egee.org/
37. http://www.prace-project.eu/
38. http://www.euforia-project.eu/EUFORIA/
39. http://www.hpc-europa.eu/
40. http://www.balticgrid.org/
41. http://www.i2g.eu/
42. http://inteligrid.eu-project.info/
43. http://www.omii-europe.org/
44. http://www.geclipse.eu/
45. http://www.qoscosgrid.eu/
46. http://www.eu-acgt.org/
47. http://www.virolab.org/
48. http://www.eurogrid.org/
49. M. Kosiedowski, K. Kurowski, C. Mazurek, J. Nabrzyski, and J. Pukacki. Workflow applications in GridLab and PROGRESS projects: Research articles. Concurrency and Computation: Practice and Experience18(10), 1141–1154, August 2006. doi: http://dx.doi.org/10.1002/cpe.v18:10
50. P. Grabowski, K. Kurowski, J. Nabrzyski, and M. Russell. Context sensitive mobile access to grid environments and VO workspaces, In Proceedings of the 7th International Conference on Mobile Data Management (MDM'06), pp. 87–96, Japan, 2006
51. M. Adamski, P. Grabowski, K. Kurowski, B. Lewandowski, B. Ludwiczak, J. Nabrzyski, A. Oleksiak, T. Piontek, J. Pukacki, and R. Strugarski. GridLab middleware services for managing dynamic computational and collaborative infrastructures (VOs), In Proceedings of the Mardi Gras Conference 2005, Baton Rouge, LA, USA

52. M. Adamski, K. Kurowski, R. Strugalski, and J. Nabrzyski. The grid authorization service – internal design, interaction with external services and management, In Proceedings of the 1st Cracow Grid Workshop, Krakow, Poland, 2004
53. K. Kurowski, B. Ludwiczak, J. Nabrzyski, A. Oleksiak, and J. Pukacki. Dynamic grid scheduling with job migration and rescheduling in the GridLab resource management system. Science Program, 12(4) 263–273, December 2004
54. I. Foster. The grid: Computing without bounds. Scientific American, April 2003
55. M. Okon, D. Stoklosa, R. Oerlemans, H.K. van Langevelde, D. Kaliszan, M. Lawenda, T. Rajtar, N. Meyer, and M. Stroinski. Grid integration of future arrays of broadband radio-telescopes moving towards e-VLBI, Grid Enabled Remote Instrumentation. Springer, New York, NY, p. 571, 2008
56. L. Handschuh, M. Lawenda, N. Meyer, P. Stępniak, M. Figlerowicz, M. Stroiński, and J. Węglarz. Virtual laboratory and its application in genomics, in F. Davoli, N. Meyer, R. Pugliese, S. Zappatore, Eds., Remote Instrumentation and Virtual Laboratories – Service Architecture and Networking, Springer, New York, NY, 2010; ISBN 978-1-4419-5595-1, pp. 43–57
57. L. Handschuh, M. Lawenda, P. Stępniak, M. Figlerowicz, M. Stroiński, and J. Węglarz. New approach to genomics experiments taking advantage of virtual laboratory system. Computational Methods In Science and Technology, 15(1), 31–40, ISSN 1505-060, 2009
58. S. Salon, P.-M. Poulain, E. Mauri, R. Gerin, D. Adami, and F. Davoli. Remote oceanographic instrumentation integrated in a GRID environment. Computational Methods in Science and Technology, 15(1), 49–55, ISSN 1505-0602, , 2009
59. http://www.europeanalocal.eu/
60. http://enrich.manuscriptorium.com/
61. http://www.cacaoproject.eu/
62. http://www.driver-repository.eu/
63. A. Dudczak, C. Mazurek, and M. Werla. RESTful atomic services for distributed digital libraries. 1st International Conference on Information Technology, Gdańsk, 18–21 May, ISBN 978-1-4244-244-9, pp. 267–270, 2008
64. C. Mazurek, T. Parkoła, and M. Werla. Building federation of digital libraries basing on concept of atomic services. In ACM/IEEE Joint Conference on Digital Libraries, JCDL 2008, Pittsburgh, PA, USA, 16–20 June, ACM 2008, ISBN 978-1-59593-998-2, p. 458, 2008
65. M. Czyrnek, C. Mazurek, and M. Stroiński. Two-level content delivery system in PIONIER optical network for interactive TV services. Proceedings of 11th International Workshop on Systems, Signals and Image Processing, IWSSIP'04, Ambient multimedia, Bartkowiak M. et al. Eds., 13–15 September, Poznań, PTETiS, pp. 457–460, 2004
66. M. Czyrnek, C. Mazurek, and M. Stroiński. Management and publishing of digital content on the iTVP Platform. Proceedings of 11th International Workshop on Systems, Signals and Image Processing, IWSSIP'04, Ambient multimedia, Bartkowiak M. et al. Eds., 13–15 September, Poznań, PTETiS, pp. 481–484, 20004
67. M. Czyrnek, E. Kusmierek, C. Mazurek, and M. Stroinski. Large-scale multimedia content delivery over optical networks for interactive TV services. Future Generation Computer Systems, 22(8), 1018–1024, October 2006
68. M. Czyrnek, E. Kuśmierek, C. Mazurek, M. Stroiński, and J. Węglarz. CDN for live and on-demand video services over IP.Content delivery networks series: Lecture notes electrical engineering, vol. 9,, R. Buyya, M. Pathan, A. Vakali, Eds., ISBN 978-3-540-77886-8, pp. 317–342, 2008
69. E. Kuśmierek, M. Czyrnek, C. Mazurek, and M. Stroinski. iTVP: Large-scale content distribution for live and on-demand video services. Multimedia computing and networking SPIE-IS&T electronic imaging, vol. 6504, R. Zimmermann, C. Griwodz, Eds., SPIE, Article CID 6504-8, 2007

Part IV
Applications in User Communities

The Impact of Global High-Speed Networks on Radio Astronomy

An Operational Viewpoint

M. Leeuwinga

Abstract The advent of worldwide high-speed optical networks has drastically changed the operational model of VLBI (Very Long Baseline Interferometry). This document describes these changes through a time line of an e-VLBI observation at the EVN-correlator from an operational point of view. All the tools that are needed to setup and run a successful e-VLBI session are discussed in chronological order. Tools developed especially for e-VLBI by the EXPReS project group will be mentioned and suggestions on possible improvements are given where appropriate. Because the incoming data from the telescopes are being processed at the correlator in real time, problems with the correlator system can cause total data loss from all the participating telescopes. It is therefore important to realize that the job of the correlator operator is directly related to the success of the e-VLBI observation, with respect to the monitoring of the data quality and error checking. New tools that will help the operator detect and solve problems in the data or with the network need to be developed. Our experiences from working with this online scientific instrument can be beneficial to other users or developers of online and distributed Grid based systems.

1 Introduction

Very Long Baseline Interferometry (VLBI) [1] is a technique used in radio-astronomy in which signals received at several distant radio telescopes are combined to form one large telescope, producing images of stars and galaxies with very high resolution. The resolution of the image depends on the size of the network (distance between the telescopes). The sensitivity of the instrument depends on the total collecting area of all the participating telescopes and the bandwidth of the connection between the telescopes.

M. Leeuwinga (✉)
Joint Institute for VLBI in Europe (JIVE), Postbus 2, 7990 AA Dwingeloo, The Netherlands
e-mail: leeuwinga@jive.nl

F. Davoli et al. (eds.), *Remote Instrumentation Services on the e-Infrastructure*,
DOI 10.1007/978-1-4419-5574-6_15,

Combining the data of all the telescopes is done with a purpose-built "supercomputer" called a data processor or correlator [2]. To get the data to the correlator, the received signals at all the telescopes are recorded on large capacity disk packs and shipped by postal mail to the correlator site. Apart from recording the data on disks, e-VLBI works in the exact same way as regular VLBI, except in this case the data from each telescope is streamed to the correlator over high-speed fiber-optic networks and correlated in real time.

The Joint Institute for VLBI in Europe[1] (JIVE), located in Dwingeloo in The Netherlands is operating the correlator for the European VLBI Network[2] (EVN).

A typical VLBI observation with the EVN can use telescopes ranging from the UK to China, including The Netherlands, Germany, Poland, Sweden, Finland, Italy, Spain and South Africa.

2 The Road To E-Vlbi

The European VLBI Network started using disk-based recording systems back in 2003. Before the disk era, tape recorders and large magnetic tapes were used as a recording medium. The theoretical recording speed for tapes was 1 Gbit/s, as is the case now with disk recordings. In practice though, it was virtually impossible to record, or playback, the tapes at that data rate because of various tape guidance problems. In general, 256 Mbit/s was the maximum obtainable data rate with tapes, making them last for about 6 h before they were fully written. Nowadays recording data at 1024 Mbit/s on a disk pack is no problem at all. This makes the disk-based systems fully compliant with the specifications of the VLBI Standard Interface [3], which was developed as a joint effort between the geodesy and astronomy communities.

The continuous increase in capacity of hard disks in general has worked to the advantage of VLBI. The earlier disk packs consisted of eight 120 Gigabyte Parallel ATA disks, housed in a chassis, that can be inserted into the recording systems as one pack. The latest disk packs are outfitted with eight times 750 Gigabyte PATA disks, adding up to a total capacity of 6,000 Gigabytes of storage space. At a recording speed of 1024 Mbit/s, these packs can be used for more than 13 h unattended. The chassis of the disk pack has recently been upgraded so that it can now be populated with Serial ATA disks as well, making it ready for the new generation of commercially available hard disks of even bigger capacities.

The disk recording and playback equipment is called the Mark5 [4]. It was developed at Haystack observatory[3] of the Massachusetts Institute of Technology in the US. Figure 1 shows the Mark5 with the top cover removed. The Mark5 recording system, the successor of the Mark4 tape recorder, is basically a Linux PC outfitted

[1] http://www.jive.nl

[2] http://www.evlbi.org

[3] http://www.haystack.mit.edu/

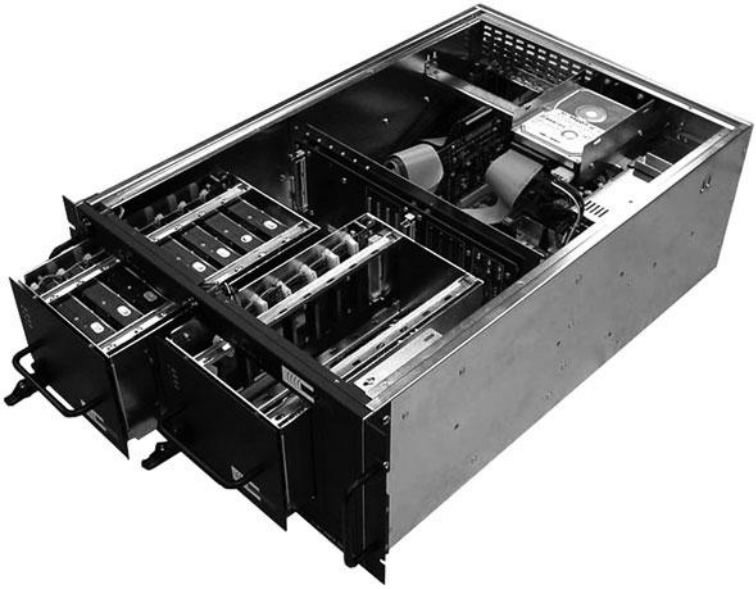


Fig. 1 Mark5 unit with one fully populated disk pack and one empty disk chassis

with a high-end dual processor server board with two 1 Gigabit Ethernet interfaces. This connectivity gives us the capability to stream the data to the correlator at 1 Gbit/s over one network connection, while at the same time we can send commands and status checks from the correlator to the Mark5s over the other network interface.

At the correlator side there are 16 Mark5s, normally used to playback the recorded data from disk packs, up to a maximum of 16 stations (telescopes) at a time at 1024 Mbit/s each. In the case of e-VLBI each Mark5 at the telescopes is connected to one Mark5 at the correlator. The data at the sending Mark5 is streamed onto the network in the same format in which it is normally written onto the disk pack. At the receiving end, at the correlator, the data comes into the Mark5's high-speed network connection and is passed on to the correlator. Since there is no difference in the format of the data of regular disk-based VLBI and e-VLBI, the rest of the correlator system behind the Mark5s does not need any hardware changes.

The network infrastructure between the correlator and the different telescopes is made-up of different types of networks [5]. Figure 2 illustrates where all the e-VLBI telescopes are located across the globe. Table 1 gives an overview of the connectivity between the correlator and the telescopes.

With these global high-speed networks in place, most of the European VLBI telescopes are now connected to the central correlator at a bandwidth that equals, or is close to, the data rate of that of disk packs (1024 Mbit/s). Unfortunately there is still not sufficient bandwidth to Hartebeesthoek, Sheshan, and to the two stations in South America, Arecibo and TIGO. These very long baselines are very important for the sensitivity of the interferometer.

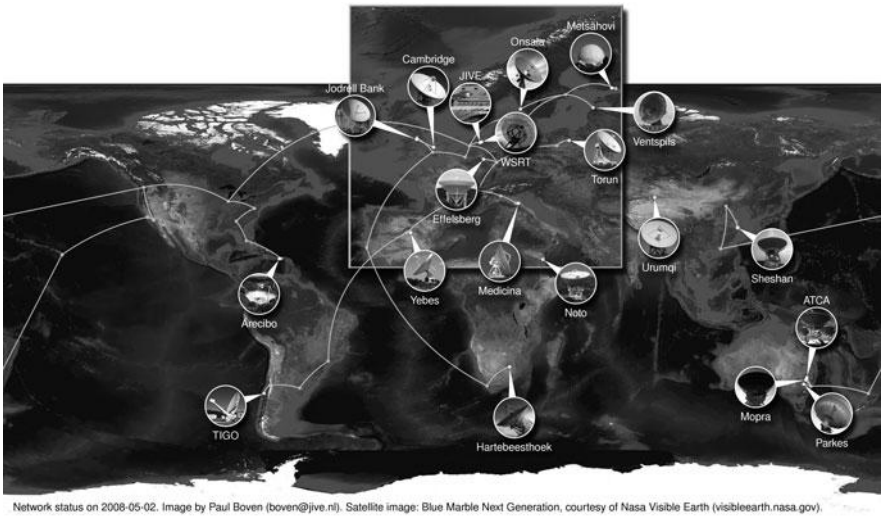


Fig. 2 World map with the locations of all the current e-VLBI telescopes

Table 1 Overview of the connectivity between the correlator and each telescope

Telescope	Country	Type	Bandwidth
Westerbork	The Netherlands	Dark fiber	2 × 1 Gbit/s
Jodrell Bank + Cambridge	United Kingdom	2 × Light path	2 × 1 Gbit/s
Onsala	Sweden	VLAN	1.5 Gbit/s
Torun	Poland	Light path + Routed	1 + >1 Gbit/s
Medicina	Italy	Light path	1 Gbit/s
Effelsberg	Germany	VLAN	10 Gbit/s
Metsähovi	Finland	Routed	10 Gbit/s
Sheshan	China	Light path	622 Mbit/s
Hartebeesthoek	South Africa	SAT-3	64 Mbit/s
Arecibo	Puerto Rico	VLAN	512 Mbit/s
TIGO	Chile	Routed	64 Mbit/s
ATNF	Australia	Light path	1 × 1 Gbit/s
Yebes	Spain	Routed	1 Gbit/s

The SAT-3 network connection to Hartebeesthoek is routed traffic over an undersea Internet cable

All the e-VLBI developments and investments in infrastructure were done by the EXPReS[4] project. EXPReS stands for Express Production Real-time e-VLBI Service. This is a 3 year project, started March 2006, funded by the European Commission (DG-INFSO), Sixth Framework Programme, Contract #026642.

[4] http://www.expres-eu.org/

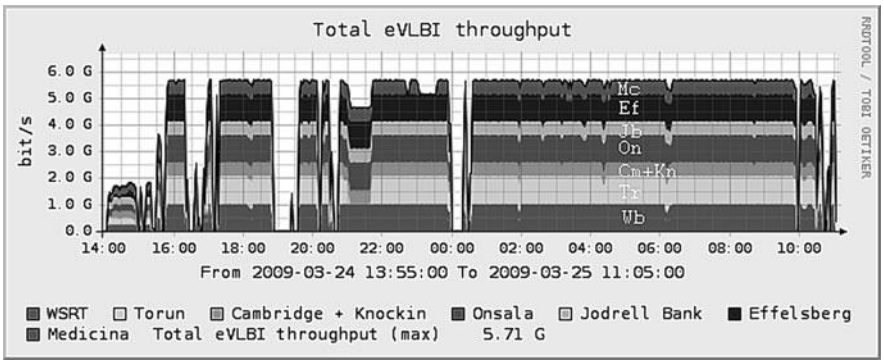


Fig. 3 Graph of the network throughput of a 19-h science observation with eight stations on 24–25 March 2009. Westerbork, Torun, Onsala, and Effelsberg observed at 1024 Mbit/s. Cambridge + Knocking, Jodrell Bank, and Medicina observed at 512 Mbit/s

More than once JIVE has had the opportunity to showcase the results of their e-VLBI efforts to a broader audience, for instance in a demonstration at the opening of the International Year of Astronomy 2009, when we conducted a nearly continuous 33-h observation with seventeen different telescopes around the world.[5]

As an example of the performance of the networks, Fig. 3 shows a network throughput graph of a 19-h science observation with 8 stations in Europe on 24–25 March 2009. Westerbork, Torun, Onsala, and Effelsberg observed at 1024 Mbit/s. Cambridge + Knocking, Jodrell Bank, and Medicina observed at 512 Mbit/s. Cambridge and Knocking are sharing the same network, so each of them are observing at 256 Mbit/s.

The major benefits of doing e-VLBI are:

- Eliminate weeks from the observation and correlation schedule.
- Monitor problems in data collection in real time.
- Detect transient events and schedule near-immediate follow-on observations.

The three abovementioned benefits of doing e-VLBI can be ascribed to the real-time nature of an e-VLBI observation. A normal VLBI session takes place three times a year and usually lasts between 3 and 4 weeks of almost continuous round the clock observing. After the observing session is done, the telescopes send their disk packs to the correlator. On average, the correlator receives somewhere between 150 and 200 disk packs from all the different telescopes. Correlating all these experiments then takes about 3–4 months, after which the disk packs are erased and send back to the telescopes for the next observing session. e-VLBI, on the other hand, knows none of these delays. Correlating the data happens in real time while it is being observed and the astronomer can start working on his data the very next day after the observation.

[5] http://www.expres-eu.org/IYA2009_opening.html

Comparing the time line of an e-VLBI observation to that of a conventional VLBI observation also explains why this real-time form of VLBI is so useful for detecting transient events in the sky and for quickly observing astronomical phenomena that require immediate follow-on observation.

3 Doing e-VLBI

Next we will discuss the various steps of an e-VLBI observation in chronological order and talk about the different tools that are needed to control and monitor all the systems and processes at the correlator and the telescopes. Table 2 shows a listing of all the necessary steps towards a successful e-VLBI science observation.

Table 2 Necessary steps to run an e-VLBI observation

Type	Activity	Tool
Setup	*Start of schedule*	
	Setting up the networks	*ControlEvlbi*
	Starting the Mark5s at the correlator	Controlmk5
	Starting the Mark5s at the stations	Controlstation
	Starting the Correlator Control Software	Start_procs
	Loading the schedule	Start_runjob
	Starting a run	Processing Job
	Synchronizing the data	Clock Searching
Science	Start of the science observation	
	Data quality analysis and feedback	Data Status Monitor

3.1 Start of Schedule

The first 4 h of the observing schedule are reserved for setting up the entire system and for correcting all the delays between the telescopes. All apparent network problems and the inevitable last minutes changes to the software need to be solved in this preparation period. Although the telescopes are already on source and observing according to the schedule, this is not part of the science observation yet, so the PI (Principal Investigator, astronomer who requested the observation) is not losing any "science" data.

3.2 Setting up the Network

Tool: ControlEvlbi

The tool used for connecting the telescope's Mark5s to the local Mark5s at the correlator is called "ControlEvlbi", specially developed for e-VLBI by the EXPReS group.

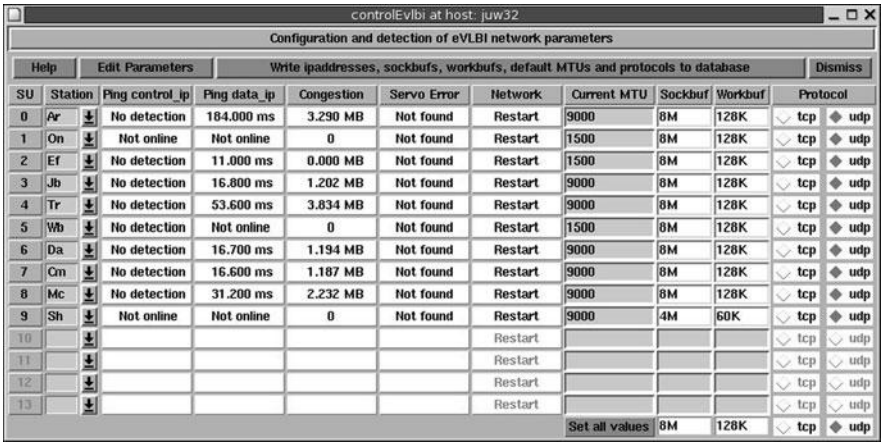
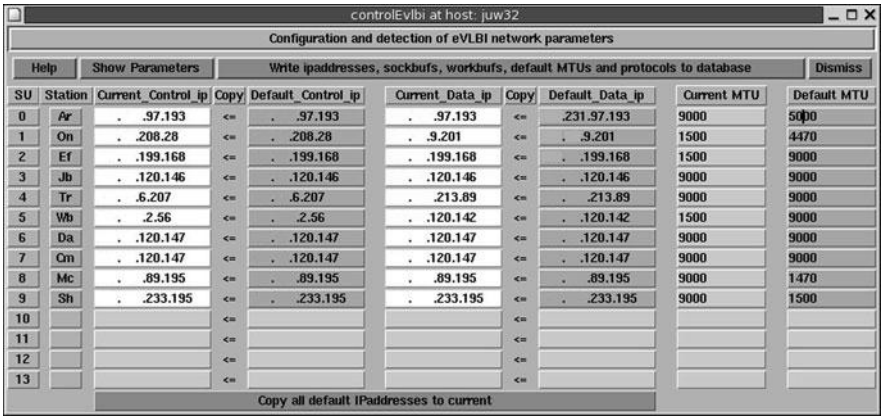


Fig. 4 ControlEvlbi connects each telescope to one of the inputs of the correlator

ControlEvlbi has two screens. In Fig. 4 the stations, represented by their abbreviations Ar, On, Ef, etc, are each linked to one of the Mark5s at the correlator under the column named SU (Station Unit). The columns Ping control IP and Ping data IP indicate whether the remote Mark5s are visible through their respective control and data interfaces. The values in the columns Congestion and Servo Error are derived from the results of the Ping data IP. Looking at the column Protocol on the right it is obvious that e-VLBI data transfer is done using the UDP protocol, with which significant higher data rates are obtained with respect to the use of the TCP protocol. Of the three remaining columns called Current MTU, Sockbuf, and Workbuf, it is the MTU size that varies per station, depending on the characteristics of the network connecting the correlator to the telescope.

The IP addresses of all the Mark5s for both the control and the data interfaces are stored in a database. In case changes need to be made to the IP addresses or MTU sizes, a second screen has been added to controlEvlbi (Fig. 5). In this screen the operator can change any IP address or MTU size and write the new settings into the database and upload them into the local Mark5s.

In the first attempts to do e-VLBI with two stations connected to the correlator, ControlEvlbi had not been developed yet. Every IP address, Sockbuf size, Workbuf size and MTU size had to be manually entered each time a new correlation run was started or the hardware was reset. When more stations got involved in doing e-VLBI it became impossible to keep track of all these different settings without the aid of an automated tool. Especially when one decided to swap the connection of two telescopes there was the risk of getting completely entangled in IP addresses. ControlEvlbi takes care of all the connectivity for us; we can now link any telescope to any input of the correlator without worrying about the underlying network implications.



SU	Station	Current_Control_ip	Copy	Default_Control_ip	Current_Data_ip	Copy	Default_Data_ip	Current MTU	Default MTU
0	Ar	. .97.193	<=	. .97.193	. .97.193	<=	.231.97.193	9000	5000
1	On	. .208.28	<=	. .208.28	. .9.201	<=	. .9.201	1500	4470
2	Ef	. .199.168	<=	. .199.168	. .199.168	<=	. .199.168	1500	9000
3	Jb	. .120.146	<=	. .120.146	. .120.146	<=	. .120.146	9000	9000
4	Tr	. .6.207	<=	. .6.207	. .213.89	<=	. .213.89	9000	9000
5	Wb	. .2.56	<=	. .2.56	. .120.142	<=	. .120.142	1500	9000
6	Da	. .120.147	<=	. .120.147	. .120.147	<=	. .120.147	9000	9000
7	Cm	. .120.147	<=	. .120.147	. .120.147	<=	. .120.147	9000	9000
8	Mc	. .89.195	<=	. .89.195	. .89.195	<=	. .89.195	9000	1470
9	Sh	. .233.195	<=	. .233.195	. .233.195	<=	. .233.195	9000	1500
10			<=			<=			
11			<=			<=			
12			<=			<=			
13			<=			<=			

Fig. 5 ControlEvlbi: useful tool for easy changing and uploading network settings into the database

3.3 Starting the Mark5s at the Correlator

Tool: Controlmk5

Controlmk5 (Fig. 6) is a tool that lets us manage one or more Mark5s at the same time. This interface only controls the Mark5s at the correlator; it will not affect the Mark5s at the telescopes. It is also used in regular VLBI. Apart from starting, stopping and rebooting the Mark5s, we can issue any command from the command set and send it to the selected units.

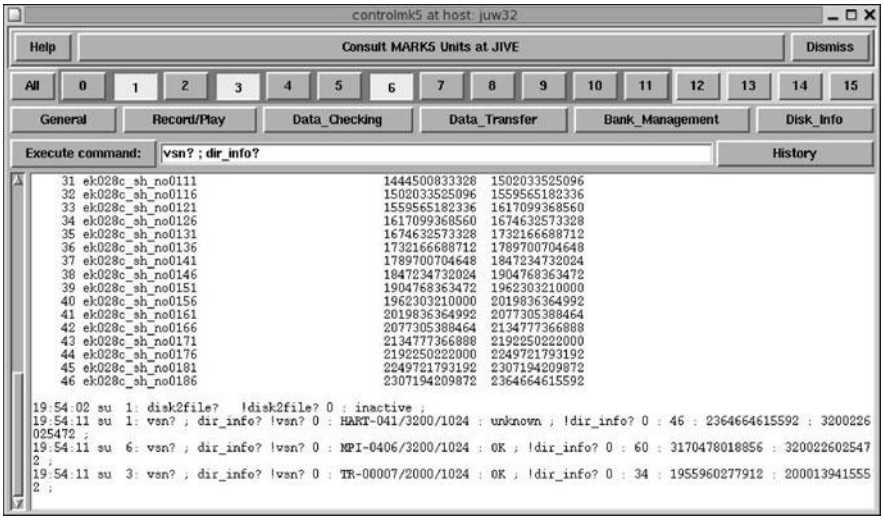


Fig. 6 Controlmk5: group control over all the Mark5s at the correlator

Before we had this tool at our disposal we had to open a remote window to each Mark5 and type in the required commands for each of the Mark5s that we wanted to query or manipulate. This would not only clutter up the screen with dozens of windows making you loose oversight, but it was also very inconvenient to type in the same command 16 times to query 16 different Mark5s. Controlmk5 gives us group control over all the Mark5s.

3.4 Starting the Mark5s at the Stations

Tool: Controlstation

Figure 7 shows a screenshot of Controlstation, an interface that was specially developed for e-VLBI to control the Mark5s at the telescopes. The functionality of Controlstation is exactly the same as that of Controlmk5, but this time all the commands are executed on the remote Mark5s at the telescopes.

The need for this tool became clear after the first few e-VLBI observations when the correlator operators constantly had to ask the observers at the stations to check the status of their Mark5s, or to change some of the settings. Communication between correlator operators and station observers is done in the form of a Skype chat session. Errors in the data or problems with the network transfers are often only visible at the correlator. Soon, it became apparent that it was much easier for the correlator operator to have control over each station's Mark5 instead of skyping questions and answers back and forth. With the level of control provided by Controlstation, it is not uncommon for a station to run an overnight e-VLBI observation while the telescope is unmanned, with the observer merely being available on call.

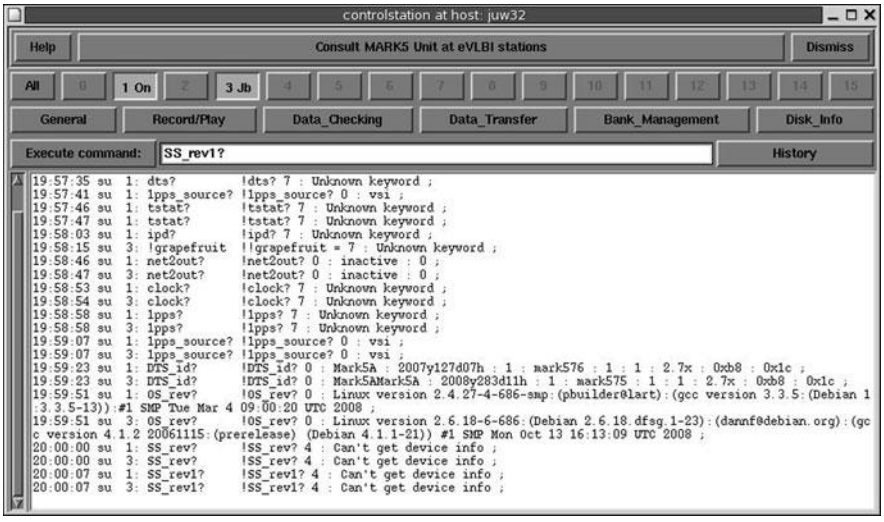


Fig. 7 Controlstation: remote group control over the Mark5s at the telescopes

3.5 Starting the Correlator Control Software

Tool: Start_procs

The Correlator Control Software (CCS) coordinates the processes of all the different parts of the correlator. The interface to the users is called Status Monitor (Fig. 8). Some of the parts that make up the correlator are: the actual data processor, the data distribution unit (DDU), all 16 data playback units (Mark5), the timing equipment (TSPU) and the data handling of the correlator output data.

When something in the hardware or software goes wrong, or when there is a problem in the timing between different processes the Status Monitor will generate a specific error message. During a correlation run dozens of different processes need to communicate with each other. The CCS makes sure that this all happens in an organized way and that each process waits its turn. The remote Mark5s at the stations are also controlled by the CCS. They get their start and stop command from the CCS and during a run the CCS is controlling the synchronization between the stations and the correlator. This software has been in use for as long as the correlator has existed, but particularly after the introduction of e-VLBI the CCS can be considered a centralized automated control tool.

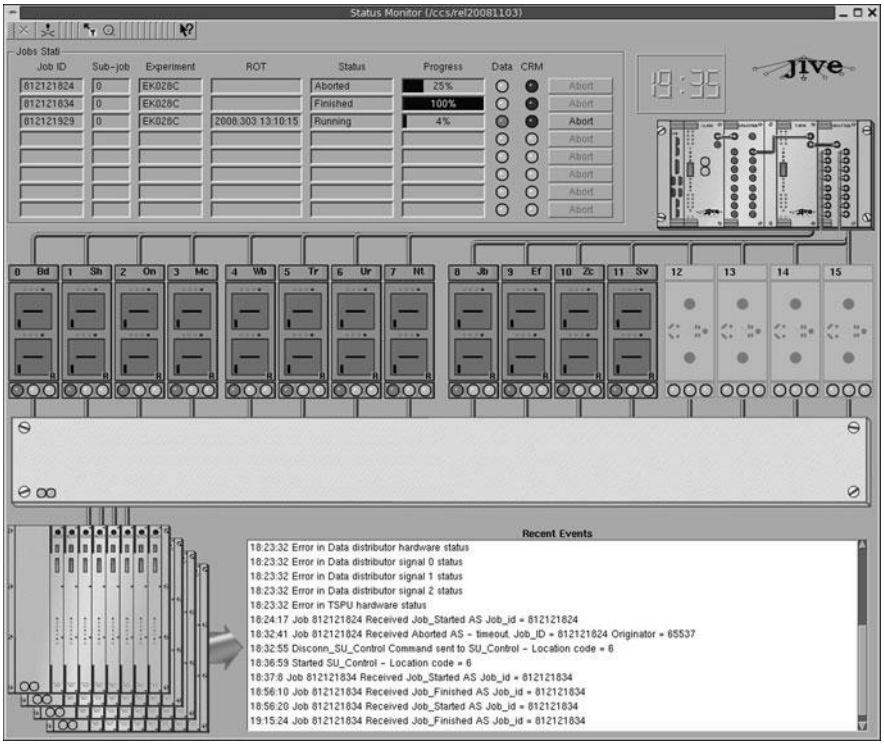


Fig. 8 Status Monitor. Interface to the correlator Control Software. It coordinates all the different processes involved in running a correlation job, including the Mark5s at the telescopes

3.6 Loading the Schedule

Tool: Start_runjob

The duration of the schedule is broken down in small parts called scans. In order to calibrate the data afterwards, the observation frequently needs to alternate between the target source and a bright calibrator source. A transition in source means a new scan in the schedule. Runjob (Fig. 9) is a tool that reads in the observing schedule and lists the individual scans vertically in a graphical user interface. Single or multiple scans can be selected and started.

During the observation the stations are running the same schedule on their control computers, but, since no two telescopes are the same, with additional parameters that are unique for each telescope. These additional parameters deal with matters like the mechanical limits of the telescopes, bandwidth restrictions of receivers, and system calibration procedures. The schedules at the stations run independently from that of the correlator. If, for some reason, the correlation process is interrupted the stations

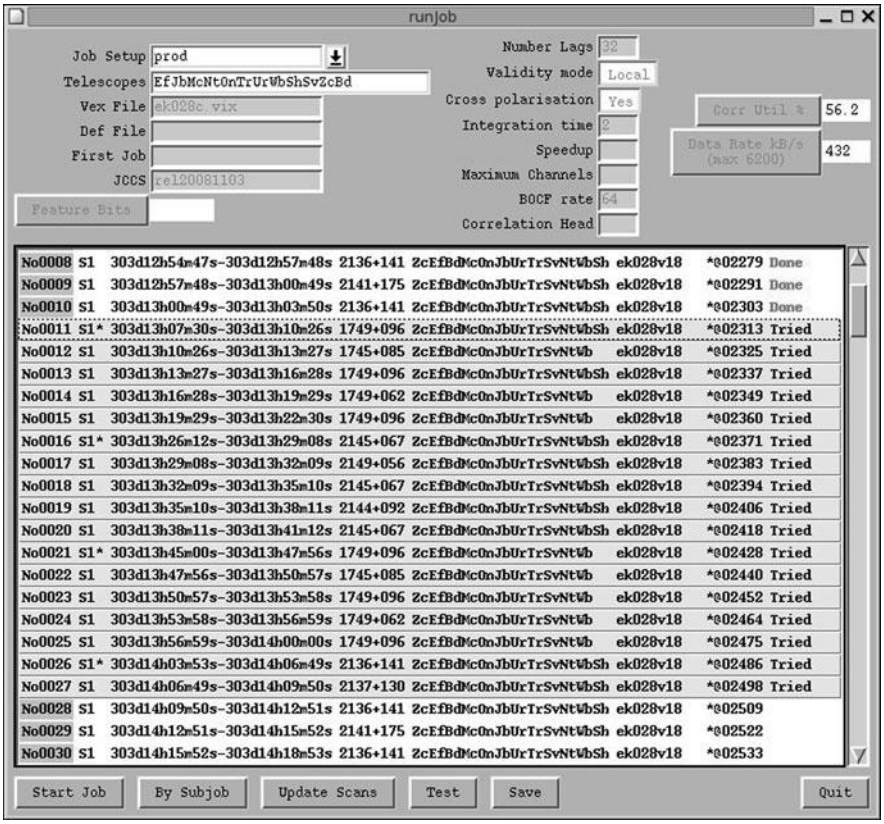


Fig. 9 Runjob puts the observing schedule, scan by scan, in chronological order and allows one to select and start any range of scans

just continue observing according to the schedule. When the correlation schedule is resumed the correlator will pick up the data that is coming from the telescopes at that very moment.

An e-VLBI observation is in essence a single user drawing data from multiple locations. Further, that user is always the correlator operator at the operations centre at JIVE. Because of this structure it is unnecessary to implement services such as the Grid middleware layer that exists to support flexibility of general purpose computational Grids.

3.7 Potential Setup Problems

The five programs we have discussed so far are all part of the pre-correlation preparation; an actual correlation run has not taken place yet. Getting these tools and interfaces up and running will take somewhere between 10 and 15 min..

The weak link in this chain, which could seriously delay the start of the observation, is controlEvlbi. Or rather, the thing it tries to accomplish, which is setting up the network connections between the stations and the correlator. It is fair to say that the weakness is not the actual tool itself, but the environment in which it is used. Setup time before a science observation tends to get used for last minute testing of changes in the hard- or software.

A large scientific research facility like this correlator, together with a network of radio telescopes, connected through a Grid of high speed networks will always be subject to constant development and testing. But once the use of this instrument has moved from the R&D stage into the operational phase, there should be a more distinct line between testing and operations. Especially when the EVN is advertizing the availability of e-VLBI to the international astronomical community and is calling for proposals, it is not desirable to test changes made to the software or hardware, either at the correlator or at the stations, shortly before a science observation. System- or software changes should be tested in special non-science test observations.

The temptation to use the preparation time before a science observation for all sorts of tests lies in the fact that it is often hard to get test time on telescopes. A lot of things can be tested without the use of a telescope signal, for instance the network connections between the Mark5s. But as soon as you want to test with real telescope data, you need a schedule, an observer, but most of all, you are using up precious telescope time that comes out of the station's daily observing program.

We are now two steps away from a successful e-VLBI observation. The first one, obviously, is to start a run. The second one is to find the delays between the stations and correcting them in a process we call "clock searching".

3.8 Starting a Run

Tool: Processing Job

In Runjob (Fig. 9) we start a range of scans that we want to correlate and the computer now calculates the models of the different stations that participate in this

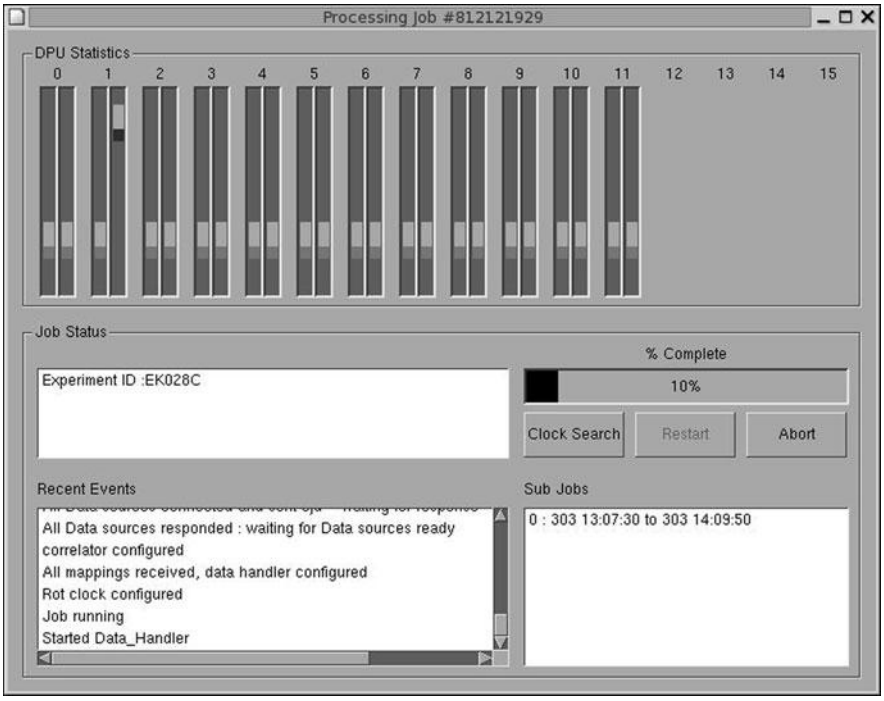


Fig. 10 The Processing Job window passes the calculated models of the telescopes trough to the CCS. The Clock Search functionality activates the automatic fringe detecting and correcting tool on the current scan

run. These models contain parameters of the stations such as the geometrical positions of the telescopes, earth orientation parameters (e.g. continental drift, ocean tidal forces) and atmospheric models. The results of these computations are a number of settings for the correlator hardware that will apply a delay to the data stream from each station (see clock searching).

The correlator configuration, derived from the telescope models, is passed on to the "Processing Job" window (Fig. 10).

The Mark5s at the telescopes are now told to stream their data to their counterpart at the correlator and the Mark5s at the correlator are told to listen to their assigned Mark5s at the stations. As soon as the timing system notifies all the processes that the start-time of the selected scans has been reached, the correlator will start integrating the incoming data streams.

Activating "Clock Search" in the Processing Job window will select the current scan to be used as a clock search scan.

3.9 Synchronizing the Data

Tool: Clock searching

This step in the process is probably the most important part of doing VLBI. The only way the correlator is able to find a relationship between signals from different

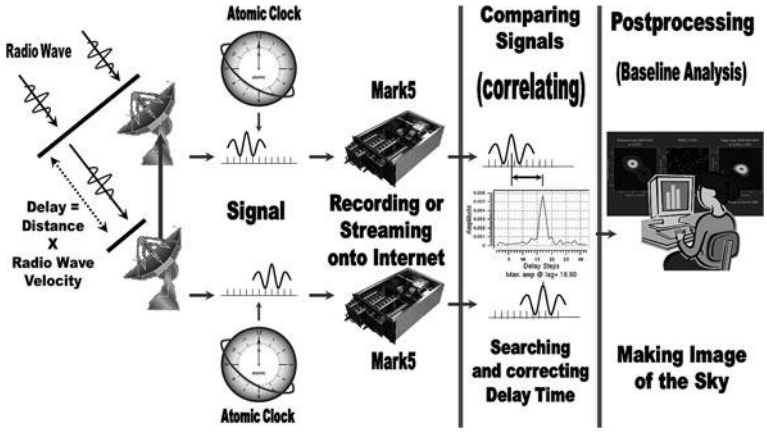


Fig. 11 Basic principle of VLBI

telescopes is to correlate signals belonging to the same wave front. Therefore we have to make sure that the signals from all the telescopes arrive at the correlator at the same time. This has to be done with the greatest possible accuracy.

Figure 11 illustrates the basic principles of VLBI. From the illustration we can see that, because of their geographical separation, the wave front from the celestial source will hit the surface of the upper telescope first and after a tiny delay, that same radio wave will arrive at the bottom telescope. Each station is equipped with a highly accurate atomic frequency standard in the form of a hydrogen maser [6], locked to a GPS time standard, which is used to sample the received signal and turn it into a digital data stream. Headers with timing information are added to the digital data stream; which is then either recorded onto disk packs, or streamed onto the network (e-VLBI), by the Mark5 unit. Over the course of an observation the positions of the telescopes with respect to the source in the sky keep changing due to the rotation of the earth. As a result of that, the delays between the signals of the telescopes constantly need changing.

Synchronizing the data takes place at the correlator during the correlation process. The delay settings that compensate the earth's rotation and the estimated times of arrival of the radio signal at each of the telescopes are derived from accurate telescope models and GPS information. Final delay offsets, caused by differences in system hardware at the telescopes, need to be corrected by the correlator operator in a process called "clock searching".

Fine adjustments to the delays are made over a range of microseconds until interference fringes between telescopes are detected. Figure 12 is an example of a fringe plot of a baseline between two stations where the corrected delay has put the fringe perfectly in the middle of the search range.

An automated fringe detecting and correcting tool [7] was made especially for e-VLBI by the EXPReS software engineers to make the painstaking process of clock searching a lot faster and easier. But perhaps the best feature of this new tool is

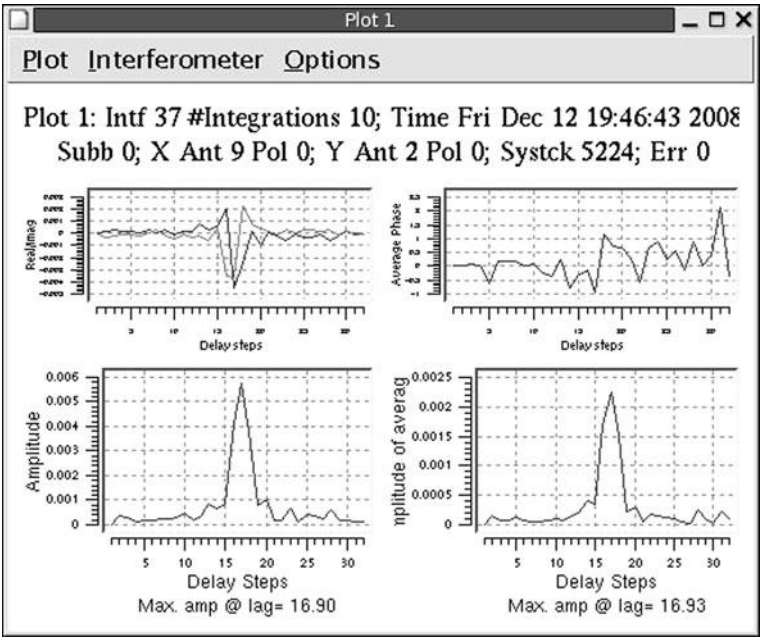


Fig. 12 Interference fringe between two telescopes

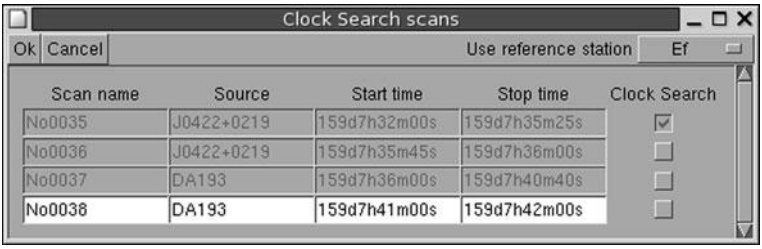


Fig. 13 Clock search interface

the fact that the corrected delays can be added to the correlation process in real time, without the need of stopping and restarting the correlator, as was the case with conventional VLBI. This is a very important added value to the clock searching process, because stopping a correlation run and starting it up again during e-VLBI can be time consuming, while precious telescope data is not being correlated and thus lost.

This automated clock searching tool proved to be so convenient that it is now used for disk-based VLBI as well. Figure 13 shows the interface of the clock search tool.

Now with all the delays corrected and the fringes to all stations centered, the observation's setup is complete. If all systems work properly and the networks

perform up to standard, the entire setup so far should not have lasted more than approximately half an hour. Usually the delays to all the telescopes are pretty well known and clock searching is just a matter of adjusting the timing so the fringes sit perfectly in the middle of the fringe window. If one or two of the stations' fringes are outside the search window, it might take a few iterations of adjusting the delays and scanning the data again. As long as the data is correct and the telescope signal is actually present, these additional iterations should not last more than maybe another half hour. However, if there is a problem with the network or with the telescope hardware, so the signal does not even make it to the correlator, it might take hours to get that particular station included. Sometimes the problem cannot be solved at all within the duration of the experiment and the telescope involved is a total loss for the observation.

3.10 Start of the Science Observation

The local correlator settings needed for clock searching usually differ a lot from the settings the PI needs for his astronomical science data. Therefore, in the Runjob window (Fig. 9) the correct settings for the number of lags, cross/no cross polarization and integration time are changed to values that suit the PI's needs best. With these new settings in place a new correlation run is started and the science part of the observation has begun.

The emphasis of the correlation process now lies on maintaining as stable a system as possible. Changes in delay setting are from now on no longer made. The major task of the correlator operator is now to monitor the data quality and to correct any system errors that may occur.

3.11 Data Quality Analysis and Feedback

Tool: Data Status Monitor

The Data Status Monitor [8, 9] (Fig. 14) was designed by the EXPReS group to keep a complete overview of all the aspects of an e-VLBI run. All there is to know about the correlation details, such as the data rates from each telescope, the correlation parameters, the duration of the scans, and even the correlated output data can be monitored with this tool. When one has to keep track of this many variables at the same time, it is practically a must to have a tool that presents all this information graphically in an organized way.

A very important aspect of the Data Status Monitor is the fact that all the information in this window is also available in real time on the JIVE web page, so the observers at the telescopes have immediate feedback on the performance of their systems. Giving as much feedback as possible to everyone involved in a major operation like an e-VLBI observation is of great value. Without this tool, the outside

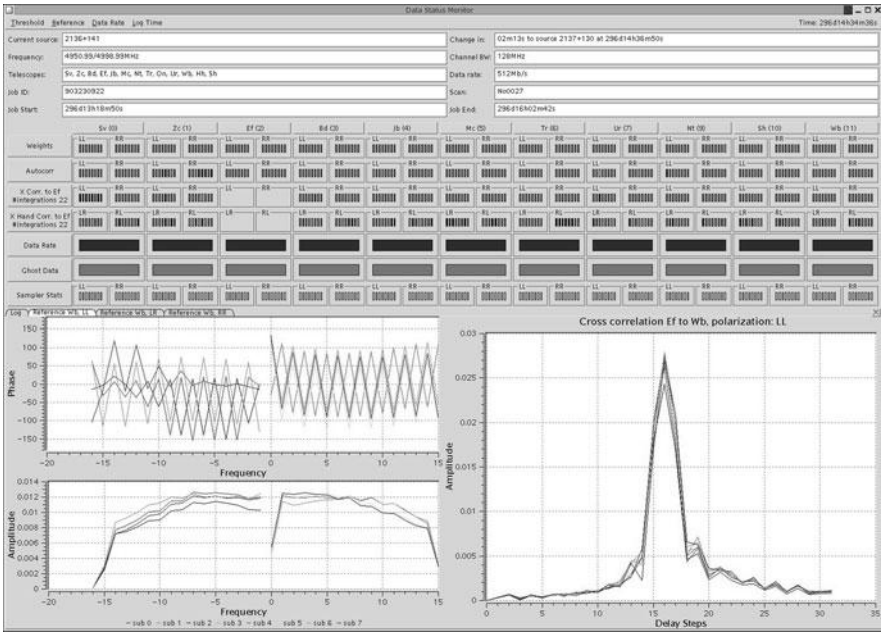


Fig. 14 Data status monitor: Error checking, diagnosis, and feedback utility

world has no idea what is going on at the correlator, or what the status of the current job is. The people at the telescopes are now much more involved with the e-observation. They can, for instance, see when the next source change is due, so when they can expect to see their telescope to start slewing, or whether they are missing data in one of their sub bands. These are all subjects that used to take a lot of communicating back and forth. Data Status Monitor provides all this information in one interactive feedback tool. Feedback is crucial.

Besides the Data Status Monitor the fringe plots are also available online in real time. Seeing a fringe is the final proof that that a telescope is working fine. If there is something wrong in the signal path from the source to the correlator, there will not be a fringe on that baseline. The lack of a fringe can be caused by numerous things, for instance a pointing error of the telescope, cross-connected wires, unsynchronized local oscillators, errors in the timing system, strong Radio Frequency Interference (RFI) [6], and many other things. In conventional VLBI these problems all need to be addressed at the correlator before the observation can be processed, usually weeks or months after the experiment is observed. If the error is irrecoverable then the data are lost. With e-VLBI we can quickly detect any error in the data and give immediate feedback to the observer at the telescope. This gives the observer the opportunity to solve the problem in an early stage of the observation, thus saving a lot of valuable data for the PI.

4 Most Frequent Problems

The most frequent cause, by far, of data loss from one or more stations is a timing problem in a part of the correlator hardware called the Station Unit [10]. There is one Station Unit connected to every Mark5, taking care of the data of one telescope. This timing problem, known as "diagonal weight", will randomly result in a slow decline in the data quality of the telescope involved, until it has reached a level at which the data have become useless.

During regular disk-based VLBI the solution to this problem is to stop the current run and to start up a new run, picking up the data from where it went wrong the first time. This stop-start procedure can easily take up to 30 min.. During e-VLBI this is not acceptable, because the data coming from the telescopes are being correlated in real-time, so for the time it takes to start up a new correlation run, the data from all the telescopes are lost. To prevent this from happening, the EXPReS software engineers developed a tool [11] that allows us to stop and restart just the one single telescope that is affected by the timing problem. The correlation run itself is not stopped and all the other, good working stations keep producing data. The moment the restarted station is back online, it is automatically entered back into the running job where it will join the other, still running, telescopes. Because of its ability to stop and restart one single telescope and to add it back into a running correlation job, this tool is an absolute life-safer for e-VLBI. Before we had this tool we sometimes lost unacceptable periods of data, simply because we had to restart the correlator whenever a station experienced this timing error.

The Station Units are one of the oldest parts of the correlator system; keeping them in good shape is becoming more and more difficult because of the lack of spare parts. There are other problems with the Station Units that constantly require a workaround, even up to the point were we sometimes have to make compromises concerning the data quality. Fortunately there is a solution at hand in the form of the successor to the Mark5, the Mark5B [12]. The functionality of the Station Unit is completely implemented in the Mark5B's hardware, making the entire Station Unit obsolete. However, before we can start using the Mark5Bs, some major changes to the Correlator Control Software are needed. The telescopes also need to upgrade some of their hardware before they can make the transition to Mark5B.

If we want to maintain the high level of data quality that we provide to the astronomers worldwide, we need to start using the Mark5Bs as soon as possible. Ninety percent of the time the operator has to intervene in the correlation process, the root of the problem lies with the Station Unit. The implementation of Mark5B should be on the top of the priority list, for both the stations and the correlator.

Another problem, about which we can do absolutely nothing, is that sometimes the network connection to a telescope goes down. We can see that the sending Mark5 at the station is streaming its data onto the network, but we cannot see any data coming into the Mark5 at the correlator. All we know is that somewhere along the way the light path must be interrupted. This is a very frustrating problem because the systems, both at the telescope and at the correlator, are working fine, but the data cannot reach the correlator. The operator has no means of fixing it, he does not

know if the network provider is aware of the problem or if anyone is working on a solution. All the operator can do is wait until the light path gets back up and all the time he has no idea how long it is going to take, nor if the connection will even be restored before the end of the observation. There are no means available to the operator to report performance metrics back to the network provider.

We have dedicated connections to some of our telescopes, see table 1. For telescopes that rely on routed connections we have had lengthy discussions on the amount of free space available for e-VLBI. In the case of congestion the e-VLBI system is capable of dynamically adjusting the amount of data send over the network.

5 Tools We Need

5.1 Network Monitoring Tool

At the moment, network quality is measured by several different groups at the same time. We assume network operators are able to monitor the status of fibers and that NRENs are aware of load across their systems. However, in our specific scenario, there is not an automated, programmatic system to report our network metrics back into the infrastructure.

Something which would be very useful in case of a network outage is a tool that allows us to investigate the network connection to the telescopes. It should be able to show us where the interruption in the network has occurred. It should also have a link to the provider, with which the user can report that the network to one of the telescope sites is down.

5.2 Internal Warning System

A tool that we definitely need is an audible warning system that will alert us when something in the correlation process is going wrong. In the evening and during the night the operator is the only person present at the correlator. He has to monitor the data quality and the behavior of the system at all times. After all, when the correlator software crashes, the data of all the telescopes are lost. Obviously, it is impossible to sit at a desk and stare at the screen for 8 h straight. So what we need is a piece of software that will produce a warning beep whenever there is something wrong with the data or the processor. In case the operator is not behind the correlator computer, a message should be send to the mobile phone that he carries around with him at all times. A similar system is used at the Westerbork observatory with great success.

Thinking further along the line of an automated warning system, if a program can detect errors in the data, besides warning the operator, it could also undertake the necessary steps to fix the problem, for instance, by restarting one of the telescopes. An automated response tool should be completely programmable by the operator.

This way the level of intervention by the software can be changed, depending on the nature of the observation.

6 Conclusion

High-bandwidth fiber-optic networks have provided the EVN data processor at JIVE with a real-time VLBI network, connecting telescopes in Europe, Asia and South America to the correlator in the Netherlands.

e-VLBI has become a reliable scientific instrument for conducting radio astronomy with real-time data processing at the correlator. With seven telescopes in Europe connected to the correlator at 1 Gbit/s each and two telescopes in China and South America at 512 Mbit/s each, e-VLBI can now compete with many conventional disk-based VLBI observations. Still there are EVN telescopes that do not have full e-VLBI capabilities yet. New telescopes currently being build also need to be taken up into the e-VLBI array. Expansion of network connectivity is therefore needed. To keep up with the demands of the international astronomical community, the bandwidth of the network connections needs to be increased in the future.

Due to the real-time nature of an e-VLBI experiment, the role of the correlator operator is crucial to the success of the observation.

The great benefit of doing e-VLBI is the ability to give real-time feedback to stations on the performance of their systems. Early detection of problems in the data, related to the telescope behavior, gives the station's observer the possibility to solve the problem before the science observation starts. Automated response tools that monitor the data quality and network performance are indispensable.

Network problems are an absolute showstopper for e-VLBI. A network outage to one of the telescopes, or even poor network performance, means total data loss for the telescope in question. A system for reporting back our network metrics to the provider must be developed.

This document gives a better understanding of the breadth of an e-VLBI observation. Lessons learned from our experiences can be applicable to other user or developers of online and distributed Grid based systems.

References

1. R.M. Campbell. So you want to do VLBI, November 1999. http://www.jive.nl/~campbell/jvs2.ps.gz (January 2009)
2. R.T. Schilizzi, W. Aldrich, B. Anderson, A. Bos, R.M. Campbell, J. Canaris, R. Cappallo, J.L. Casse, A. Cattani, J. Goodman, H.J. van Langevelde, A. Maccafferri, R. Millenaar, R.G. Noble, F. Olnon, S.M. Parsley, C. Phillips, S.V. Pogrebenko, D. Smythe, A. Szomoru, H. Verkouter, A.R. Whitney. The EVN-MarkIV VLBI Data Processor, Experimental Astronomy, 12(1), 49–67(19), 2001, DOI: 10.1023/A:1015706126556 (December 2008)
3. VLBI Standard Interface (VSI), August 2000. http://vlbi.org/vsi/index.html (January 2009)
4. Mark5 VLBI Data System. http://www.haystack.mit.edu/tech/vlbi/mark5/ (January 2009)
5. P. Boven. Full steam ahead – 1024 Mb/s data rate for the e-EVN First Draft, January 8, 2008. http://www.jive.nl/techinfo/evlbi/1024Mbps.pdf pdf (December 2008)

6. A.R. Thompson, J.M. Moran, G.W. Swenson, Jr. Interferometry and synthesis in radio astronomy, Wiley, New York, NY, 1986 (February 2009)
7. B. Eldering, On-the-fly clock searching, November 10, 2008. http://www.jive.nl/~jive_cc/sin/sin14.pdf (December 2008)
8. B. Eldering, Data status monitor tool, October 11, 2006. http://www.jive.nl/~jive_cc/sin/sin3.pdf (November 2008)
9. B. Eldering, Extensions to the data status monitor tool, July 21, 2009. http://www.jive.nl/~jive_cc/sin/sin18.pdf (July 2009)
10. B. Anderson. The EVN/Mark IV Station Unit Requirements, MarkIV Memo #140, Revision D, 930301. http://www.haystack.mit.edu/geo/mark4/memos/140.pdf (December 2008)
11. B. Eldering. Automatic diagonal weight detection, May 6, 2009. http://www.jive.nl/~jive_cc/sin/sin15.pdf (May 2008)
12. A.R. Whitney, R.J. Cappallo. Mark 5B design specifications, Mark5 Memo #19, 24 November 2004. http://www.haystack.mit.edu/tech/vlbi/mark5/mark5_memos/019.pdf (November 2008)

Observatory Middleware Framework (OMF)

Duane R. Edgington, Randal Butler, Terry Fleury, Kevin Gomes, John Graybeal, Robert Herlien, and Von Welch

Abstract Large observatory projects (such as the Large Synoptic Sky Telescope (LSST), the Ocean Observatories Initiative (OOI), the National Ecological Observatory Network (NEON), and the Water and Environmental Research System (WATERS)) are poised to provide independent, national-scale in-situ and remote sensing cyberinfrastructures to gather and publish "community"-sensed data and generate synthesized products for their respective research communities. However, because a common observatory management middleware does not yet exist, each is building its own customized mechanism to generate and publish both derived and raw data to its own constituents, resulting in inefficiency and unnecessary redundancy of effort, as well as proving problematic for the efficient aggregation of sensor data from different observatories. The Observatory Middleware Framework (OMF) presented here is a prototype of a generalized middleware framework intended to reduce duplication of functionality across observatories. OMF is currently being validated through a series of bench tests and through pilot implementations to be deployed on the Monterey Ocean Observing System (MOOS) and Monterey Accelerated Research System (MARS) observatories, culminating in a demonstration of a multi-observatory use case scenario. While our current efforts are in collaboration with the ocean research community, we look for opportunities to pilot test capabilities in other observatory domains.

1 Introduction

We are creating a prototype cyberinfrastructure (CI) in support of earth observatories, building on previous work at the National Center for Supercomputing Applications (NCSA), the Monterey Bay Aquarium Research Institute (MBARI), and the Scripps Institution of Oceanography (Scripps). Specifically we are researching alternative approaches that extend beyond a single physical observatory to

D.R. Edgington (✉)
Monterey Bay Aquarium Research Institute, 7700 Sandholdt Road, Moss Landing, CA, 95039, USA
e-mail: duane@mbari.org

F. Davoli et al. (eds.), *Remote Instrumentation Services on the e-Infrastructure*,
DOI 10.1007/978-1-4419-5574-6_16,

support multi-domain research, integrate existing sensor and instrument networks with a common instrument proxy, and support a set of security (authentication and authorization) capabilities critical for community-owned observatories.

Various scientific and engineering research communities have implemented data systems technologies of comparable functionality, but with differing message protocols and data formats. This creates difficulty in discovering, capturing and synthesizing data streams from multiple observatories. In addition, there is no general approach to publishing derived data back to the respective observatory communities.

Similarly each observatory often creates a security infrastructure unique to the requirements of that observatory. Such custom security solutions make it difficult to create or participate in a virtual observatory that consists of more than a single observatory implementation. In such circumstances the user is left to bridge security by managing multiple user accounts, at least one per observatory.

In recent years, several technologies have attempted to solve these problems, including a Grid-based approach of Service Oriented Architecture [1] and Web Services [2]. We have taken the next evolutionary step in the development of observatory middleware by introducing an Enterprise Service Bus messaging system like those routinely used in industry today. The benefits of message-based systems in support of observatory science have been documented by other projects including ROADNet [3] and SIAM [4]. ESBs are well suited to integrate these systems because of the performance and scalability characteristics of ESBs, and their ability to interconnect a variety of message systems, thereby easing the integration of legacy middleware.

To meet these requirements, we investigated existing technologies, including sensor, grid, and enterprise service bus middleware components, to support sensor access, control, and exploitation in a secure and reliable manner across heterogeneous observatories. For this project, we investigated two open source ESB implementations: Mule and Apache ServiceMix, and built a prototype based on Apache's ServiceMix ESB implementation. We focused on two functional areas within this framework to demonstrate the effectiveness of our proposed architecture: (1) Instrument Access and Management, and (2) Security (specifically access control).

2 Architecture

Our approach leverages an Enterprise Service Bus (ESB) architecture capable of integrating a wide variety of message-based technologies, a Security Proxy (SP) that uses X.509 credentials to sign and verify messages to and from the ESB, and an Instrument Proxy based on widely-accepted encoding and interface standards that has been designed to provide a common access to a number of different instrument management systems, including MBARI's Software Infrastructure and Application for MOOS (SIAM), Scripps' Real-Time Observatories, Applications, and Data management Network (ROADNet), and native (stand-alone) instruments.

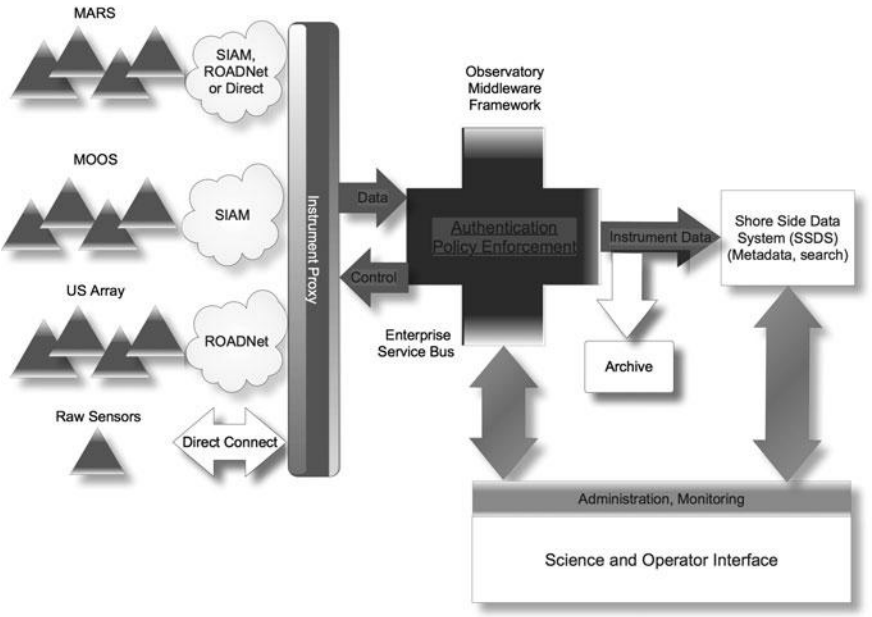


Fig. 1 Deployment diagram for our observatory middleware framework, interfacing with MARS, MOOS, US Array [5] or raw sensors, and using the shore side data system [19]

Figure 1 shows how the middleware fits into the observatory architecture with the Instrument Proxy providing common interfaces for sensors and controllers, and authentication, authorization and policy enforcement being embedded with the ESB.

As described above, our architecture and implementation draws from previous work demonstrating the benefits of a message-based system such as that found in ROADNet [3, 6, Fig. 2] and in industry, and takes the next evolutionary step with an Enterprise Service Bus (ESB) architecture. ESBs have been widely accepted in industry and proven to readily integrate web service, Grid, HTTP, Java Message Service, and other well-known message-based technologies. Within an ESB, point-to-point communication, where each of n components requires n–1 interfaces for full communication, are replaced by a bus solution, where each component requires a single interface to the bus for global communication. An ESB provides distributed messaging, routing, business process orchestration, reliability and security for the components. It also provides pluggable services which, because of the standard bus, can be provided by third parties and still interoperate reliably with the bus. ESBs also support loosely-coupled requests found in service oriented architecture, and provide the infrastructure for an Event Driven Architecture [7].

The resulting cyberinfrastructure implementation, known simply as the Observatory Middleware Framework (OMF), is being validated through a series of bench tests, and through pilot implementations that will be deployed on the

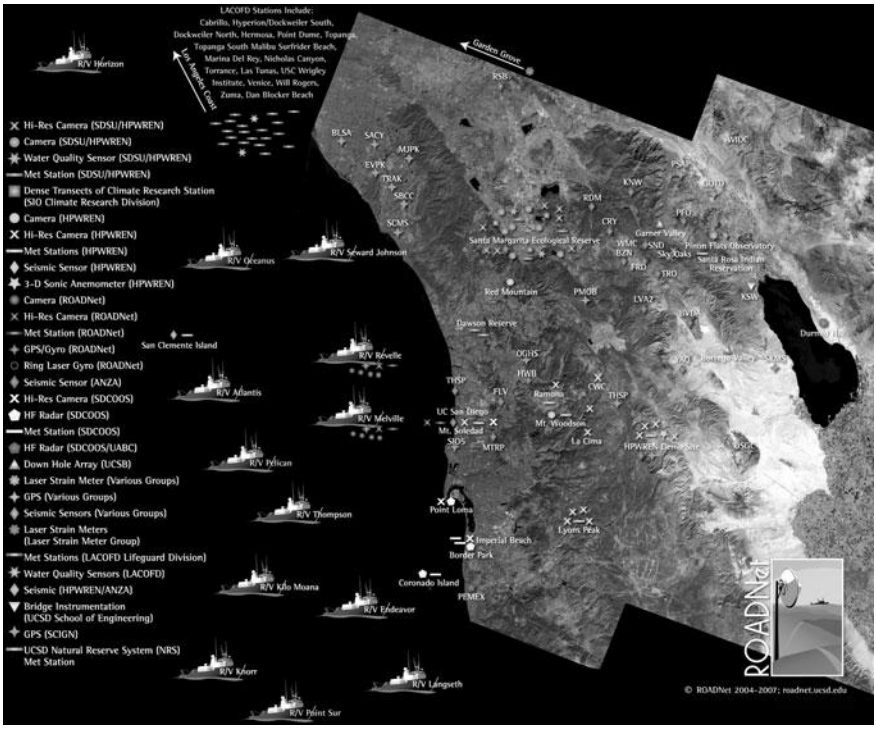


Fig. 2 ROADnet sensor map, illustrating distribution and types of land and sea environmental sensors

Monterey Ocean Observing System (MOOS) [8, 4] and Monterey Accelerated Research System (MARS) [9, Fig. 3] observatories, culminating in a demonstration of a multi-observatory scenario. We are working closely with the ocean research community, as their observatory architecture is one of the most mature, but we are targeting OMF for broader adoption. We welcome collaborative opportunities to pilot these capabilities in other observatory domains.

3 An Observatory Framework for Instrument Management

A general-purpose observatory framework must support a wide variety of instrumentation, enabling devices to be discovered, controlled, and seen by human and automated systems. This requires consistent overarching protocols to enable instrument control and message output, as well as to validate the authority of a user or system to control the instrument and see its data. These messaging, security, and policy enforcement capabilities have been architected into the Observatory Middleware Framework in a way that scales to future instrument installations, allowing the diversity of new instrumentation to be incorporated into OMF systems.

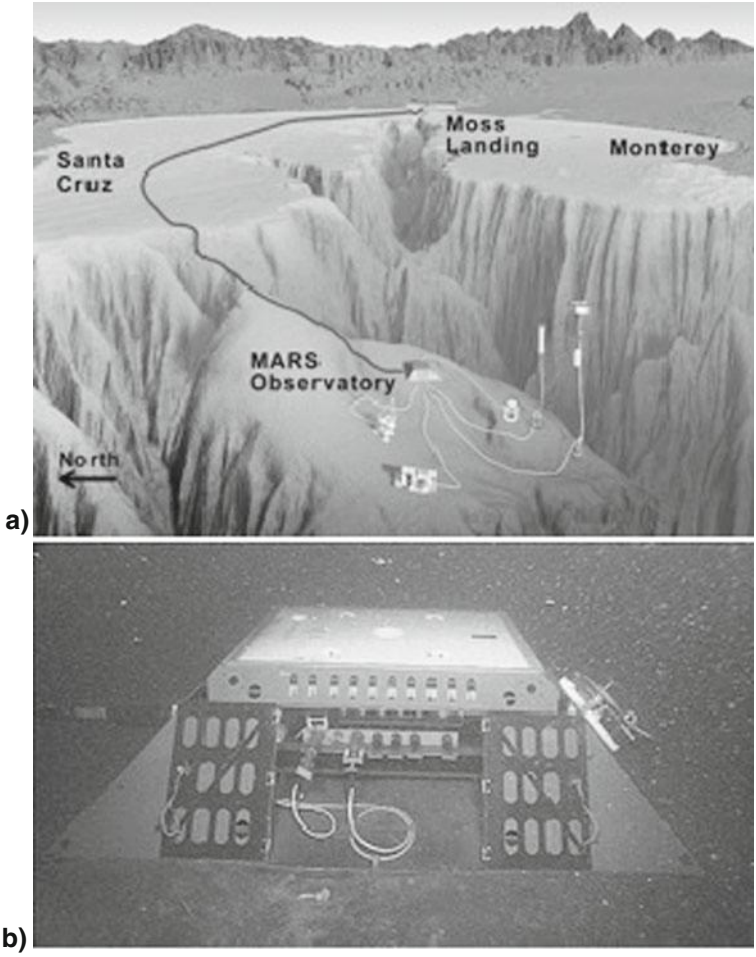


Fig. 3 MARS (Monterey Accelerated Research System). (**a**) The main "science node" of the MARS observatory has eight "ports," each of which can supply data and power connections for a variety of scientific instruments. (**b**) The hub sits 891 m below the surface of Monterey Bay off the coast of California, USA, connected to shore via a 52-km undersea cable that carries data and power. *Drawing: David Fierstein © 2005 MBARI*

Many observatories enable common delivery of data from multiple instruments, and quite a few have established common command protocols that all their instruments follow, either by design or through adapters. Such observatories invariably have one or more of the following characteristics: the instruments and platforms are largely uniform (ARGO [10] is a ready oceanographic example; the fundamental instrument unit for NEON [11] is an ecological example); they have relatively few instruments (or adapters for them) that are custom-developed to support a specific protocol (many astronomical observatories follow this model); or a small subset of

the available data is encoded using a relatively narrow content standard and reporting protocol (for example, the National Data Buoy Center (NDBC) [12] reporting of data from oceanographic buoys).

With widespread adoption of data reporting and aggregation protocols such as OPeNDAP [13], or more recently standardized web services like those produced by OGC [14], it has become more feasible to consider data and control integration across a wider range of data sources, data formats, and control interfaces. To date, these solutions have been quite limited; only certain regularized interfaces are supported, and there are no overarching ("system of systems") grid-style architectures that tie together multiple service providers gracefully. Furthermore, existing architectures have not provided for security and policy enforcement of resources – whether limiting access to particular data sets, or constraining the configuration and usage of data providers and services, such as instruments and models – that will be needed to flexibly control multiple observatories and observatory types.

The Observatory Middleware Framework addresses all these instrument issues through the use of standard interface protocols, a common set of core capabilities, and an Instrument Proxy to adopt new instruments to the existing framework. The Instrument Proxy sits between the ESB and the managed instruments, and provides a common instrument interface for command and control. We use the Sensor Modeling Language (SensorML) and the Observations and Measurements (O&M) encoding standards, as well as the Sensor Observation Service (SOS), and Sensor Planning Service (SPS) interface standards. Those specifications, coupled with the MBARI-developed Software Infrastructure and Application for MOOS (SIAM) system, provide a basis for the Instrument Proxy. We have documented a set of "least common denominator" services and metadata to support a collection of basic instrument commands. Our prototype will support common instrument access to SIAM managed instruments as well as native instruments. In the latter part of our project we plan to demonstrate support for the Scripps-developed Real-Time Observatories, Applications, and Data Management Network (ROADNet) instrument management system as well.

4 Security Model

The goal of our security model is to allow the ESB to enforce access control on messages that it transports. By controlling who can send what form of messages to an instrument, the ESB can effectively control who can manage that instrument. To manage the access to data published by an instrument, the ESB controls access privileges for those publication messages.

To effect this message control, we implemented an entity in the ESB, which we titled the Authorization Service Unit (ASU). Routing in the ESB is configured such that any messages to which access control should be applied are routed through the ASU, which inspects the message sender and recipient as well as any relevant contents, and then applies access control policy. The ASU is, in security terms, a combined policy decision point and policy enforcement point.

Initially we thought we could achieve this access control solely through the ASU alone. The problem with this approach is that the instrument proxy does not connect directly to the ESB, but instead through some message transport system. This transport system was ActiveMQ in our prototype, but could conceivably be any technology the ESB is capable of decoding (e.g. SMTP, HTTP). This flexibility, which is one of the strengths of an ESB, presents a challenge from a security standpoint since we cannot make assumptions about what properties the systems provides in terms of resistance to modification, insertion, etc. A transport system that allowed message insertion could allow a malicious (or even accidental) insertion of control messages that bypass the ESB and the ASU.

To address this challenge, we decided to implement message-level security between the ASU and the instrument. Modifying every instrument to support this security would be infeasible, so using the same approach as the instrument proxy, we implemented a Security Proxy which sits between the instrument and the transport system. The Security Proxy signs messages, to allow for their validation by the ASU, and verifies messages signed by the ASU, serving to prevent any modification or insertion. (With further enhancements we can prevent replay attacks; currently these are a security hole.) These messages could also be encrypted to provide confidentiality, but it's not clear that this is a requirement of our target communities and worth the performance impact.

The specific implementation in our prototype uses SOAP messages signed with X.509 credentials. The initial ASU will implement simple access control policies based on sender and recipient.

The following list enumerates the sequence of events under our envisioned security model:

1. A researcher uses a web portal to send a request to remotely modify the data collection process from a specific instrument in the offshore instrument network.
2. The Security Proxy signs the outgoing modification request and passes it through to the Enterprise Service Bus (ESB) via the Message Broker.
3. ActiveMQ, serving as a Message Broker, delivers the message to the Enterprise Service Bus.
4. The Authorization Service Unit verifies the message signature, applies policy, authorizes the message, and resigns it with its own key. The Enterprise Service Bus then routes the message to its intended destination (in this case, the networked instrument).
5. ActiveMQ, serving as a Message Broker, delivers the message to the networked instrument.
6. The Security Proxy verifies incoming messages to ensure that the Authorization Service Unit in the Enterprise Service Bus has processed them.
7. The Instrument Proxy converts the message (as needed) to the syntax and commands specific to the instrument for which it is intended.
8. After reaching the deployed instrument network, the message is relayed to the intended instrument.

9. The instrument then sends a confirmation or other response, which is returned to the researcher via the same logical route as used by the original request. The message destination has a unique identity in OMF, as encoded in the original request and authenticated by the Security Proxy.
10. The response is returned to the researcher by the web portal. Additional diagnostic information, accumulated as the communication passes through the OMF and instrument network, is also made available to the user and system operators as appropriate given their respective authorizations.

5 Discussion

Several major projects, including OOI [19], WATERS [16], NEON [11], and Integrated Ocean Observing System (IOOS) [17], are planning the deployment and integration of large and diverse instrument networks. The Linked Environments for Atmospheric Discovery (LEAD) [18] project also demonstrates the intention of the meteorological community to engage a direct real-time observe-and-response control loop with deployed instrumentation. The Observatory Middleware Framework described here would apply to all of these systems, providing a data, process and control network infrastructure for each system to achieve key observatory capabilities. In addition, this framework would enable data and control interoperability *among* these observatory systems, without requiring them to have a common technology or semantic infrastructure. Our strategy employing an ESB links multiple observatories with a single interoperable design.

Acknowledgement The National Science Foundation funds the OMF project (award #0721617), under the Office of Cyberinfrastructure, Software Development for Cyberinfrastructure, National Science Foundation Middleware Initiative.

References

1. T. Erl. Service-oriented architecture: Concepts, technology, and design. Prentice-Hall, ISBN-10: 0-13-185858-0; Published August 2, 2005
2. Web Services: http://www.w3.org/2002/ws (2002–2010)
3. T. Hansen, S. Tilak, S. Foley, K. Lindquist, F. Vernon, and J. Orcutt. ROADNet: A network of sensornets. First IEEE International Workshop on Practical Issues in Building Sensor Network Applications, Tampa, FL, November 2006
4. T.C O'Reilly, K. Headley, J. Graybeal, K.J. Gomes, D.R. Edgington, K.A. Salamy, D. Davis, and A. Chase, MBARI technology for self-configuring interoperable ocean observatories, MTS/IEEE Oceans 2006 Conference Proceedings, Boston, MA, IEEE Press, September 2006 DOI: 10.1109/OCEANS.2006.306893
5. US Array: http://www.earthscope.org/observatories/usarray (2009)
6. ROADNet: http://roadnet.ucsd.edu/ (2010)
7. Michelson, B. Event-Driven Architecture Overview: Event-Driven SOA is Just Part of the EDA Story. Patricia Seybold Group, February 2006. http://dx.doi.org/10.1571/bda2-2-06cc
8. MOOS: http://www.mbari.org/moos/ (2010)
9. MARS: http://www.mbari.org/mars/ (2010)
10. ARGO: http://www-argo.ucsd.edu/ (2010)

11. NEON: http://www.neoninc.org/ (2010)
12. NDBC (National Data Buoy Center): http://www.ndbc.noaa.gov/ (2010)
13. OPeNDAP: http://www.opendap.org/ (2010)
14. OGC: http://www.opengeospatial.org/ (1994–2010)
15. OOI:http://www.oceanleadership.org/programs-and-partnerships/ocean-observing/ooi (2010)
16. WATERS: http://www.watersnet.org/index.html (2010)
17. IOOS: http://ioos.noaa.gov/ (2010)
18. LEAD: https://portal.leadproject.org/gridsphere/gridsphere (2006–2010)
19. J. Graybeal, K. Gomes, M. McCann, B. Schlining, R. Schramm, D. Wilkin. MBARI's operational, extensible data management for ocean observatories. In: The Third International Workshop of Scientific Use of Submarine Cables and Related Technologies, 25–27 June, 2003,Tokyo, pp. 288–292, 2003 DOI:10.1109/ssc.2003.1224165

Analysis and Optimization of Performance Characteristics for MPI Parallel Scientific Applications on the Grid (A Case Study for the OPATM-BFM Simulation Application)

A. Cheptsov, B. Koller, S. Salon, P. Lazzari, and J. Gracia

Abstract Over the last years, Grid computing has become a very important research area. The Grid allows the parallel execution of scientific applications in a heterogeneous infrastructure of geographically distributed resources. Parallel applications can foremost benefit from a Grid infrastructure in terms of performance and scalability improvement. However, performance expectations from porting an application to the Grid are considerably limited due to several factors, bottlenecks in the implementation of communication patterns are in the back of. Based on the analysis of the OPATM-BFM oceanographic application, we elaborate the strategy of the communication-intensive parallel applications analysis. This allowed us to identify several optimization proposals for the current realization of the communication pattern and improve the performance and scalability of the OPATM-BFM. As the suggested improvements are quite generic, they can be potentially useful for other parallel scientific applications.

1 Introduction

Operational oceanography is a branch of oceanography devoted to monitor, analyse and predict the state of the marine resources, as well as the sustainable development of coastal areas. OPATM-BFM is a MPI-parallel physical-biogeochemical simulation model [1] that has been implemented in the frame of the MERSEA project [2] and practically applied for short-term forecasts of key biogeochemical variables (e.g. chlorophyll, salinity and other) for a wide range of coastal areas, among others Mediterranean sea.

The efficient usage of high-performance resources (like an IBM SP5 machine, installed in CINECA [3] and currently used for the main production cycle), is a major point in reaching high productivity of the OPATM-BFM for tasks of real complexity. Whereas the application is expected to deliver additional products and information for different time scales as well as for climatic scenario analyses

A. Cheptsov (✉)
High-Performance Computing Center, University of Stuttgart, Stuttgart, Germany
e-mail: cheptsov@hlrs.de

F. Davoli et al. (eds.), *Remote Instrumentation Services on the e-Infrastructure*,
DOI 10.1007/978-1-4419-5574-6_17,

(multi-decadal period of integration), the resources, provided by dedicative high-performance computers, are limited in the number of computing nodes available for the application. The storage capabilities of dedicated systems are limited as well. This constitutes the most considerable limitation in the application scalability and usability for both long-term forecasts and analysis of eco-systems of higher complexity. However, this limitation can be avoided by porting the application to the Grid environment.

In recent years, the progress of the high-performance computing has enabled the deployment of large-scale e-Infrastructure like those promoted by EGEE [4] and DEISA [5] projects in the European Research Area. Focusing on the easier integration, security and QoS aspects of computation and storage resource sharing, e-Infrastructures ensure optimal resource exploitation by applications. Along with leveraging grid computing and storage capabilities, scientific applications can benefit from porting to an e-Infrastructure by using complete IT solutions provided by Grid middleware architectures for the integration of application with remote instrumentation (among others, GRIDCC project [6]), support of application interactivity (e.g. Int.EU.Grid project [7]) etc.

The experience of porting applications from different fields of science to the Grid [8] has encouraged us to examine the suitability of e-Infrastructures for improving the OPATM-BFM performance characteristics. This is being done relying on the Grid e-Infrastructure, set up by the DORII project [9]. This chapter presents the first results of research activities devoted to the adaptation of the OPATM-BFM for an efficient usage in modern Grid-based environments. For the improvement of application performance on standard generic clusters of SMP nodes such results are important as well.

2 The Current Realisation

The core of the OPATM-BFM makes a 3D Ocean General Circulation modelling System (OPA-based transport model [10]) up, coupled off-line with the BFM chemical reactor (with the 1/8° horizontal resolution and 72 vertical levels), developed at Istituto Nazionale di Oceanografia e di Geofisica Sperimentale (OGS). The adoption is done through the hard-coding, that rules the possibility to use newer versions of the third-party software (mainly with regard to the OPA part) practically out.

OPATM-BFM is parallelized using the domain decomposition over longitudinal elements (Fig. 1), that enables using massive-parallel computing resources for the application execution. The number of domains corresponds to the number of computing nodes, the application is running on. The consistence of the computation by the parallelization and domain decomposition is ensured by the inter-domain communication pattern that enables passing data cells resided on the domain bounds needed for the computation inside domains.

The current realization of the communication pattern is purely based on the Message-Passing Interface (MPI) [11] and implemented by means of single point-to-point operations. Cumulatively, more than 250,000 messages are transmitted on

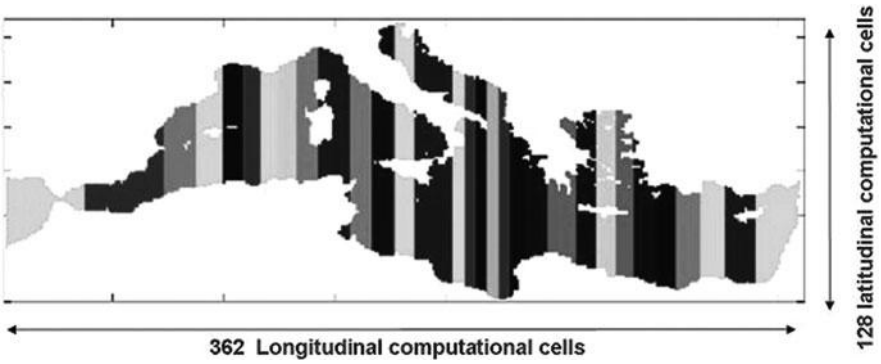


Fig. 1 The domain decomposition used in the OPATM-BFM

each step of the numerical solution in the current realisation. The analysis duration for a standard short-term forecast (17 days) on a generic cluster Intel Xeon 5150 processors is approximately 8 h.

3 The Analysis Strategy and Experiment Set up

The OPATM-BFM investigation was performed in the framework of the DORII project [9] which aims at deployment of a Grid e-Infrastructure for e-Science communities, focusing among others on environmental applications. The OPATM-BFM was implemented with the Open MPI library which is a production quality MPI-2 standard's implementation, developed by HLRS in cooperation with a consortium of research and industrial organizations [12].

The application was analyzed for a standard use case providing several types of input data that differ in complexity and duration depending on the corresponding simulation. Figure 2 shows the analysis scheme of the application investigation.

The analysis of internal message-passing communication patterns is based on the profile collected during the application execution. The collection has been performed by means of instrumentation tools and libraries which are a part of the instrumentation framework offered by the DORII project, alongside with other services. Based on the trace files, a communication profile can be analyzed by means of the visualization tools, also provided by the framework.

Due to the long duration of the execution (several hours) for a standard OPATM-BFM use case (namely, 816 steps of the main simulation – corresponds to 17 days of simulated ecosystem behavior) the size of recorded trace data increases accordingly.

However, as a standard production run has many iterations and each iteration performs many communication routine calls, the trace files get very large (up to tens of gigabytes for the analyzed use case). Therefore, the iteratively repeated regions can be profiled for a limited number of iterations that are representative of the generic pattern communication of a longer run. The initialization and finalization of the simulation are profiled as usual. This can be done by means of event filtering in the

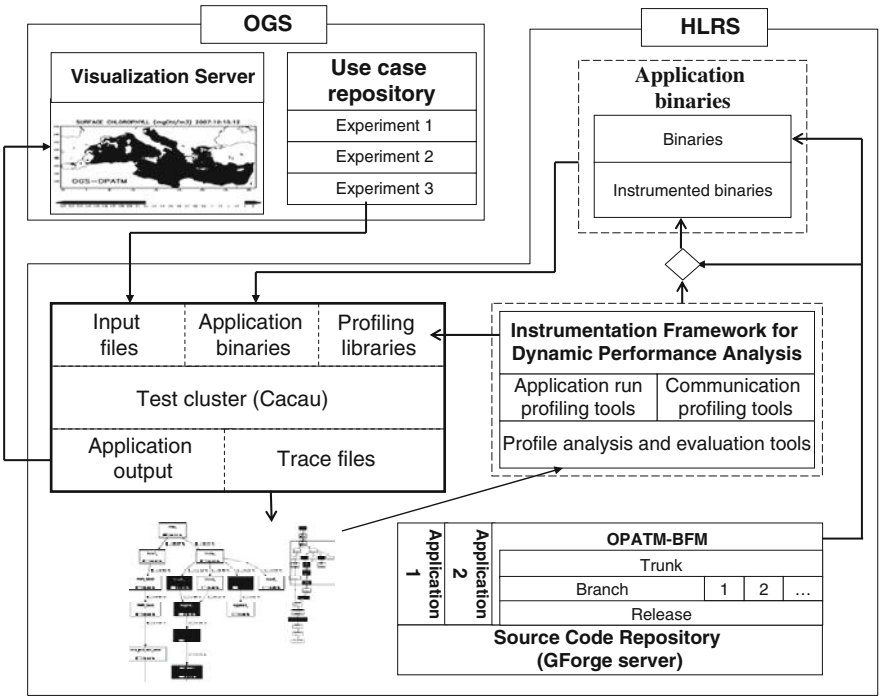


Fig. 2 The analysis scheme

defined regions of the execution or launching the application for special use cases with a limited number of iterations. The second approach is preferable because it allows us to reduce the time needed for launching the application in the test mode.

In order to proceed with the communication analysis efficiently, the phases of the application execution are to be identified. The localization of the most computation- and communication-intensive phases without the help of profiling tools is a non-trivial and quite complicated process that requires deep understanding of the application source code as well as the model basics. However, a so called application call graph from a profiling tool is sufficient for a basic understanding of dependencies between the regions of the application as well as time characteristics of the communication events in those regions. A fragment of the application call graph for the most important execution phases is presented in Fig. 3.

Such a run profile analysis is an excellent starting point for further investigation of the message-passing communication in the parallel application. This can be done by performing profiling MPI operations only in the most significant application execution regions.

4 Analysis Results

This section gives an overview of the main characteristics of the application run profile. The most communication- and computation-intensive phases of the application execution are identified for both test (3 steps of numerical solution) and standard

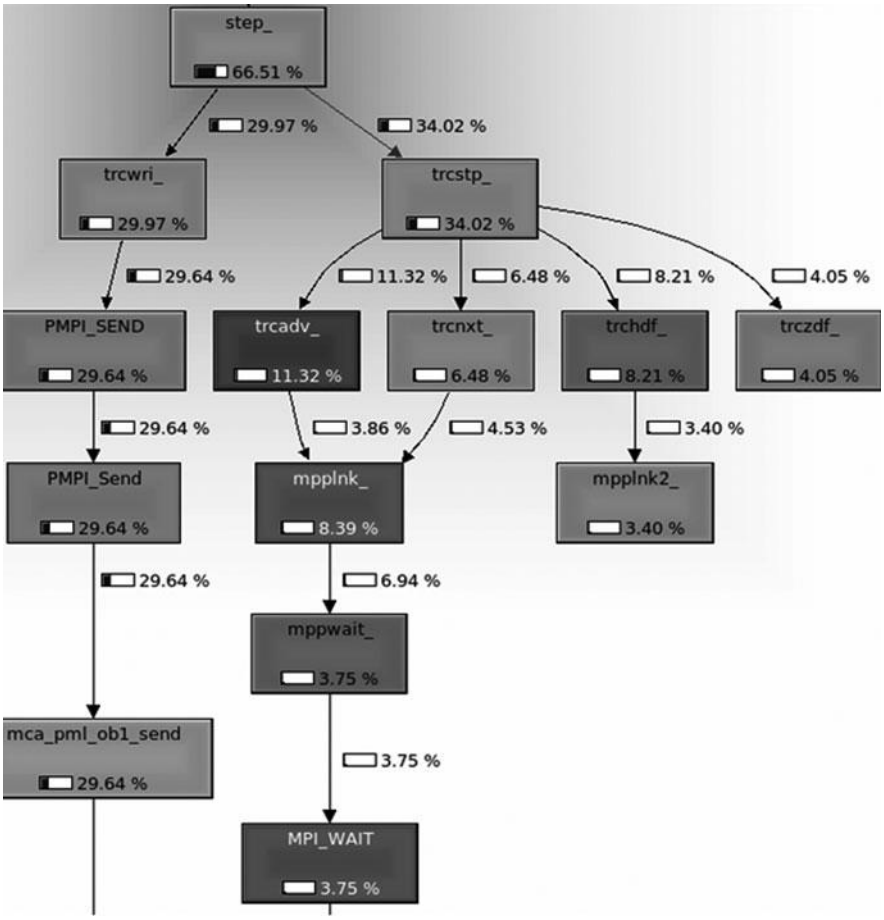


Fig. 3 A fragment of the application call graph (obtained with KCacheGrind tool of the Valgrind tool suite included to the DORII performance analysis framework)

(816 and more steps) use cases. The measurements and analysis results are presented as well.

4.1 Analysis of the Application Run Profile

For application run profiling we used tools from the instrumentation framework (e.g. the Valgrind tool suite [13]). For the investigation of the message-passing communication pattern we used the Paraver tool, developed in the Barcelona Supercomputing Center [14]. Assuming that the communication mechanism implemented in the main simulation step routine does not depend on iterations, we were able to limit the number of iterations that are profiled in the main simulation routine. For this purpose a special test use case which required only 3 steps of the main simulation was specified. That corresponds to 1.5 h real time of the ecosystem evolution. Main results of profiling for the test use case are collected in Table 1. The timing characteristics are

Table 1 Time distribution among the main phases of the execution (for the test use case)

Phases of the execution	Operations performed	# iterations, real case	# iterations, test case	Computation		MPI-calls		Total	
				Durat. [s]	Percent [%]	Durat. [s]	Percent [%]	Durat. [s]	Percent [%]
1. Initialization, Input	Loading of input data	1	**1**	939	99	6	<1	945	80
2. Main simulation loop	Internal MPI calls, halo cells exchange	816	**3**	3	38	5	**62**	8	2
3. Data storage	Storing of output and restarting data on disc, internal MPI communication	17	**1**	7	3	204	**97**	213	18
Total time				949	80	226	20	1,175	~100

followed by results for the application scalability acquired by launching the application on a larger number of nodes (Table 2). It is important to emphasize that the time distribution of the different phases compared to the total run time differs hugely for the test use case and for a real long-term simulation that requires a larger number of steps of the numerical solution. This is due to the fact that only the iterative part changes but not the initialization and finalization parts. The application scalability for running on different number of nodes is also changing accordingly to the size of the use case. Nevertheless, the internal characteristics of the iterative phase are iteration-independent and valid not only for the test use case but for all use cases.

While the most considerable part of the application is executed by operations of data input from the disk storage for the test use case (80% of the application execution time), the iteratively launched main simulation routine that implements a numerical solver will become a dominating part of the execution (up to 90% of the application execution) for a real production cycle use case that requires a big number of steps of numerical solution. For example, for a 2 weeks prediction 816 steps of numerical solution are required (see Table 1). Being launched only each 48th step of the numerical solution and at the end of the application execution, disk storage and data output operations are also an important point of the application performance optimization for both short-term and long-term forecasts.

4.2 The Message-Passing Communication Profile

The application is parallelized using the domain decomposition method. This required a message-passing mechanism for the data exchange among domains. This mechanism is implemented in the main step simulation routine (exchange of boundary elements of neighboring domains is performed at each simulation step) and in the output routine (*trcwri*) which performs the dataset storage in the NetCDF format

In the phase where the data storage and output operations are used to save restart data, the blocking point-to-point communication is used for data gathering in the root process. It is performed by means of 612 MPI_Send calls that are executed by each of the non-root processes and a corresponding number of MPI_Recv calls in the root process (all-to-one communication pattern). Cumulatively for the analyzed region 18,972 data messages are transmitted by means of MPI. The size of the transmitted messages varies from 4 kB (12,648 messages) up to 1.125 MB (204 messages).

However, the most MPI calls occur in the main step simulation routine. The frequency of message transmission in this phase of the application execution is very high (for the investigated use case totally 252,960 messages are being transmitted through MPI in the main step phase). Due to these factors the main simulation step routine became the most important study point in our further investigations.

The exchange of data fields stored in different domains is realized by means of non-blocking point-to-point MPI operations (MPI_Isend, MPI_Irecv, MPI_Wait) in routines responsible for the simulation of the 3D advection scheme (trcadv, 816/408 MPI calls used in each non-boundary/boundary domain), horizontal diffusion

Table 2 Scalability characteristics of the application (for the test use case)

Phases of execution	32 nodes, with one process per node			64 nodes, with one process per node			Scalability coefficient, t_{64}/t_{32}		
	Comput. duration [s]	MPI calls duration [s]	Total time [s]	Comput. duration [s]	MPI calls duration [s]	Total time [s]	Comput.	MPI calls	Total
1. Initialization, Input	939	6	945	2,238	6	2,244	2,4	1	2,4
2. Main simulation loop	3	5	8	2	5	7	0,7	1	0,9
3. Data storage	7	204	213	3	170	173	0,4	1,25	0,8
Total	949	226	1,175	2,243	181	2,424	2,4	0,8	2

(trchdf, 7242/3621 MPI calls accordingly) and time integration (trcnxt, 102/51 MPI calls accordingly).

The size of a standard message is 72 kB for the advection and time integration routines (a total of 1821 MB and 227.6 MB of data are transmitted respectively) and 1 kB for the horizontal diffusion routine (224.5 MB). Hence, the total amount of transmitted data for one execution step amounts to 2273.1 MB even for this small test case.

Therefore, the application should be classified as communication-intensive. The optimization of the currently implemented communication pattern together with I/O pattern improvement will be important for the further application performance and scalability improvement

5 Optimization Proposals and Improvement Evaluation

This section describes several optimization proposals for the current realization of application communication patterns as well as preliminary evaluation results of their impact on the application performance and scalability. Estimation of the performance improvement expectation for the long-term forecast is given as well.

The message transmission mechanism that is currently used for the inter-domain communication (blocking MPI calls) is uniform – the order, structure and type of separately transmitted data do not change within the execution in each simulation step. All the state variables are independently communicated each using separate MPI communication calls. Due to this property messages can be rearranged into segments transmitting more than one variable per MPI call. The current implementation foresees a transmission of only one variable per MPI operation (in the proposed implementation that would be the lower-bound segment size). Encapsulation of additional data into a segment enlarges the segment size and reduces the total number of MPI calls for data transmission. The highest data encapsulation level is reached by packing all messages into one single segment (that would be the upper-bound segment size). The effect is even larger for a network with a high latency and low bandwidth. However, a procedure of packing/unpacking messages into/from a segment can require an additional computational overhead reducing fractionally the performance effect on the communication.

Therefore, a detailed study of the optimal number of segments (varying the segment size from the lower-bound to the upper-bound size) is required for both homogeneous and heterogeneous types of message encapsulation. In case of the heterogeneous encapsulation an additional evaluation of the optimal segment size is necessary. For this purpose we have modified one of the application routines where the inter-domain communication is implemented (the time integration routine *trcnxt*) in order to allow the size of a segment to be explicitly specified by a user or an automatic tuning procedure. The first investigations for the current test configuration have proved that the data encapsulation influences positively the overall performance. The highest impact on reducing the communication time was reached

using the segments with the upper-bound size (in the tested time integration routine the communication time was reduced by 50%). The message encapsulation mechanism will also be implemented for other application routines which perform the inter-domain communication (described in the previous section).

The data encapsulation can be also beneficial in terms of the application performance for the serialization of datasets from multiple processes to the root process that is required for the sequential data storage to a NetCDF file. This is implemented through the all-to-root communication pattern (currently implemented by means of blocking point-to-point MPI operations). However, for this pattern we have concentrated on using collectives instead of point-to-point operations.

As the practical experience showed for the large-scale target platforms [15], for the observed type of communication the usage of collective gather and reduce MPI operations is an advantage. We developed a new message transferring mechanism that is based on collective MPI operations and provides encapsulation of data arrays into segments similarly to the approach described above. For the test configuration the usage of collective MPI operations reduces the communication time by 50%. However, it is important to note that the optimal number of segments and the segment size for both analyzed types of MPI operations can differ for other configurations, target platforms, number of launched processes, implementations of the MPI library etc. and this will be also an important point of our future research.

Whereas file I/O operations are dominating for the application execution for a small number of simulation steps (3 steps for the test case), the overall performance improvement due to optimization of the MPI communication becomes significant only for a long-term simulation (816 steps for the real case). As Fig. 4 shows, the total amount of realized optimizations allowed us to reduce the duration of the application execution for the real case by up to 5% (from initially measured 309 min down to 293 min). Furthermore, the optimization for the increasing number of nodes

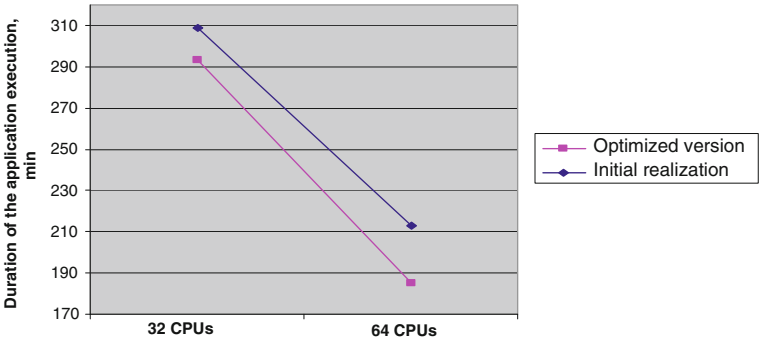


Fig. 4 The time characteristics of the application execution for the real case (816 steps of the numerical solution)

from 32 up to 64 grew up to 15% (from 213 min down to only 185 min). The application scalability grew accordingly from 145% up to 158%

6 Further Improvement and Future Research

While parallel file systems provide distribution of application input and output data on different storage elements of the Grid, any sequential implementation of I/O poses an obstacle for obtaining high application performance and scalability. On the contrary, parallel I/O provides many opportunities for an optimized access to underlying file systems. For applications implemented with MPI the parallel I/O can be accessed through MPI-IO, the I/O mechanism specified in the MPI-2 standard. Usage of I/O libraries built on top of MPI-IO for retrieving and storing data is therefore a good solution for the I/O optimization on a parallel file system. Parallel NetCDF (PNetCDF) follows this idea and is a library providing high-performance access to data arrays stored in the NetCDF format. As indicated in [16], implementation of the application I/O by means of parallel operations is preferable compared with sequential operations and results in the significant improvement of I/O performance. OPATM-BFM will take advantage of PNetCDF using a number of optimization mechanisms provided by the underlying hardware-specific modules for MPI-IO. We are going to investigate the performance impact of implementing the OPATM-BFM's I/O by means of the parallel NetCDF library.

With regard to the communication pattern, there are some strong arguments in favour of a hybrid model which tend to underline the assumption that OpenMP/MPI realization should lead to improved parallel efficiency of the OPATM-BFM as compared to current pure MPI implementation. This is especially important from the perspective of the application evolution towards running in high-performance grid environments, where the price/performance sweet spot is settled at a point. The experience for other applications from different problem areas has shown, that to the expected benefits of the hybrid parallelization can be referred: additional level of parallelism with inherent performance and scalability optimization, improved load balancing, reduced memory consumption and many others [17]. Development of the hybrid OpenMP/MPI communication pattern will be performed in the collaboration with the Earth Science Department of the Barcelona Supercomputing Center in the frame of the HPC-Europa project. The Paraver performance analysis tool will be very beneficial for our investigations, as it fully supports hybrid OpenMP+MPI communication models.

7 Conclusions

The article presents a technique of application performance and scalability characteristics analysis. We used the technique to analyze in details the communication

patterns of the OPATM-BFM. This allowed us to identify the shortcomings of the application communication and I/O patterns and elaborate several optimization proposals, minimizing these shortcomings and dramatically improving the application performance on a generic cluster of workstations. On the other hand, the described improvements will also allow us to increase the application performance on the IBM SP-5 machine, currently used for running the application's main operation procedure.

The presented technique should also support the developers of the OPATM-BFM in further understanding the communication patterns improving the load balance for different use cases, defining bottlenecks as well as working out solutions on how to resolve the shortcomings and maximize the performance and scalability of the application. Moreover, the results for the OPATM-BFM application will be important for other scientific parallel applications that implement similar communication pattern.

References

1. A. Crise, P. Lazzari, S. Salon, and A. Teruzzi. MERSEA deliverable D11.2.1.3 – Final report on the BFM OGS-OPA Transport module, 21 pp., 2008
2. See the web page of the Marine Environment and Security for the European Area (MERSEA) European Integrated project – http://www.mersea.eu.org, accessed 1.06.2010
3. See the description of the IBM SP5 machine on he CINECA's web page – https://hpc.cineca.it/docs/user-guide-zwiki/SP5UserGuide, accessed 1.06.2010
4. Enabling Grids for E-Science (EGEE) project website, http://www.eu-egee.org/, accessed 1.06.2010
5. Distributed European Infrastructure for Supercomputing Applications (DEISA) project website, http://www.deisa.eu/, accessed 1.06.2010
6. Website of the GRIDCC project, http://www.gridcc.org, accessed 1.06.2010
7. Website of the Int.EU.Grid project, http://www.i2g.eu, accessed 1.06.2010
8. B. Simo, O. Habala, E. Gatial, and L. Hluchy. Leveraging interactivity and MPI for environmental applications. Computing and Informatics, 27, 271–284, 2008, accessed 1.06.2010
9. See the web page of the DORII project – http://www.dorii.org, accessed 1.06.2010
10. Madec G., P. Delecluse, M. Imbard, and C. Lévy. OPA 8.1 Ocean General Circulation Model reference manual. Note du Pôle de modélisation XX, Institut Pierre-Simon Laplace, France, 91 pp., 1998
11. MPI. A message-passing interface standard version 2.1. Message passing interface forum, June 23, 2008. http://www.mpi-forum.org/docs/mpi21-report.pdf, accessed 1.06.2010
12. R.L. Graham, G.M. Shipman, B.W. Barrett, R.H. Castain, G. Bosilca, and A. Lumsdaine. Open MPI: A high-performance, heterogeneous MPI. Proceeding of the Conference HeteroPar′06, in Barcelona, Spain, September 2006. http://www.open-mpi.org/papers/heteropar-2006/heteropar-2006-paper.pdf, accessed 1.06.2010
13. J. Seward, N. Nethercote, J. Weidendorfer, and the Valgrind development team. Valgrind 3.3 – Advanced Debugging and Profiling for GNU/Linux applications. http://www.network-theory.co.uk/valgrind/manual/, accessed 1.06.2010
14. See the description of the Paraver tool at the site of the Barcelona Supercomputing Center – http://www.bsc.es/plantillaA.php?cat_id=485, accessed 1.06.2010
15. O. Hartmann, M. Kühnemann, T. Rauber, and G. Rünger. Adaptive selection of communication methods to optimize collective MPI operations. Parallel computing: Current & future issues of high-end computing, Proceedings of the International Conference ParCo 2005, G.R.

Joubert, W.E. Nagel, F.J. Peters, O. Plata, P. Tirado, E. Zapata (eds), John von Neumann Institute for Computing, Jülich, NIC Series, Vol. 33, ISBN 3-00-017352-8, pp. 457–464, 2006

16. J. Li, W. Liao, A. Choudhary, R. Ross, R. Thakur, W. Gropp, R. Latham, A. Siegel, B. Gallagher, and M. Zingale. Parallel netCDF: A High-Performance Scientific I/O Interface. SC2003, Phoenix, Arizona, ACM, 2003
17. G. Hager, G. Jost, and R. Rabenseifner.Communication characteristics and Hybrid MPI/OpenMP Parallel programming on clusters of Multi-core SMP nodes.proceedings of the cray users group conference 2009 (CUG 2009), Atlanta, GA, USA, May 4–7, 2009, https://fs.hlrs.de//projects/rabenseifner/publ/CUG09_Hager_Jost_Rabenseifner.pdf, accessed 1.06.2010

Network-Centric Earthquake Engineering Simulations

Paolo Gamba and Matteo Lanati

Abstract This chapter shows a Grid application for the seismic community and specifically for earthquake engineering. The high complexity, cost and wide geographical dispersion of testing facilities, together with the extreme complexity of actuator control systems, concur to make seismic tests a very scarce resource for interested researchers. Moreover, the high computational burden of the simulation software developed for the same task calls for distributed processing. As a consequence, network-based simulations, as well as the possibility to share and remotely control resources in different locations are all the *el dorado* for earthquake engineers.

The recent development of Grid elements able to connect physical instruments and tools to a system already capable of distributed processing and storage opened the field for the development of a first pilot application, able to demonstrate the Grid potentialities. This chapter shows the Grid and network requirements of this pilot application, the elements that have been developed for its definition, with stress on the connection of the Grid to earthquake engineering test instrumentation.

1 Introduction

There is a growing need by many different research groups of collaborative research made using scarce and geographically sparse instrumentation [1, 2]. This request has been addressed in these years, as for the computing power and storage space is concerned, by means of Grid technologies. Only recently [3] the Grid has moved to consider instrumentation as a resource that can be shared using the same structure. Correspondingly, there are only a few examples of applications showing the potentials of the Grid and this paper is aimed at describing one of these, developed in the framework of the DORII (Deployment Of Remote Instrumentation Infrastructure) project, funded by the 7^{th} framework programme of the European Community.

M. Lanati (✉)
Eucentre, via Ferrata, 1-27100, Pavia, Italy
e-mail: matteo.lanati@eucentre.it

F. Davoli et al. (eds.), *Remote Instrumentation Services on the e-Infrastructure*,
DOI 10.1007/978-1-4419-5574-6_18,

Requirements from physical instrumentation are likely to be more diverse than those from computing or storage elements, due to the wider variability of instrumentation types, outputs and interfaces. As long as it is possible to connect this wealth of possibilities to the unique Grid management structure, the power of the Grid will become huge. This has been implemented recently by means of a similar abstraction than the storage or computing element, the so called "instrument element" [4]. The availability of this specific piece of middleware makes the realization of instrumented Grid applications possible and opens the door for interesting collaborations between, for instance, civil and electronic engineers.

A second aspect which is interesting and marks a difference when physical instrumentation is added to the Grid is the set of network performances that start to be requested. While computing power and storage areas are substantially time independent resources (in the sense that they can tolerate delays), the instrumentation is tightly connected to the work flow and delays or time alignments are more important. The Quality of Service provided by the network is thus usually considered as an important aspect for remotely controlled instrumentation [5, 6, 7]. Finally, the possibility to add wireless sensors to Grid instrumentation through the definition of "ad hoc" protocols [8] provides another very appealing possibility for enhancing the supported application ranges.

The research work described in this chapter refers to a specific application, the collaborative use of software and hardware resources for earthquake engineering simulations and tests. The following section will thus summarize the state of the art, while Section 3 will be devoted to introduce the Grid application developed as a pilot study to address some of the current research issues in this area by means of the Grid tools. Stress will be on the instrumentation connected to the Grid, especially on the actuators, that provides to Grid applications the possibility to directly interact with the reality, by applying forces on actual structures. Finally, Section 4 discusses the issue related to network performances and requests and shows the path to future developments and improvements.

2 State of the Art of Earthquake Engineering Simulations and Testing Activities

The main objective of simulation activities in civil engineering applied to earthquakes is the estimation of building's behavior during a seismic event. There are a variety of tools to accomplish this task, both analytical and experimental, showing complementary features. In the first category we can find *computer-aided simulations*, based on solution of differential motion equations regulating the physics of a building or exploiting a finite element analysis [9]. Models can be arbitrarily complex, but computation time gets greater and greater, especially if scientists does not rely on parallelization techniques for hardware and/or software [10]. An example of a graphical input for these software is proposed in Fig. 1a.

The complementary approach consists in setting up a test environment involving a reproduction, scaled or not, of the building to be studied. This solution needs,

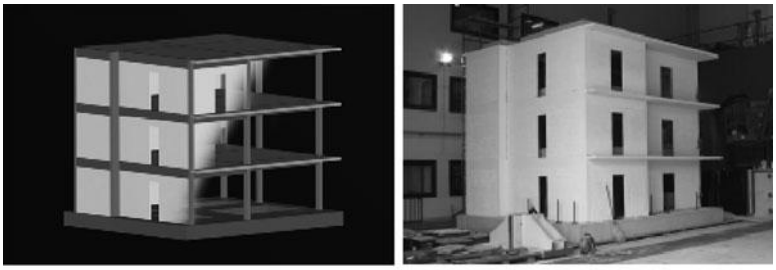


Fig. 1 Simulation of the behavior of a structure subject to seismic stress: (**a**) simulation input and (**b**) actual 1:2 scaled model placed on the shaking table in EUCENTRE

of course, a specimen and some expensive equipment to interact with it: for example, a shaking table (for dynamic tests) or a hydraulic press (for static tests), plus all the instrumentation to take measurements. Moreover, these instruments are not very common because a certain level of customization and trained dedicated staff is usually required. Finally, the specimen can be employed in a limited number of experiments, since it is subject to damages and cracks, as in destructive tests. As an example the shaking table in EUCENTRE, the biggest in Europe [11], is shown in Fig. 1b.

It is easy to understand that negative aspects of one solution can be greatly compensated by the other one and the optimal approach should take into account all the advantages. In the particular case of structural movement prediction, it is possible to successfully merge computer simulations and physical experiments in a so called *pseudo-dynamic test*. This is a well-known and studied subject and its main concept is summarized in Fig. 2.

The starting point to describe specimens' behaviour is always represented by structural motion equations. Unfortunately, even if they are linear, some aspects that should be taken into account, such as reaction of the walls, show great non-linearities. This is the point where computational efforts are concentrated.

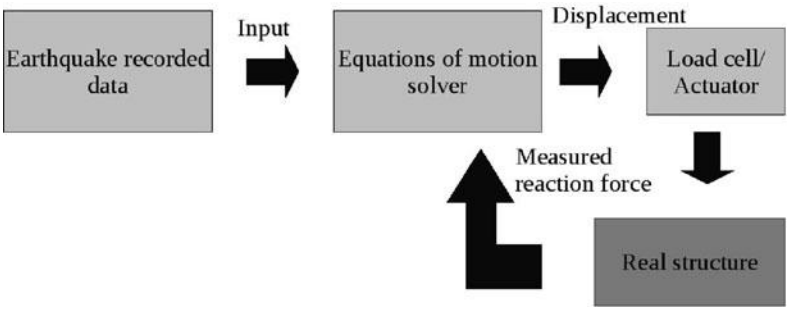


Fig. 2 A schematic representation of the concept of a pseudo-dynamic test

Pseudo-dynamic tests mitigate this drawback by means of measurements on a real structure, specifically, the aim is to extract information about non-linear components. It is easier to get these data from the specimen rather than waste time and resources in mathematical operations. In order to record its reaction, the specimen has to be stressed by an actuator, imposing a force or the corresponding displacement (conversion is usually made internally by the actuator controlling unit). The force (or displacement) is obtained by the solution of the motion equations, according to strains imposed by the "simulated earthquake" shaking the building. In turn, the physical non-linear response of the object is fed into next step of the equation solver. This closed loop is repeated step by step in a time-discrete domain which is of course much slower than the real time flow, hence the "pseudo-dynamic" name.

Examples of this "mixed" approach have been already implemented in a distribute environment: the most important for relevance, funds and people involved is NEES [12] and its technological extension, NEESGrid [13]. NEESGrid aims to at developing a common software platform based on Grid services in order to acquire and manage data together with cooperation and telepresence services. This represents a relevant step, since it is not only possible to remotely access and share data as the World Wide Web revolution already showed, but instruments too can be part of this process. It was necessary to adapt a pure computation and storage platform to telecontrol, stressing the demand for uniformity of performances, answer to data loss and strict coordination.

A further improvement can be the introduction of *substructuring*: the structure under study does not need to be placed in the same laboratory, but it can be split in various parts, even in different locations [14]. It is also possible to replace one of the parts with an analytical model run by a computer, in a hybrid and distributed way.

Again, former examples are related to NEES, in particular to the MOST (Multi-Site Online Simulation Test) experiment [15]. The core of the application was the so called NTCP (NEESGrid Teleoperations Control Protocol), a software component giving seamless access to instruments and software model, disregarding the implicit heterogeneity. Via this software it is possible to define whether a building component is simulated or represented by a specimen. NTCP is split in two parts: a server plugin in different versions, to be associated to hardware controllers or software components, and a client module, acting as a central simulation controller and collecting the status of all servers. The client sends a "proposed action" to the server, which verifies the consequences of the operations; if it is safe, then the answers is an "execute" command, in the other case the action is cancelled. This transaction-oriented structure is necessary because it is not possible to "undo" an operation and safety is always the priority. The experiment started on July 30th, 2003 and the structure being simulated was a a two-bay single-story steel frame. The central element was simulated while an experimental set up was deployed for the two side columns. The test needed 5 hours to complete 1493 of the 1500 steps scheduled, and MOST demonstrated the feasibility of this approach.

3 The Proposed Grid Application

According to the previous discussion, distribution and high volume computation are fundamental prerequisites for a platform dedicated to pseudo-dynamic tests but there are some non-trivial problems. The ideal working environment for this purpose should:

- support fast distribution and deployment of applications;
- be aware of user privilege hierarchy;
- provide enough computation resources;
- simplify and standardize the interaction with physical instruments;
- take into account the dynamics of scientific communities.

In our opinion, Grid technologies meet most of the above mentioned points and some recent European projects such as GRIDCC [16] and RINGrid [17] added a standardized interface for real instruments by means of the Instrument Element (IE) [18], the Instrument Manager (IM) and a common user tool, the Virtual Control Room (VCR) [19].

Given that instrumentation is a strategic part of the DORII project, we think that a brief overview of the Instrument Element (IE) and Instrument Manager (IM) can be useful. The IE is defined by the authors as a set of services for remote configuration, monitoring, management and integration of physical instruments with computation capabilities. The needed interfaces are made available as web services, called VIGS (Virtual Instrument Grid Service). All the information regarding the status of the instruments are collected in a centralized way, then disseminated to interested subscribers. Monitoring service is also used by a specific component of the IE to face arising problems, applying a sort of automatic recovery procedures. Since the goal is to virtualize the usage of equipment, it is clear that is not possible simply to restart the instrument if there are problems, so there is a problem solver which tries to solve simple malfunctions.

It should be clear that the IE is a composite software component and, among all, its core part is the IM, a protocol adapter responsible for the communication with the real instrument. It implements all the functions needed to access specific features and to retrieve status, so the IM is a high customized piece of software that can be developed following a general template. If each instrument needs its IM, in an IE there can be multiple IM instances, one for each logical group of instruments. This means that IMs can be organized in a hierarchical way, so an IE is able to support up to $O(10^4)$ nodes. The IM receives inputs from the user and from the instrument itself and it can decide to perform two actions: update the finite state machine modeling the equipment or interact with the hardware. The implementation of this "intelligence" is left to the developer, because the IM supports from a simple function to a Neural Network.

Concerning the last point, the DORII project [20], under which this work is carried on, focuses on distributed and interactive aspects of applications in some scientific domains. The idea came from the consideration that, besides researchers,

other subjects are involved directly or indirectly in tests. For example, there are software specialists working on algorithms, network manager interested in communication aspects and technicians (structural and civil engineers, in our case) studying the results. However, the release of a pseudo-dynamic platform is a long process and in this article we describe the preliminary results. During the following of the project, a complete and more elaborated system will be implemented and validated.

In substructuring analysis, one or more components of the building under test need to be represented by an analytical model [21]. The problem is that the material employed for walls and other parts is not uniform, so the behaviour is not linear and it is not so easy to calibrate the model. In this case, it is common practice to start from some measured data, in particular a displacement-force diagram, showing the force that should be applied to obtain the given displacement. Therefore, the pilot application is a realization of this preliminary part of the network-based simulation. It varies some significant parameters such as shear stress and elastic modulus (describing the stiffness of the material) in order to fit as best as possible the experimental diagram. Fig. 3 summarizes the procedure.

First of all, it is necessary to load data describing the wall under test: height, width, orientation, centre of gravity and constraints, since the wall is modeled as a frame between two points. Data include maximum and minimum values of some material's properties, such as ultimate strength, while their average represents the starting point for the analysis. The second block summarizes the collection of empirical data, that is the reference graph obtained stressing the physical specimen.

We implemented a pilot version of the application which automates and remotizes this procedure, simply using available instruments through the IE and the VCR. In particular, the involved equipment includes an accelerometer, a displacement sensor, a load cell and an actuator (or better, the electro-valve regulating the oil pressure of the hydraulic ram).

More details about the instrumentation involved and its connection to the Grid are available in next subsection. Here we focus on the computational part, which was originally developed in Visual Basic and then translated into a C main with the

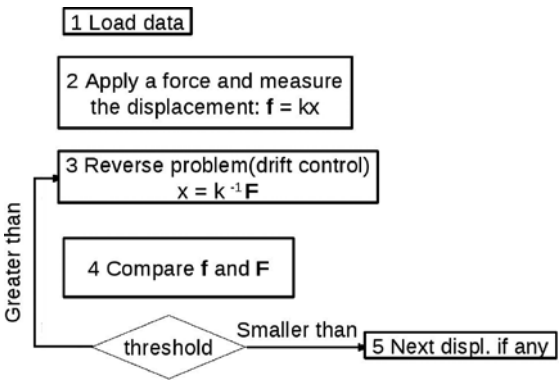


Fig. 3 A graphical representation of the step by step procedure implemented into the pilot application

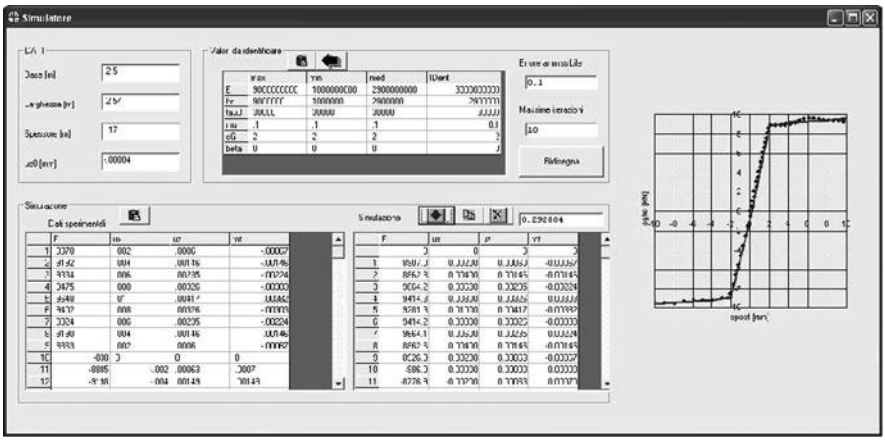


Fig. 4 The original software input and output mask developed for software used in the pilot application

aim of an easier porting into the Grid. The original input and output mask of the application is shown in Fig. 4.

The algorithm implemented in the software is relatively simple. The basis is the following equation

$$f = \vec{K} \cdot x \tag{1}$$

that is, the force is the product of the displacement x and matrix $\vec{K}$, which in the usual notation represents the stiffness properties of the material.

The software is meant to solve the reverse problem via an optimization and a step by step procedure: for each displacement (available after the measurements) due to a known force applied to the object under test, it is possible to compute the force F that would be needed, according to a certain set of mechanical parameters (the matrix $\vec{K}$ in Equation (1)).

By comparing the measured force f and the assumed force F, clues on the parameter set may be inferred and the stiffness matrix is refined in an iterative way. The procedure ends if the error is below a fixed threshold, decided by the user, and at this point it is assumed that the parameter set is good enough. On the other case, the program varies the parameters, calculating a new solution until a better performance or a maximum number of iterations is reached. This process ends when all displacements have been analyzed, and the model has been validated for the range of forces that will be used in subsequent tests for the specimen.

In Fig. 5 we compare the performances of the original Visual Basic code, the black line, and the translated C version, in grey, in rebuilding the starting force - displacement diagram.

We can see that there is a substantial improvement in the overall precision. Long double variables are used to avoid some approximation problems encountered during the porting phase.

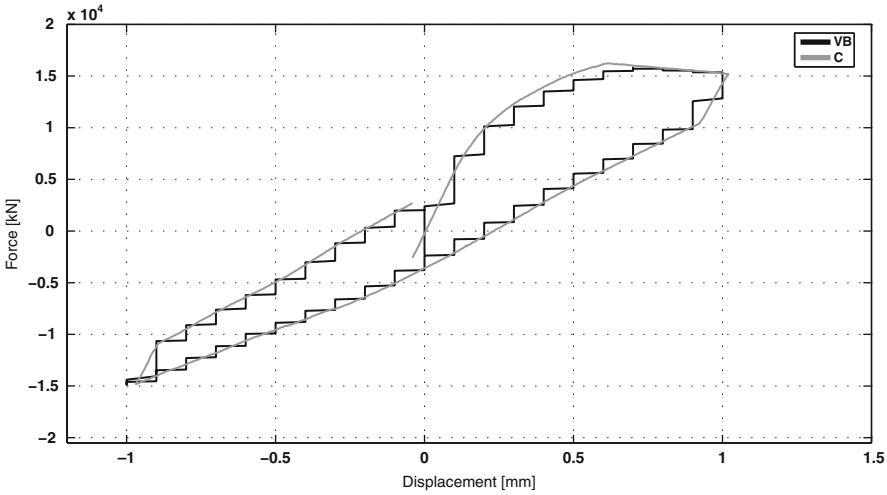


Fig. 5 Comparison between C and Visual Basic code performances

3.1 The instrumentation Put into the Grid

In order to extract inputs and use the output of the software described in the previous paragraphs, a suitable set of instruments should be connected to the Grid, via the above mentioned instrument element. In this subsection we propose a short review of the instruments employed and connected to the Grid during this project.

In the *passive* category we can find transducers, in particular accelerometers, displacement transducers and load cells. They convert a physical quantity (acceleration, displacement and force, respectively) in an electrical signal, so an A/D converter is needed to exploit this information. For this purpose, we chose an Arduino board, a very flexible micro controller running small C programs, basically to set sample rate and to read acquired data. It can communicate with a PC by means of a serial connection emulated on a USB port and it can be employed with a variety of sensors, simply changing the conditioning network to adapt the voltage range. The generic structure of the gridded instrumentation is shown in Fig. 6, while in Tables 1, 2, and 3 some technical data regarding the above mentioned three types of sensors are reported.

In order to explain in more details the IE/IM paradigm we are going to apply, in Fig. 7 we can see the XML finite state machine virtualizing a generic transducer together with the implemented Java commands used for state transition.

The initial state is called *off*, followed by *setup*, when the IE establishes the serial connection with the Arduino board by means of the Java RX/TX Library. Then the user is supposed to set in the VCR the sampling frequency and some other parameters, passed to the sensor by the *Initialize* function. Now the sensor is ready to acquire data and to transfer them back to the IE. In any moment, it is possible

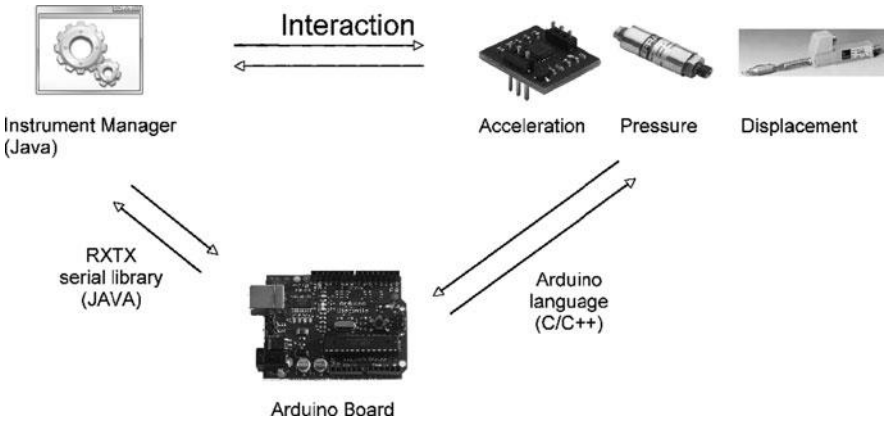


Fig. 6 The generic structure of the system developed to connect the instrumentation required for the pilot application to the Grid

Table 1 Technical specifications for the accelerometer

Sensing element:	MEMS variable capacitance
Acceleration range:	− 2.5 to +2.5 g (peak values)
Sensitivity:	1500 mV/g
Typical resolution @ 10 Hz:	< 2.5 μg

Table 2 Technical specifications for the distance transducer

Stroke:	From 10 to 100 mm
Max. displacement speed:	10 m/s
Max. displacement force:	4 N
Voltage range:	from 14 to 60 V

Table 3 Technical specifications for the load cell

Type:	Column (it measures a force applied along its axial direction)
Capacity (max. force):	200 kN
Max. voltage:	20 V
Resolution:	0.001 kN

to change parameters calling the *configure* function, switching back to the *Initialize* phase.

An example of a displacement transducer working in the user interface can be seen in Fig. 8.

The last sensor used can be classified as *active*, since it is an actuator. In particular it is an electro-valve that regulates the pressure of the oil flowing in the hydraulic circuit controlling the force of an hydraulic ram. The control algorithm

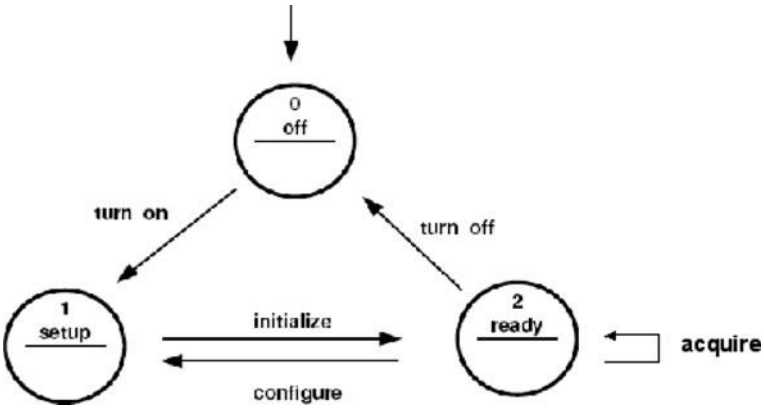


Fig. 7 Finite state machine for a generic transducer

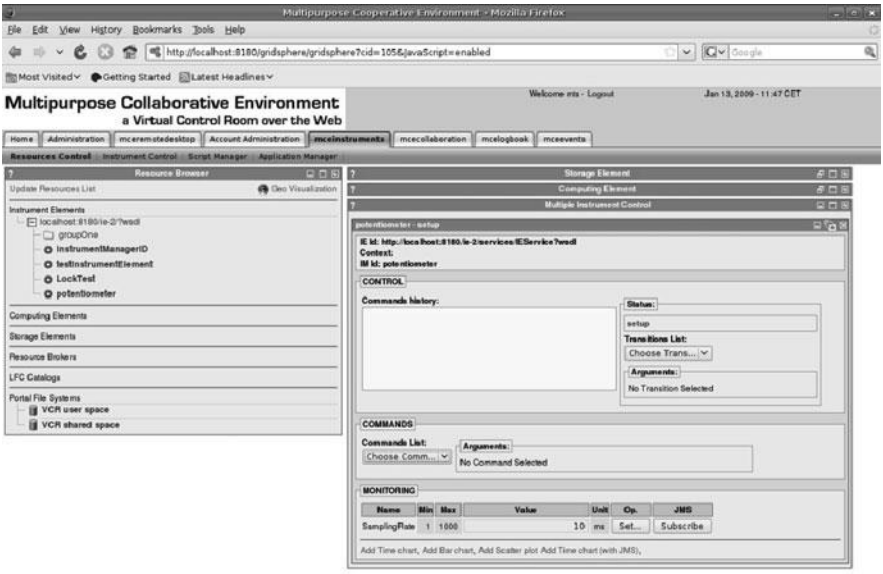


Fig. 8 An example of the user interface

establishes a linear relation between control voltage and force (in the vast majority of the characteristic curve) facing some dangerous situations, for example, saturating the force in case the specimen does not offer any reaction, in order to avoid damages to the hydraulic ram. The valve is controlled by means of a data acquisition board, while the software was implemented in C♯ by the EUCENTRE's TREES Lab (Laboratory for training and research in earthquake engineering and seismology) staff. The system is running on a single PC and it is available as a web service.

Once again, we can have again a quick look to the XML finite state machine summarizing the IM for the electro-valve (Fig. 9).

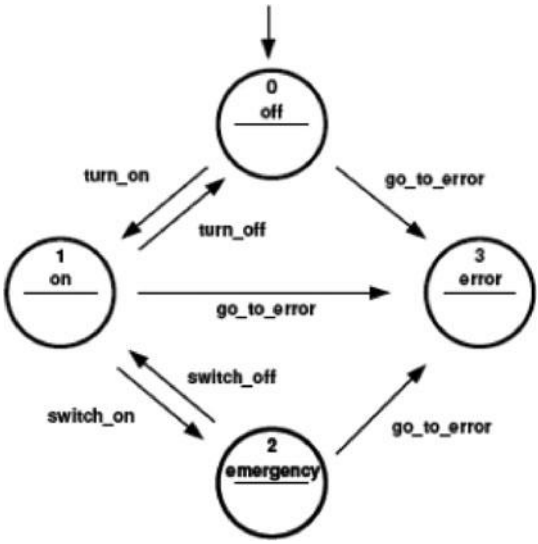


Fig. 9 Finite state machine for the electro-valve

We can notice that a specific state dedicated to initialization was removed, since the instrument should follow a specific pattern of movements, contained in a description file or as a result of the equation solver (in the future). On the contrary, there is an *error* state, acting as target when force or pressure exceed safety thresholds and controller's damage avoidance procedures are not working. Moreover, the user can pause the test in any moment switching to the *emergency* state, so the actuator is frozen in its current position.

We performed a preliminary test of the instruments, to verify the interaction between the various components. Fig. 10 shows our setup: the specimen is not a wall, but a coil spring, so the distortion is greater and much more noticeable.

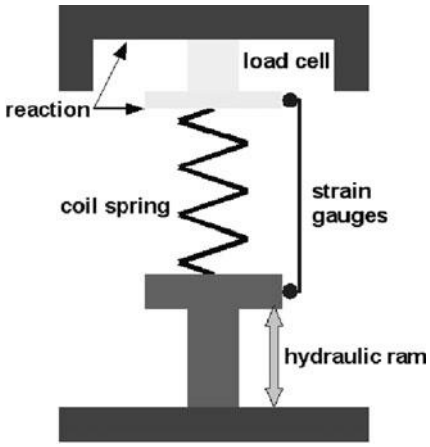


Fig. 10 Preliminary test setup

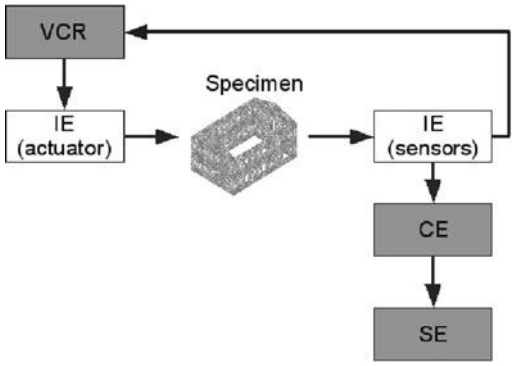


Fig. 11 Grid blocks for the application proposed

The hydraulic ram applies pressure to the coil spring and the load cell measures the related force. We also connected three displacement transducers between the plunger of the press and the reaction facility to estimate the deformation of the coil.

At the end of this section we can summarize in Fig. 11 the Grid blocks involved in our application and how they interact.

First of all, connections between blocks reflect the logical operation flow previously described and shown in Fig. 3. In particular, the IE acquiring data is feeding CE for computation and the VCR to give feedback to the user.

The grey blocks represent remote machines, while white ones are local resources, together with the specimen. EUCENTRE is going to provide only IEs for the infrastructure, running on standard PCs equipped with 1 GB of RAM and the latest release of Scientific Linux Cern. Other services are made available by other members of the DORII consortium, in particular:

- **VCR** by Elettra Sincrotrone Trieste SCpA;
- **CE** and **SE** by PSNC (Poznan Supercomputing and Networking Center), CSIC (Consejo Superior de Investigaciones Cientificas), Elettra, and Greek Research and Technology Network S.A.

There are about 2000 CPUs available for all project partners needs. At the moment, since the application is still under construction, there is not a specialized VO, but we are relying on the general catch-all DORII VO. All local nodes are on a Gigabit LAN, while National Research Networks are responsible for inter-site communications. A relevant effort is put in coordination to achieve suitable guarantees for delays and losses.

4 Discussion and Conclusions

Despite the application is quite simple with respect to pseudo-dynamic simulations, we can find all the main elements, that is computation, geographic distribution of resources and interaction with instruments, so we can draw some considerations.

Simulation output is very small: optimum parameters describing the material making up the wall and a two-axes graph showing the fitting obtained with respect to the starting diagram. But this is only a fraction of the data being transferred, if we consider exchange of information for control purposes.

All the passive sensors have the same behavior from the point of view of the system, since they are interfaced using the same type of A/D converter. The Arduino board represents an analog input with 10 bits, which is a satisfying precision for this kind of measurements, together with a sample rate equal to 10 Hz. This configuration is suitable also to monitor the force applied by the actuator with the load cell, a fundamental safety requirement in order to avoid damages to operators and structures. Due to the limited hardware capabilities of the board, it is necessary to use one converter for each transducer. The electro valve communicates the spread angle and the oil pressure (directly proportional to the force applied in the linear region of the characteristic) with a better precision and frequency, so the amount of information is approximately one order of magnitude higher.

It is common practice to save data coming from instruments in text and binary formats, both in engineering and electrical measuring units. These files should be made available moving them to a storage element, together with simulation results. Starting from our experience, we can estimate that we need to transfer about 100 MB per hour of uncompressed control data. However, since the application and the entire project stress interactive aspects, it could be a good point to support the process with a video stream showing the progress of the experiment. This is not a critical requirement and we are dealing with pseudo-dynamic test, so real-time constraints are not necessary.

Besides the total amount of transfer, it is interesting to consider Quality of Service over the network (see Table 4). It is clear that, once the user set up the input for the application, the simulation runs by itself performing a series of consecutive steps and storage data can be moved in background, but control information are of fundamental importance. They are extremely sensitive to loss events, jitter and delay and it could be useful to split this stream from the optional video support. Another relevant aspect is that there is a continuous flow made up of small packets with a maximum peak value, quite unusual, not greater than 1 MB/s.

Bringing instrumentation into the Grid is a challenging task, with some issues to tackle. First of all, equipment is not designed with interoperation in mind, on the contrary, manufacturers usually tend to protect their work and market position

Table 4 Quality of Service requirements

Fault tolerance:	No
Delay tolerance:	No
Max. delay:	up to 5 ms
Max. throughput:	1 MB/s
Control data exchange (total):	100 MB/hour
Data exchange per sensor:	from 100 b/s to 2 kB/s

behind proprietary standards. So, a deep data sheet analysis and a lot of customization is needed before putting into work an equipment. Then, there is the software component to consider. Grid platform is an attractive solution for many problems and despite there is a great effort in creating user-friendly development environments, sometimes re-implementation is required in order to exploit the full potentiality. It is clear that a good initial planning is fundamental to achieve desired results. In short terms, these are the main difficulties we encountered in our work. Inside the DORII project initiative we planned a distributed tool for hybrid earthquake simulations, an ambitious goal since this is a well investigated subject and expectations are high. In this article we described the initial step toward a complete system, focusing on all main aspects such as actuation, sensing, computation and storage.

We completed some preliminary tests to verify the level of integration. In our setup we employed a press, operated by a technician, to deform a compression coil spring. The load cell measured the force applied while three potentiometers detected the deformation. Concerning the computational side, we verified with some synthetic data the behavior of the code, noticing an improved overall precision.

Acknowledgments The authors want to thank Angelo Stramieri for the work in implementing part of the IM and the useful discussion about the IE. The help of Alessandro Galasco and Andrea Penna for the implementation of the pilot application and the original software to achieve it is gratefully acknowledged. Finally, they want to thank the people at TREESLab in EUCENTRE for the support with the actual instrumentation.

References

1. A. W. Lin, Lu Dai, Khim Ung, J. Mock, S.T. Peltier, and M.H. Ellisman. The telescience tools: Version 2.0, Proceedings of First International Conference on e-Science and Grid Computing, 2005
2. F. Davoli, G. Span, S. Vignola, and S. Zappatore. LABNET: Towards remote laboratories with unified access. IEEE Transactions on Instrumentation and Measurement, 55, 1551-1558, 2006
3. E. Frizziero, M. Gulmini, F. Lelli, G. Maron, A. Oh, S. Orlando, A. Petrucci, S. Squizzato, S. Traldi. Instrument element: a new Grid component that enables the control of remote instrumentation. Proceedings of the 6th IEEE International Symposium on Cluster Computing and the Grid Workshops (CCGRIDW'06) – 17–19 May, Singapore, 2006
4. F. Lelli. Bringing Instruments to a Service-Oriented Interactive Grid, PhD Thesis, University of Venice Ca Foscari, Italy, 2007
5. GFD-I.037, Networking Issues for Grid Infrastructure, Open Grid Forum, 2004, http://www.ogf.org/documents/GFD.37.pdf
6. D. Colling, T. Ferrari, Y. Hassoun, C. Huang, C. Kotsokalis, A.S. McGough, E. Ronchieri, Y. Patel, and P. Tsanakas. On Quality of Service Support for Grid Computing, in F. Davoli, N. Meyer, R. Pugliese, S. Zappatore, Eds., Grid Enabled Remote Instrumentation, Springer, New York, NY, 2008
7. A.S. McGough, A. Akram, D. Colling, L. Guo, C. Kotsokalis, M. Krznaric, P. Kyberd, and J. Martyniak. Enabling Scientists through Workflow and Quality of Service, in F. Davoli, N. Meyer, R. Pugliese, S. Zappatore. (Eds.) Grid Enabled Remote Instrumentation, Springer, New York, NY, 2008

8. G. Moro, G. Monti. (2006) W-Grid: a Cross-Layer Infrastructure for Multi-Dimensional Indexing, Querying and Routing in Wireless Ad-Hoc and Sensor Networks, Proceedings of 6th IEEE Internat. Conference on Peer-to-Peer Computing (P2P'06), pp. 210–220, 2006
9. A. Reali. An isogeometric analysis approach for the study of structural vibrations. Journal of Earthquake Engineering, 10, 1–30, 2006
10. Sanjay Singh1 and S V Barai Earthquake Engineering Problems in Parallel Neuro Environment In: High Performance Computing - HiPC 2004, pp. 200–210, Springer, Berlin 2005
11. European Center for Training and Research in Earthquake Engineering (EUCENTRE) web site http://www.eucentre.it. Cited 15 Jan 2008
12. G.E. Brown, Jr. Network for Earthquake Engineering Simulation (NEES) web site http://nees.org/. Cited 15 Jan 2008
13. The NEESGrid System Integration Team, Introduction to NEESGrid, Technical Report NEESGrid-2004-13 Version: 1.0,August 23, 2004
14. P. Pegon, and A.V. Pinto. Pseudo-dynamic testing with substructuring at the ELSA Laboratory. Earthquake Engineering & Structural Dynamics, 29, 905–925, 2000
15. L. Pearlman, C. Kesselman, S. Gullapalli, B.F. Spencer, Jr., J. Futrelle, K. Ricker, I. Foster, P. Hubbard, and C. Severance. Distributed Hybrid Earthquake Engineering Experiments: Experiences with a Ground-Shaking Grid Application, Proceedings of 13th IEEE International Symposium on High performance Distributed Computing, June 4 -6 2004 Honolulu, HI
16. F. Asnicar, L. Chittaro, L. De Marco, L. Del Cano, R. Pugliese, R. Ranon, and A. Senerchia. Collaborative Environments For The Grid: The GRIDCC Multipurpose Collaborative Environment. In: Distributed Cooperative Laboratories: Networking, Instrumentation, and
17. RINGRID project web site http://www.ringrid.eu. Cited 15 Jan 2008
18. M. Prica, R. Pugliese, C. Scafuri, L. Del Cano, F. Asnicar, and A. Curri. Remote Operations of an Accelerator using the Grid. Proc. of INGRID 2007 - Instrumenting the Grid 2nd International Workshop on Distributed Cooperative Laboratories - S.Margherita Ligure Portofino, Italy, pp. 279–284, 2007
19. R. Ranon, L. De Marco, A. Senerchia, S. Gabrielli, L. Chittaro, R. Pugliese, L. Del Cano, F. Asnicar, and M. Prica. A Web-based Tool for Collaborative Access to Scientific Instruments in Cyberinfrastructures, F. Davoli, N. Meyer, R. Pugliese, and S. Zappatore (Eds.), Grid Enabled Remote Instrumentation, pp. 237–251. In Springer, Berlin, October 2008
20. DORII project web site http://www.dorii.eu. Cited 15 Jan 2008
21. M.C. Griffith, G. Magenes, G. Melis, and L. Picchi. Evaluation of out-of-plane stability of unreinforced masonry walls subjected to seismic excitation. Journal of Earthquake Engineering, 7, 141–169, 2003

A Grid Approach for Calibrating and Comparing Microscopic Road Traffic Models

Luca Berruti, Carlo Caligaris, Livio Denegri, Marco Perrando, and Sandro Zappatore

Abstract The chapter deals with an original Grid based approach for vehicle traffic monitoring. Specifically, the devised solution involves a set of Wireless Sensor Networks (WSN) for acquiring the data on traffic from roads, cross-roads, and roundabouts. Then a Grid actor, an Instrument Element according to the GRIDCC paradigm, publishes the data gathered from the field by means of a continuous polling of the sinks that coordinate the WSNs. The proposed architecture employs a number of Storage Elements and Computing Elements in order to suitably archive the traffic information and process it. The final goal is to calibrate one or more traffic models on the basis of traffic data related to some actual situations. Particular attention is paid on what concerns the communication among the WSNs, the related sinks, the Instrument Managers and the Instrument Element, which are directly involved in the data acquisition and publication on a worldwide Grid.

1 Introduction

Grid computing architectures or, more generally, service oriented architectures (SOA), viewed as platforms for the integration of distributed resources, play an ever-increasingly significant role, as witnessed by the large number of projects funded by the European Union (EU) on this and related topics. In such a context, the functionality offered by the Grid architecture allows unified control, maintenance, calibration and utilization of "traditional" instrumentation and laboratory equipment, as well as of remote data acquisition probes: the aim of the Grid is to provide the user common standard interfaces and working environments [3].

A significant number of research projects has dealt with this issue, first of all GRIDCC, one of most relevant programs in Europe. GRIDCC has extended the EGEE gLite [8] middleware with some functionalities explicitly addressed to

S. Zappatore (✉)
CNIT-University of Genoa Research Unit, Genoa, Italy
e-mail: sandro.zappatore@cnit.it

F. Davoli et al. (eds.), *Remote Instrumentation Services on the e-Infrastructure*,
DOI 10.1007/978-1-4419-5574-6_19,

accessing and controlling instrumentation in real-time. Successively, the European projects RINGrid [19] and DORII [17] aimed at developing a Grid infrastructure able to integrate heterogeneous scientific instrumentation in complex workflows. Specifically, the DORII platform has extensively exploited sensor networks for environmental and earthquake monitoring, as well as for autonomous underwater vehicles' driving. Nevertheless, although the advantages of the extension of SOA to instrumentation are clear and largely recognized, no operative standard has been definitively adopted.

As concerns wireless sensor networks, they represent a promising opportunity [1, 9] to efficiently monitor and track a wide range of physical phenomena including the traffic of vehicles passing through a road. Specific protocols have been developed to configure the network, by flooding topological information, and to route data from sensors to the sink. To overcome long distances, protocols may allow multihop transmissions. Nodes may send packets to the sink by means of other nodes, but they must also be as simple as possible to save batteries and to cope with limited computational and storage capabilities of the nodes. Wireless Sensor Networks (WSNs) can be organized as meshes, by exploiting multihop protocols, but also clusters or hierarchical structures can be adepted during network design. Nowadays, the availability of cheap single-chip computers and miniaturized radio transceivers (e.g. spread spectrum radiomodems by Cypress [5], ZigBee radio-controller by Ember [6]) makes easy to design and implement small-dimension wireless nodes, which can be installed everywhere with very low effort. The approach discussed in this paper is centered on a set of single-hop WSNs described in [4], whose nodes count vehicles by means of simple active infrared sensors and/or magnetometers.

This chapter proposes a Grid application for a tele-measurement system designed for calibrating one or more traffic models that simulate urban roundabouts or traffic lights managed cross-roads. Within this framework, the Grid nodes must be able to estimate traffic volume and speed information from actual sensors and compute predefined algorithms. From this perspective, the novelty of the approach consists of neither the WSN itself nor the Grid solution, but rather of highlighting and solving the specific architectural problems involved in deploying a sensor network that fruitfully exploits a GRIDCC based infrastructure in order to publish, archive and process data gathered from the field.

The chapter is structured as follows. The next Section summarizes the models commonly used to describe the traffic behavior along roads and cross-roads, and roundabouts. Section 3 illustrates the overall architecture employed to gather information from the field and to share the acquired data among the various elements of a GRIDCC infrastructure. Section 4 focuses on the implementation of the Detector, which plays the role of coordinator of a set of sinks and acts as a IE in order to publish the data collected. Section 5 briefly presents an example of how a user can control an experiment devoted to calibrate/validate a group of traffic models. Finally in the last Section conclusions are drawn.

2 The Choice of the Traffic Model

A number of models has been developed for simulating traffic systems: they can be classified into microscopic, mesoscopic and macroscopic models. Usually, on-line and real-time applications are achieved by implementing macroscopic models, since they are characterized by relatively low computational burden [12, 16, 20, 15]. On the other side, these models are suitable to describe traffic flows like the freeway's: if we want to deal with urban traffic, microscopic models seems to be more suitable, as they represent the behavior of every single vehicle in the traffic stream. A high-detail level should produce results better describing the reality, since it takes into account many drivers' behaviors. Thus, microscopic models are usually implemented for small systems such as a single cross-roads or access ramps.

Many microscopic models have been developed (a famous ancestor is [14]): all of them need a number of parameters to work. Some of these parameters are "structural" – such as the number of access roads, the geometry of cross-roads, the time-cycle of traffic-lights, etc. – others are statistical information by which we model the driver's behavior – just like the parameters of the function that determines the arrival rate of the vehicles, the turning percentages of the vehicles, and other probabilities which aim at modeling the particular choices of the drivers [7].

If we want to carefully forecast the future behavior of the traffic, we must tune all the parameters of one model in order to make it as representative as possible of the reality. If this is not done in a proper way, we are not able to examine the model performance, since we cannot distinguish model's errors and approximations from initialization inaccuracies. The correct estimation of such parameters, obtained from raw data measured on the field, represents a crucial task for predicting traffic.

3 The Proposed Architecture

The proposed architecture aims at estimating actual traffic parameters. The estimation is based on field measurements permformed by a large and spatially distributed set of sensors, deployed in proximity of traffic roads, cross-roads and roundabouts that must be monitored. The sensors communicate the acquired data to a special node (i.e. the "sink" [1]) through an *ad hoc* wireless network.

The architecture provides a service oriented application which plays the role of a proxy between the users and the sensors deployed on the field. It consists of four loosely coupled actors; each of them implemented as a Grid node. Figure 1 depicts the proposed architecture, which includes a set of Instrument Elements (IE) [11], a Storage Element (SE), and one or more Compute Elements (CE) [10]. All these are components originally developed within the European Project GRIDCC [18] and reused afterwards within the DORII Project [17].

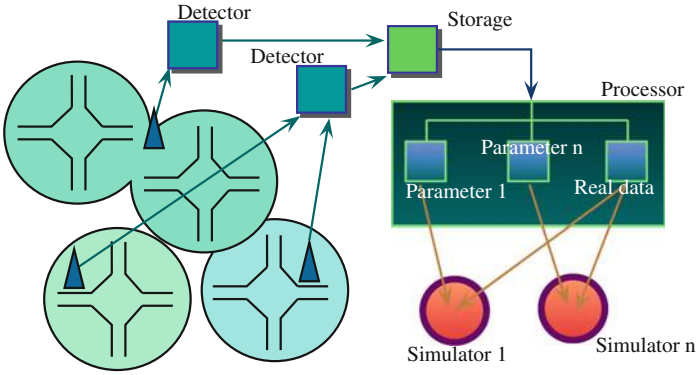


Fig. 1 The proposed grid architecture

3.1 Detector

This component gathers information from one or more of the sinks of the sensor networks. Whenever triggered by a vehicle passage the sensors send a packet to the sink. The data transmitted also include the timestamp of the event and an identifier of the road in which the event was recorded. Figure 2 shows a typical layout of a single cross-road.

The *detector*s also broadcast their active state: this is achieved with a "heartbeat" that indicates the device is correctly running; in this way, we can easily understand whether an absence of data is due to an effective absence of traffic or to a failure of the technology.

The extension of the sensor network is not expected to be as large as to motivate the adoption of a multi-hop protocol. A possible implementation is described in [4], where a single-hop sensor network had been developed. In [4] sensor units are designed to manage several digital and analog inputs/outputs and, equipped with

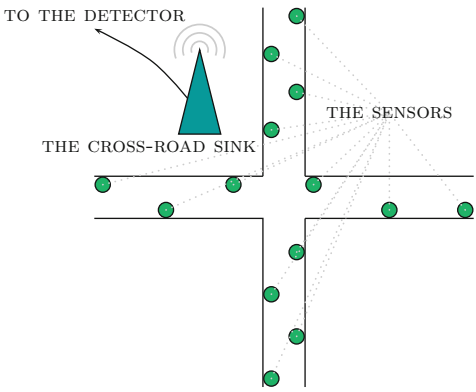


Fig. 2 An example scheme of the sensor network and the sink/grid node

proximity sensors, fully cover the need of our system, being able to capture an event whenever a vehicle passes through their working range. The *detector* exposes its services by using the GRIDCC "Instrument Element". The IE contains a number of "Instrument Managers (IM)" for driving specific local and/or remote devices.

3.2 Storage

The data detected by the *detector* are stored into a dedicated node: the *storage* node. Its positioning is not mandatory (it could be in a safe location, far from the road). The node archives measurements sent by each detector node. When data are received, they are marked with an identifier of the cross-road at which the event occurred.

The introduction of this middle-tier has two major impacts: (i) the memory of the *detector* nodes can be very small: they should only remember the set of unsent data; (ii) whenever a *detector* node should be damaged, we can still access all the data it measured before its failure. These advantages are particularly important for the *detector* nodes, since they are the most sensitive part in the structure, being hosted on single board computers and deployed in a non-protected environment; (iii) in this way, whoever deploys a detector node can use the storage service and make its data available to all Grid users.

The *storage* simply consists of the GRIDCC "Storage Element".

3.3 Processor

This actor retrieves data from the *storage* node and processes them in order to give the users a higher level information. Different nodes can be prepared to answer different requests. Specifically, a user could ask for calculating some statistical parameters or retrieving some aggregated data over a specified time span.

In the former case, the system will work with some built-in algorithms to obtain one or more statistical summaries; this values correspond to the statistical parameters we introduced in the model overview. In the latter case, one wants to know how traffic behaved: the node will report the past traffic conditions in terms of pre-defined state variables (e.g., the queue length or the time-to-travel).

The *processor* consists of a GRIDCC "Compute Element", or, alternatively, of a GRIDCC "Instrument Element". In this case, it reads data from the *storage* node and performs statistical operation by means of a dedicated IM.

3.4 Simulator

This actor is activated by the user in order to forecast traffic evolution according to a specific traffic model. Many models can be implemented and each of them will be represented as a single node. This actor has a double interaction with the *processor* elements (PEs): at first, it uses those nodes to obtain the traffic parameters in order to

initialize the model; afterwards it asks the PEs for the actual data in order to evaluate the fitness of simulation results.

The *simulator* consists of a GRIDCC "Compute Element".

4 The Detector

In this section the focus is on the system components directly involved in the data acquisition and publication on the Grid. The components include one Detector, one or more Sink Bridges and the same number of independent WSNs, with their related sinks.

The nodes of each sensor network get information from the field and deliver it to the sink. The latter is connected by a serial line to a Sink Bridge (SB), implemented on a Linux based Single Board Computer (SBC), which plays the role of a communication bridge between the WSN and the Detector. To this purpose, the SB publishes the data collected by the WSN by means of a SNMP agent. Then, the Detector can query the SNMP agents (viz. the Sink Bridges) over an ad-hoc private IP network. The choice of SNMP is motivated by the wide diffusion of this protocol, commonly adopted in a great variety of monitoring applications, making the WSN very reusable in other contexts. Generally, there are not energy constraints at this level, because of the complexity (both from a hardware and software point of view) of Detectors and SBs. The availability of power at these devices allows using common wireless communication standards, such as IEEE 802.11 in an ad-hoc mode.

Moreover, since the Detector/SBs intranet is completely isolated and private, the security of the communication can be easily assured by adopting one of many security standards. The Detector node must be equipped with two IP network interfaces, one used to create the previously mentioned ad-hoc network, the other employed by the IE (implemented in the Detector) to publish the data over a world-wide Grid.

More in detail, the Detector software architecture consists of one IE and one or more Instrument Managers (IMs). The IE receives the requests (a sensor reading, in this case) from the Grid and, if needed, invokes a proper IM in order to complete the request. The IM browses a local runtime database to seek the information. In turn, the runtime database is continuously updated by the IM, which periodically polls the SBs via SNMP queries. As the Detector runs on the same hardware of the sink and the IE may include several IMs to collect data from the SBs, the designed architecture presents a good level of homogeneity and scalability, at least at the WSNs - Detector level.

A possible implementation of the proposed architecture is diagrammatically depicted in Fig. 3. It consists of three identical SBCs: all are equipped with an 802.11 interface, but only one (the SBC of the Detector) mounts an IP WAN device (e.g. a GPRS/UMTS modem). On all the SBCs the Sink Bridge software reads the data from the sink through a serial line and the sink actually coordinates the sensor nodes of the related WSN. Finally, the Detector hosts part of the GRIDCC/DORII

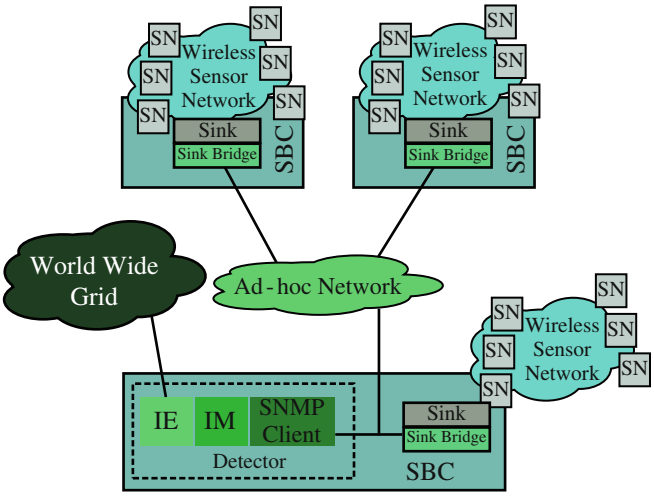


Fig. 3 Detector-sensor network architecture

software suite. Specifically, the Instrument Element publishes the data the IM gathers from all the Sink Bridges connected through an ad-hoc IP wireless network or from the internal "localhost" address, in the case the SB resides on the same SBC of the Detector. The IM implements a SNMP client, which actually queries the SBs.

5 Example of an Experiment Control

The architecture proposed for the detectors and WSNs can be fruitfully integrated within a great variety of SOAs.

In the following, we illustrate two possible applications: the former is devoted to monitor traffic and validate different traffic models, the latter is aimed at realizing a real–time traffic surveillance system.

As concerns the former application, upon the completion of the deployment of WSNs and detectors, the processor nodes will be able to validate and score different traffic models on the basis of real traffic observations.

The GRIDCC approach provides the tools to implement a sort of experimental bench: for instance a "Virtual Control Room (VCR)" [2, 11] may be employed by a user in order to coordinate/control the experiment and compare the scores of different models. In this way, each user can develop its own model, upload it on the simulator node, and get a "vote" of the effectiveness, also compared with the ones of already existing models. Fig. 4 illustrates a conceptual scheme of the proposed Virtual Control Room.

As regards the application devoted to the real–time traffic surveillance, the SOA will allow one or more users to monitor the traffic of the roads where Detector nodes and their related sensor networks have been deployed.

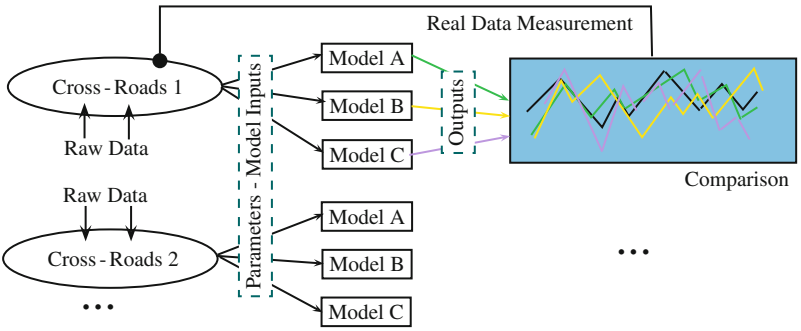


Fig. 4 The virtual control room conceptual scheme

In this case, the IEs at detector nodes may exploit ancillary Java Message Servers (sometime called JMS brokers) in order to efficiently publish the collected data through a publish/subscribe approach, thus enabling a group of monitoring stations to receive data from the same IEs/Detectors.

At the same time, though Storage nodes are not strictly required for this application, the data gathered by the IEs/Detectors may be also sent to Storage nodes, to allow long-term storage of collected data. In this manner, stored data can be successively used for offline simulations and model validations, as previously described.

In this context, the IEs at Detector nodes have to know the URLs of the Storage Elements (SEs) in order to send them the collected data; on the contrary, the IEs do not need to know anything about surveillance stations, which instead retrieve the data from the JMS broker(s).

At the user end, in order to give the user a meaningful representation of the collected data, ad–hoc developed software could be required. This software, by means of Grid APIs, retrieves, via the JMS broker(s), the data published by the IEs/Detectors selected by the user, and then, after some kind of smart processing, it displays graphs and measures useful to know the road traffic status in real time. Figure 5 illustrates a conceptual scheme of the real–time traffic surveillance system.

However, some final comments are in order. The number of Detectors involved in the monitoring activity is generally much less than the number of the IEs/Detectors that send data to the Storage Elements: as a matter of fact, surveillance stations have to monitor only a limited number of roads; on the contrary, validating a traffic model often requires data acquired by a multitude of Detectors (and archived in a possibly large number of Storage Elements): in other words, tuning a model could involve the knowledge of the traffic data accumulated over a large geographic scale. The scenario here discussed can be framed into a wider class of eScience applications where the devices to be controlled are made up by a huge number smaller components (e.g., sensors) distributed over a small– to medium–scale geographical area. In these situations, the asynchronous data collection based on publish/subscribe mechanisms (e.g., JMS messaging) may play a relevant role, especially in the case where

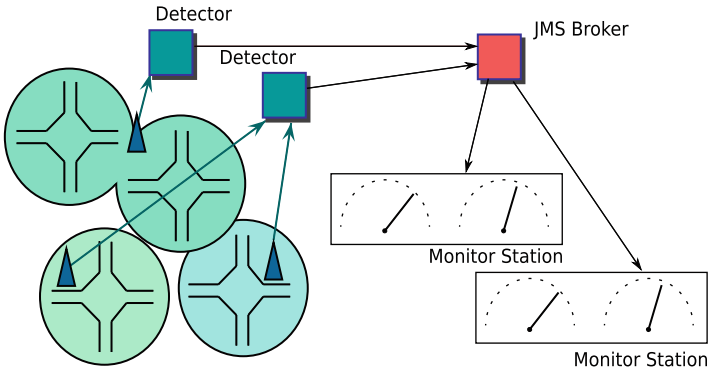


Fig. 5 The real–time traffic surveillance Conceptual Scheme

one user needs to subscribe to the output of many small devices, after a possible data aggregation at intermediate nodes, such as sinks of a sensor network. This approach allows coping with various possible drawbacks: (i) lack of application–level messaging capabilities on–board the sensing devices; (ii) better exploitation of local wireless network infrastructures and of multicast capabilities on wired networks; (iii) reduction of the processing load on the user machine.

These aspects somehow impact also on the possible extension of the IM employed in the platform here presented to a more general paradigmatic structure that enables "Remote" IMs (RIMs) to remotely handle interactions with the myriad of generic small devices that will form the so-called "Internet of Things" [13]. One of the problems to face while devising a RIM, is the design of the communication protocol by which the resident portion of RIM (LP-RIM), the Local Portion of RIM co-located within the IE) can communicate with its remote portion (RP-RIM). The latter could indeed reside even on platform with reduced local computational and storage capabilities. In some sense the use of the SNMP based platform proposed in this paper can be regarded as a specific impletation of a RIM.

6 Conclusion

The chapter has presented an original Grid-based architecture useful for vehicles traffic monitoring. The data related to the traffic of roads, cross-roads and roundabouts are acquired by a group of sensors that form a set of wireless sensor networks. The sensors exploit cheap active infrared sensors and/or magnetometers in order to count vehicles: although the single count measure should be affected by errors, the possible (and hoped) presence of a large number of sensors (within the zone to monitor) may strongly improve the overall quality of the estimation. To this aim, the gathered data are properly integrated at the Detector, a GRIDCC IE which collects (via SNMP) the information received from the Sinks/Sinks Bridges deployed on the field. As the employed components (i.e., the IE and the Sink Bridges) are

hosted on the same kind of hardware and a number of IMs can be used to handle the communication between the Detector and the Sink Bridges, the devised architecture is characterized by a good level of homogeneity and scalability. Furthermore, the adoption of the SNMP makes the information acquired through the WSNs reusable in other contexts. The chapter 4 has also illustrated two examples of possible applications in which the various components of a GRIDCC architecture (i.e. IEs, CEs, SEs, and, a Virtual Control Room) are fruitfully integrated to implement a distributed system for monitoring traffic and validating different traffic models. Surely, a deep performance analysis has to be carried out in order to evaluate the actual efficiency of an IE running on a limited resources hardware platform (in our implementation, a Single Board Computer by Technologic Systems Inc.) Specifically, a comparison between a standard Java implementation and an ad-hoc (simplified) "C" implementation of the IE should be in order with the purpose to establish the power needs, robustness, and reliability of the overall system.

Finally, it should be highlighted that the concept of Remote Instrument Manager, briefly outlined above, can also play the role of mapping "universal" network layer addresses to simpler local addressing schemes like those used in WSNs.

Indeed, in the light of IPv6 deployment, where each small device may possess an individual address (at the network layer), it may be advisable to maintain this identification, even in the presence of devices supporting only "tiny", non–IP protocol stacks (e.g., sensor nodes forming a mesh). In this case, a gateway functionality may be implemented at sinks, where a correspondence can be maintained between the physical device and its "outer" IPv6 address, which will be seen by the external world.

References

1. I.F. Akyildiz, Weilian Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. Communications Magazine, IEEE, 40(8):102–114, Augusts 2002
2. F. Asnicar, L. Chittaro, L. De Marco, L. Del Cano, R. Pugliese, R. Ranon, and A. Senerchia. The GridCC project, Distributed Cooperative Laboratories: Networking, Instrumentation, and Measurements, F. Davoli, S.Palazzo, and S. Zappatore, Eds. Springer New York, NY, pp. 279–294, 2006
3. L. Berruti, L. Caviglione, F. Davoli, M. Polizzi, S. Vignola, and S. Zappatore. On the integration of telecommunication measurement devices within the framework of an instrumentation grid. In Grid enabled remote instrumentation, signals and communication technology, F. Davoli, N. Meyer, R. Pugliese, and S. Zappatore, Eds. Springer New York, NY pp. 283–300 ISSN 1860-4862, ISBN 978-0-387-09662-9 (Print) 978-0-387-09663-6 (Online), 2008
4. L. Berruti, L. Denegri, and S. Zappatore. Wireless sensor networks and SNMP: Data publication over an IP network. Wireless Communications: 2007 CNIT Thyrrenian Symposium, S. Pupolin, Eds., IEEE South and Central Italy Section, IEEE Communications Society, pp. 295–208. Springer New York, NY, 2007
5. E. H. Callaway Jr. Wireless sensor networks: architectures and protocols. AUERBACH, Auerbach Publication, New York, NY, USA, August 2003
6. Cypress Perform. http://www.cypress.com/ 2010, accessed Sept. 30 2010
7. Ember. http://www.ember.com/ 2010, accessed Sept. 30 2010
8. E. M. Fouladvand, Z. Sadjadi, and R. M. Shaebani. Characteristics of vehicular traffic flow at a roundabout. Physical Review E, 70(4): 2004, DOI: 10.1103/PhysRevE.70.046132

9. gLite — EGEE:. http://glite.web.cern.ch/glite/ 2010, accessed Sept. 30 2010
10. F. Lelli, E. Frizziero, M. Gulmini, G. Maron, S. Orlando, A. Petrucci, and S. Squizzato. The many faces of the integration of instruments and the grid. International Journal of Web Grid Services, 3(3):239–266, 2007
11. F. Lelli and G. Maron. The GridCC project. Distributed cooperative laboratories: Networking, instrumentation, and measurements, F. Davoli, S.Palazzo, and S. Zappatore, Eds., Springer, New York, NY, pp. 269–277, 2006
12. M.J. Lighthill and G.B. Whitham. On kinematic waves II. A theory of traffic flow on long crowded roads. Proceedings of the Royal Society of London, A229:317–345, 1955.
13. G. Mulligan. The internet of things: Here now and coming soon. IEEE Internet Computing, 14(1):35–36, January-February 2010
14. K. Nagel and M. Schreckenberg. A cellular automaton model for freeway traffic. Journal de Physique I, 2(12):2221–2229, December 1992
15. M. Papageorgiou. Applications of automatic control concepts to traffic flow modeling and control. Lecture Notes in Control and Information Sciencies. Springer, Secaucus, NJ, USA, 1983
16. H.J. Payne. Models of freeway traffic and control. Simulation Council Proceedings, 1,51, 1971
17. Project DORII (Deployment Of Remote Instrumentation Infrastructure):. http://www.dorii.eu 2010, accessed Sept. 30 2010
18. Project GridCC (Grid Enabled Remote Instrumentation with Distributed Control and Computation). http://www.gridcc.org. 2010, accessed Sept. 30 2010
19. RINGrid: http://www.ringrid.eu/ 2010, accessed Sept. 30 2010
20. G.B. Whitham. Linear and nonlinear waves. Wiley, New York NY, 1974

Part V
Learning Environments

Networking Resources for Research and Scientific Education in BW-eLabs

Sabina Jeschke, Eckart Hauck, Michael Krüger, Wolfgang Osten, Olivier Pfeiffer, and Thomas Richter

Abstract The major aim of the BW-eLabs architecture (networked virtual and remote labs in Baden-Württemberg) is the expansion of access to heterogeneous experimental resources (remote as well as virtual or hybrid) for cooperative execution of experiments in natural sciences and engineering as well as the reuse of raw data and experiments for research and education purposes. Thus, three major tasks take center stage for the BW-eLabs portal: 1st, the creation of efficient possibilities of external access to local experimental surroundings, 2nd, the guarantee of transparency and reproducibility of experiments, and 3rd, the promotion of cooperation and collaboration within the scientific community in experiment-driven high-technology fields. Nanotechnology and robotics serve as demonstrator disciplines because especially in these cost intensive areas access to experimental equipment is an important prerequisite for ensuring access to professional tools for all scientific communities involved. Corresponding raw data and related documents are examined along their life cycle and embedded into the entire process chain of experimental environments through sustainable indexing and field specific ontologies, traceable and reusable by means of semantic search. Existing infrastructure, e.g. digital libraries, decentralized tools, and repositories, are embedded into the BW-eLabs. As a framework, the 3D platform Wonderland (Sun) comes into place, taking the complexity of professional experimental set-ups into account. The BW-eLabs portal, together with its partner projects LiLa and NetLabs, is designed as an open network for scientific data and experimental set-ups under OpenSource/Open Access/Open Content policy.

1 Introduction

The experiment is the essential research methodology component in natural sciences and engineering. Unfortunately, a contemporary experiment sometimes is

O. Pfeiffer (✉)
Technische Universität Berlin, Berlin, Germany
e-mail: pfeiffer@math.tu-berlin.de

F. Davoli et al. (eds.), *Remote Instrumentation Services on the e-Infrastructure*,
DOI 10.1007/978-1-4419-5574-6_20,

subject to several restrictions such as financial and spatial limitations, and missing support [1]. The new media offer two fundamental models to tackle these challenges: *virtual laboratories* [2] and *remote experiments* [3]. A virtual laboratory is an elaborated software application to simulate experiments on the computer. These multimedia-based and interactive experiments are realistic, virtual simulations of an experimental set-up. The benefits, for example, are that many researchers or students can simultaneously work on the same experimental set-up or conduct abstract non-realistic feasible experiments. A remote experiment is a real experiment remotely controlled from outside the actual laboratory. Such an experiment consists of two main components: the experimental set-up and the technique and software needed to facilitate the remote access. An advantage is the simplification of the access to experimental set-ups, independent from space, time, and security considerations.

1.1 Scientific Networking of Virtual Laboratories and Remote Experiments

Virtual laboratories and remote experiments are already used in many scientific research groups around the world. In addition, many existing experiments separated in laboratories could be easily equipped with remote access techniques. Regrettably, central access points (portals, repositories, etc.) do not exist yet to locate such useful experimental resources or information. This is an unsatisfactory situation for everyone involved, and therefore, a lot of experimental set-ups remain hidden. Moreover, collaboration between scientific research groups has to be furthered. To this end, an efficient networking between these groups would overcome challenges, e.g. to review experimental set-ups or enhance set-ups for better accuracy of results. Besides, seminal scientific studies do not emerge from isolated laboratories as individual achievements. It is vitally important to collaborate in a different kind of relation without organizational limits and geographical distances. At the same time, library and scientific document management systems, e.g. “eSciDoc”, do not contain links and information of those experiments that are explained in publications. Repeatability and transparency of results can hardly be ascertained in this way. Within this context the realization of scientific networking is a key factor in today’s information society.

1.2 Introduction to BW-eLabs

The applicability of virtual laboratories and remote experiments is not limited to single disciplines. As described above, contemporary experiments are subject to several restrictions. A lot of these experiments are expensive and/or consist of very complex set-ups. Some of these experiments are very susceptible or need special environmental conditions (dust-free, temperature, pressure, etc.). A prominent example is the field of nanotechnologies having a huge influence on the current technology development as a key technology of the twenty first century. Miniaturization of the

observed objects and effects and the requirements for the accuracy of the measurements themselves and the equipment are closely connected to tremendous financial challenges (e.g. for clean rooms, electron microscopes, susceptible spectrometers). Hence, because only a small group of well-subsidized institutions obtain access to this equipment we chose nanotechnologies as a prototype for the BW-eLabs [4, 5].

The BW-eLabs project provides a repository of the "second generation" to face the technical challenges of experimental natural sciences and engineering in the field of nanotechnology. We use the service-oriented eResearch environment "eSciDoc" developed by FIZ Karlsruhe that can include dynamic and changeable data recorded along the scientific workflow (e.g. informal documents, experimental results, documentation) and experiment itself. The infrastructure keeps experimental results and the required workflow sequences available for subsequent use such as research and education. The benefits are the better networking mechanisms of geographically distributed research groups to share results through the Internet. The former safeguards the reproducibility of the results by employing a post-connected "electronic laboratory journal" and therefore avoids repeated measurements by consequently processing existing results, the latter allows for the smooth exchange of laboratory facilities over distances, maximizing the usefulness of the corresponding equipment. To protect the results of research groups, all the material (raw data and equipment) is shielded by appropriate security policies and authentication via Sibboleth. Publications are an important means of information exchange in the scientific society. This can be realized by establishing the corresponding policies.

The main target audience of the BW-eLabs project are scientists. Another target group are graduated students of the related topics. Additionally, the infrastructure can be useful in academic education scenarios to conduct complex professional experiments within a lecture or seminar.

1.3 Infrastructure of the BW-eLabs

The infrastructure of the BW-eLabs Project consists of the following different components:

1. *System Structure*: The eResearch environment "eSciDoc" [6] used in the BW-eLabs comprises a set of services and solutions that facilitates user-inquires for static data to the corresponding repositories. It supports the development of suitable metadata concepts to describe experiments and raw data of the technical disciplines for documentation and repeatability purposes. In addition, different views on experiments are taken into account (experimenter's perspective, experiment perspective, room perspective), thus providing a semantic infrastructure for indexing special data (semantic search) [7].
2. *Digital Library-Static Content Provider*: Libraries of the participating universities store and manage static data (journals, proceedings, etc.) including metadata such as a content manager of a digital classical repository. This is the first step to integrate scientific documents into the virtual laboratory spaces [8].

3. *Experimental Content Provider:* The corresponding research groups and institutions act as experimental content managers providing virtual laboratories and remote experiments. The infrastructure will be tested in a pilot phase tailored to the requirements of the partners.
4. *Access-Rights-Management:* The infrastructure offers an adequate access-rights-management similar to open access and open content.
5. *Integration of Digital Holography:* By integrating the concepts of digital holography [9] real objects are visualized three-dimensionally in virtual spaces.
6. *Interfaces:* To facilitate inserting data and experiments and to keep the entrance barriers as low as possible, interfaces for software used in the laboratory routine are created in the project. These software applications are e.g. LabVIEW for remote- and process-control of laboratory equipment, computer-algebra-systems like Mathematica or numerical packages, e.g. Matlab. Suitable interfaces are created on the side of the content providers for integration in already existing search engines and library portals to make the obtained data accessible and archive them.
7. *Communication tool "Virtual World":* The "virtual world" is a simulated three-dimensional world providing the integration of available digital libraries and experiments, and a communication environment to promote collaboration between research groups. As a result, geographical distances between researchers are bridged and measurement data and experimental processes are archived. Integration of familiar tools and established infrastructure as well as intuitive system access and individualization procedures help to raise the acceptance and the usability of the portal.

2 Unique Features and State of Technology

2.1 Current State of Affairs in Baden-Württemberg and Germany in General

The design, development and implementation of virtual knowledge spaces is being pursued on the national level throughout Germany, in particular at the University of Paderborn (sTeam [10, 11]), as well as CURE [12, 13] at the Distance University of Hagen (Fernuniversität Hagen). However, despite years of extensive research and many visionary ideas, the field has yet to to come to fruition. This challenge is in no small measure the result of shortcomings in the existing base technologies and infrastructure required for the envisioned scenarios.

Until now, most development fell into one of two categories: either isolated computer-based laboratory environments, each applicable to a few specific experiments, or portal technologies for the integration of diverse remote experiments and virtual labs. However, since the actual labs were designed for quite specific purposes, the lack of a coherent interface architecture prevented their content-oriented combination into more complex research scenarios.

2.2 Deployment of a 3D-Engine for Improved Usability

The complexity of such extended or combined scenarios requires new concepts, posing further challenges not only for the IT-technologies involved but also in the field of usability. Intuitive access and operation of the system are vital to increase user-acceptance. To achieve this goal it is necessary to transfer real-world utilization concepts into the virtual world of the lab. In addition, this will reinforce the concept of the virtual lab or remote experiment as it trains the operation of real-life equipment.

Vital input has come from research in the areas of virtual environments and gaming engines. Combined with developments in web service technologies, service-oriented architectures, open 3D-Engines, social web and the digital extensions of libraries, this should provide the technological base for virtual knowledge spaces [14]. A suitable infrastructure is the open source three-dimensional platform "Wonderland" [15], developed by Sun, to be completed by suitable interfaces.

2.3 Document-Management Systems for Scientific Content

Extending the eSciDoc infrastructure by incorporating these technological advances supports the publication, visualization and management of the finished objects as well as their continuous editing and use. The system is capable of handling text-based document at various stages of completion as well as primary (raw) research data from a wide scope of scientific disciplines. As a result, eSciDoc is well-suited to fulfill the knowledge and information management requirements of universities or other research or R&D related institutions, both public and private.

The eSciDoc system goes beyond the capabilities of the widely deployed Dspace from MIT (Cambridge, USA, www.dspace.org), Eprints[1] of the University of Southampton or the repository system OPUS,[2] which is widely used in Germany and was developed at the University of Stuttgart (cf. A.1). All of these systems were custom-tailored to the requirements of an institutional publication repository, a limitation reflected in their user interfaces.

The eSciDoc infrastructure represents a generic infrastructure incorporating aspects of data quality, data management and long term availability. eSciDoc is based on the repository-software Fedora,[3] and enriched by a series of newly developed services. The system provides features such as persistent referencing by assignment of appropriate identifiers, automatic extraction of technical metadata, and the administration of objects with heterogeneous metadata models.

The system is designed to track and map the complete life cycle of a scientific document/data object, from its first inception to final publication and archiving.

[1] http://www.eprints.org

[2] http://elib.uni-stuttgart.de/opus/doku/about.php

[3] http://www.fedora-commons.org

This is achieved by combining the concept of revision control with the mapping of formal and semantic relations between objects. By connecting data and documents/publications and by making them available to users, eSciDoc acts as an "enabling technology" for the comprehensibleness and the possibility to re-use research results required in scientific collaborations. Current solutions addressing these challenges only exist isolated. One example of an existing solution is the "Archer"[4] system, promoted by the Australian Ministry of Innovation, Industry, Science and Research. It provides a software platform for managing and documenting the research data of Australian universities.

Similarly to an existing set of applications developed for the humanities at the Max Planck Society, the eSciDoc infrastructure is suitable to be extended to a "scientific workbench", supporting cooperation in virtual, distributed workgroups. The BW-eLabs will incorporate methods of data management in the natural sciences [16, 17] in co-operation with the scientific data processing service-group (Servicegruppe Wissenschaftliche Informationsverarbeitung) of the Freiburg Materials Research Center (FMF) at the University of Freiburg. Early implementations will address scenarios posed by the FMF, ranging from the characterization of the chemical analysis with high throughput technologies and the development of electronic laboratory journals to scientific information repositories for investigating individual technical questions.

The FMF has developed methods for systematically managing analogue and electronic scientific information in the form of laboratory journal entries and primary data files in preliminary work, covering some basic aspects of scientific data managements [18]. Based on the FMF-developed text-based full metadata format, these concepts permit the direct addition of experimental parameters and measured variables to the actual data, exactly as it is common in everyday work (e.g. in laboratory journals or measurement records), mapping the standard work-processes of the prevalent mode of operation.

The BW-eLabs also address the complex problem of retrieving primary data: in difference to standard bibliographic indices, primary data catalogues have to support indexing by relevant experimental parameters. The BW-eLabs introduce the concept of using feature vectors consisting of experimental variables as a new classifier. The concept of feature vectors was first explored in the fields of pattern recognition and machine learning. Complex objects are described as n-tuples of real numbers, facilitating semantic search algorithms based on custom-tailored distance measures within this n-dimensional vector space of primary data sets.

2.4 Remote Experiments with Real Objects using Digital Holography

Holograms possess a unique property: they contain the complete optical information (at the wavelength of the illuminating laser) of the recorded object. As a result,

[4] Australian ResearCH Enabling Environment, http://archer.edu.au/

holograms can replace "real" objects in a number of experiments where purely optical properties about these objects are sufficient. In digital holography the wavefront resulting from the interference of light diffracted by the object and an undisturbed reference beam is recorded digitally, either directly from a CCD sensor or by digitizing an existing analogue hologram from film. This digital hologram can be reconstructed in one of two possible ways: by illuminating a suitable light modulator displaying the recorded hologram (for example the LCD-panel of a digital projector), or numerically in the computer. The former is basically identical to the classic optical reconstruction of an analogue hologram. In contrast, the latter method provides new possibilities not available in analogue holography. Numerical reconstruction does not require the exact replication of the original set-up to reconstruct an object, a severe limitation in sharing analogue holograms. A digital hologram can be stored on a computer or distributed over the internet, providing researchers in distributed collaboration access to the "real" object (or, at least, its optical properties). Current challenges in digital holography are the miniaturization of the recording devices, the robustness of the method and a simultaneous increase in measuring accuracy.

Short coherence digital holography allows for layered scans of a three-dimensional object (i.e. in depth, not just the surface of the object), enabling the reconstruction of layers of up to 20 μm thickness. This technique is used in microscopy and endoscopic medical procedures. It is capable of deducing the overall composition of an object by detecting areas of different elasticity using digital holography, in effect providing a surgeon with a sense of touch in endoscopic procedures.

Comparative holography is a technique that compares nominally identical but physically different objects (sample-probe-comparison). By using digital holograms this method can be extended beyond the classic scope requiring an exact replication of the original set-up. As a result, it provides a very precise (fractions of the wavelength of the used laser), flexible testing method not requiring the simultaneous physical presence of the sample and the probe for comparison. The recorded hologram of the sample is reconstructed optically by illuminating a suitable light modulator. The reconstructed wavefront is used to illuminate the probe together with a reference wave. The resulting interference wavefront is recorded as a second digital hologram or simply viewed directly. Numerical reconstruction (or visual inspection of the probe) will show the interference phase between the wavefronts, the difference in the surface of the probe and the sample. This allows for real time inspection of objects revealing deformations of the surface on the scale of the laser wavelength, well suited for quality assurance (Fig. 1).

3 Demonstrator Scenario Nanotechnology

Nanotechnology deals with the synthesis, the properties, the characterization and the use of materials, on a scale between 1 and100 nm. Moreover, it is considered to be the key technology of the twenty first century. During the last 15 years, the nanotechnology developed more and more into a highly multi-disciplinary science.

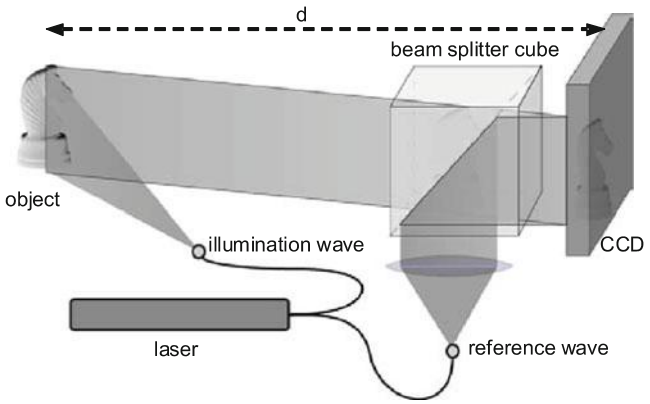


Fig. 1 Schematic set-up for recording a digital hologram onto CCD

Its own Bachelor and master courses of studies have been established at universities and technical colleges. The high-grade multi-disciplinarity presupposes an active exchange and overlap between chemistry, physics, material sciences and even biology, if essential. The multiplicity of different scientific approaches and methods stimulates and requires collective experimenting, evaluating and publishing of completely diverse working groups necessary for the accomplishment of the research. This led to the installation and establishment of larger scientific centers and networks focused on nanotechnology in universities worldwide. Some examples are the German Center for Functional Nanostructures (CFN) at the University of Karlsruhe, together with Forschungszentrum Karlsruhe, the planned German Center for Nanotechnology at the University of Würzburg, the Swiss Nanoscience Institute (SNI), established at the University of Basel as a Swiss competence centre in the field of nanotechnology. The acquisition and shared use of expensive giant equipment in particular could be realized by the joint efforts of some of these centers. Due to the multi-disciplinarity and the necessity of networking, nanotechnological subjects are ideally suited for the application of virtual laboratories and remote experiments. In the context of BW-eLabs, these experiments can be cross-linked in order to establish new standards in the field of knowledge management.

The FMF at the University of Freiburg, Germany, analyzes synthesis, characterization and application of semiconducting and metallic nano-particles. Their insights are made accessible online/remotely as part of the BW-eLabs project made available for further transfer and application scenarios:

The physical and chemical parameters relevant for the synthesis of nano-particles were evaluated exemplarily on the basis of the synthesis of high-grade luminescent CdSe nano-crystals [19, 20]. This synthesis is now being established in the context of the BW-eLabs project as a remote-enabled model synthesis. The parameters determined in the conventional reaction piston are transferred to microwave synthesis equipment, in which all decisive synthesis-parameters, e.g. temperature (profile)

and concentration of reactants, can be standardized and are controlled and monitored remotely.

The reaction process is amended by appropriate on-line analytics. By means of absorption- and emission-spectroscopy the synthesis process is monitored and controlled and can be compared to previously accomplished syntheses. Corresponding spectra of this on-line analytics are recorded and logged automatically. Alongside with this on-line analysis (small analysis) more complex analysis methods are connected in a later stage, e.g. connecting the transmission electron microscopy (large analysis) from another remote laboratory with the goal of a common structured data-management comprising a structured and clear representation of the results in form of an electronic laboratory journal. After the synthesis setup is concluded and the remote-ability of the synthesis process as well as appertaining analytics have been tested, further material syntheses are standardized and transferred to microwave basis. Gold nanoparticles are sought out to be the following model system and contacts to the department of inorganic chemistry of the University of Freiburg already exist on the basis of a common project for the synthesis of metalliferous nanoparticles. In addition, external partners can enter on this level and standardize their systems, accordingly. Thus, we develop a material synthesis library which is made accessible to other groups and contributes to the objectives of the BW-eLabs by securing the compiled research results of a working group beyond the period of the presence of a specific researcher. Moreover, this library serves as a point of contact for advanced research on the corresponding materials. Furthermore, virtual laboratories can be connected at this time, using the generated data for semi-empiric modeling [21] (structural characteristic statements of materials) and also provide input for the synthesis, structure and characteristics of new materials by using ab initio methods [22]. This forwards and accelerates tailored material development and permits the integration of nanomaterials with special characteristics into new applications.

One example for the mentioned model systems CdSe nanocrystals and Au nanoparticles is the systematic introduction of impurity atoms (doping) of nanocrystals [23] which opens new perspectives for the development of IR lasers or IR to emitting pigments to be used for bio marking. Then, nanohybrid materials, e.g. nanoscopic intermetallic phases as Au@Pt [24] or Pt/Bi [25, 26] compounds can be developed as catalysts for reactions with a high industrial application potential.

The responsible group at the FMF is working within the area of theoretical modeling of nanoclusters and nanocatalysts. Their results will be integrated at an appropriate time as virtual laboratory. Another applied component is the integration of nanomaterials in well-defined superimposed macro-structures. This is another area to integrate further remote experiments and virtual laboratories, especially in the field of nanostructuring of materials (remote experiment) and theoretical physics of complex systems (virtual laboratory).

Synthesis as well as analysis and application of nanomaterials interact with the virtual laboratories of theoretical research groups. Figure 5 is outlining the linkage of a remote synthesis experiment with other remote laboratories, potential applications and virtual laboratories (Fig. 2).

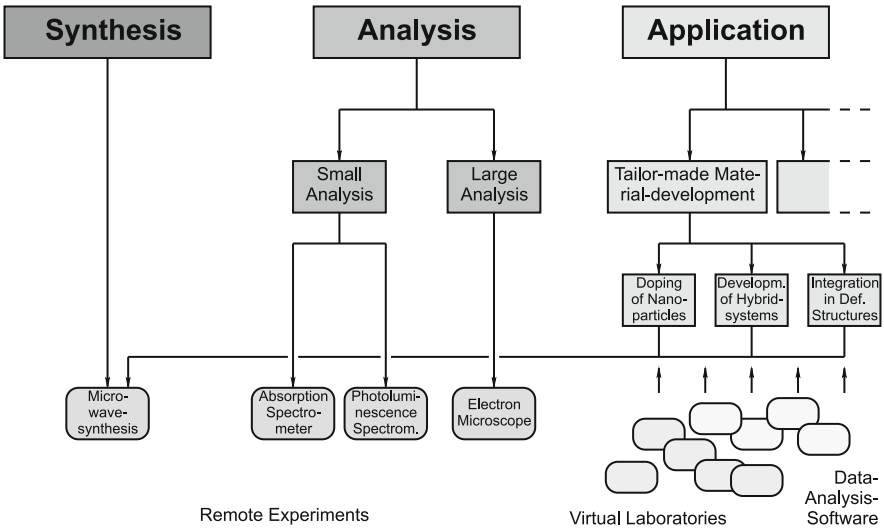


Fig. 2 BW-eLabs sample architecture

An interactive platform linking diverse remote and virtual laboratories features sustainable long-term scientific advantages:

- A more efficient and systematic development of new materials is promoted.
- Compiled knowledge is indexed and secured on a long-term basis. This is of particular importance as once developed syntheses can by a successful standardization be made accessible to researches whose primary interests are, for example, , physical measurements of that special material rather than its synthesis.
- Constantly obtainable material quality that would be independent of the synthesizing chemist permits a continuously reliable research on this material. This item is of particular importance since syntheses are usually developed by graduate students and/or Postdocs who only stay at their respective research institute for a limited period and knowledge transfer does not always take the prominent role that it should.
- Since only well-functioning syntheses are standardizable, some kind of quality control for the synthesis directives is automatically integrated. Thus, standardization of the synthesis will also be an important section of challenge within a synthetic PhD or master's thesis in the future.
- An interactive platform also facilitates the access to research methods that are not available locally, because of missing giant equipment, special devices, or particularly developed research equipment. Similarly as in internet chat rooms, extended ways of contact are possible on the scientific level which would not have come about in any other way (promotion of the development of national and international contacts).

- An independently generated electronic laboratory journal has the advantage that all results together with the necessary parameters and all steps (synthesis ⇒ characterization ⇒ application) are retained uniformly and clearly arranged. Integratability into appropriate data bases prevents unnecessary parallel research and the amount of work for data evaluation is reduced for each researcher. In addition, research projects that have been classified as not successful by rule of a student's thumb and therefore remained unevaluated are archived. At second glance, an expert might possibly come to a surprising discovery (increase the ratio of innovative random discoveries).
- Established synthesis approaches are used also as exemplary sample experiments in teachings.

4 Demonstrator Scenario Robotics

Robotics is one of the key technologies present in the twenty first century. The essential research areas can be classified in: perception, actuator engineering, apperception and decision making. In terms of perception, the question of sensor technology and sensor combination arises. In the area of actuator engineering, manipulators and movement systems of different areas of application are decisive. With respect to apperception and decision making, the central points are data processing and filtering of knowledge representations, along with path finding and decisive derivation.

Due to the accessibility of the different robotics applications and by means of evaluating any attempts already made in the decision making phase on an identical platform, new trials are tested faster and at a reasonable price. The environments where these systems can be examined in both a simulative as well as a realistic manner are manifold – two areas of application protrude in the context of the BW-eLabs/NETLABS project:

4.1 Cooperative Scenarios

The availability for use of robotics systems in production environment continues to rise steadily. The necessary cooperation scenarios (between robots or between human and machine) lead to novel research questions that have yet to be answered (Fig. 3).

Further investigations however – in the case of realistic scenarios, e.g. with a collaborating installation – remain financeable only for a few research teams. Within the investigation of cooperative scenarios in robotics, the problem of cost-intenseness becomes obviously even more important due to the necessarily higher number of robots. Additionally, investigation for "cooperative behaviour" is only to some extend depending on the hardware – rather, concepts of awareness, decision making processes and task sharing play a central role. To investigate these research

Fig. 3 Robocup-team 2009, TU dortmund

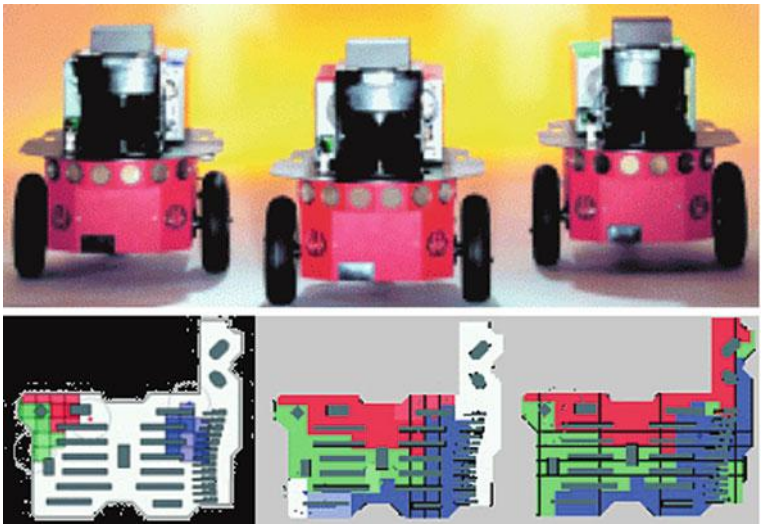


Fig. 4 Team of cleaning machines (SINAS), *below*: map of task distribution

goals in a general and transferable manner, it is important to dispose of a large number of different robots in different application scenarios (Fig. 4).

By means of suitable remote laboratories offered in BW-eLabs/NETLABS, production based research – also interdisciplinary – is carried out in multi-robot-applications and in human-machine cooperation scenarios. Any gained primary data is made available to other institutes.

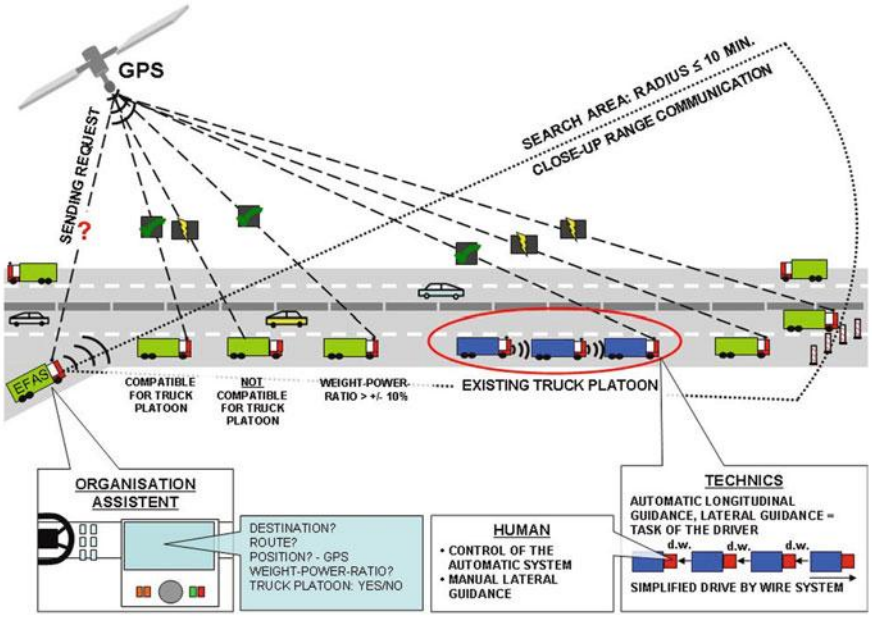


Fig. 5 Driver organized truck platoons [29]

4.2 Autonomous Vehicles

Since 1977, autonomous vehicles (a.k.a. driverless cars) have played a more and more important role in traffic technology. One opportunity to manage the increase of freight transportation and to optimize utilization of motorway capacities is the concept of truck platoons. With the aid of Advanced Driver Assistance Systems, trucks are electronically coupled keeping very short gaps (approx. 10 m) to form truck platoons on motorways [27]. This contributes to the optimization of traffic flow and reduction of fuel consumption advantaged by slipstream driving [28] (Fig. 5).

Other concept and application scenarios of partly autonomous and fully autonomous cars are known. Within the DARPA Grand Challenge Cup, for example, different concepts of fully autonomous vehicles in substantial off-road courses are tested (Fig. 6).

BW-eLabs/NETLABS integrates movement-supported simulators in the form of virtual laboratories that allow for testing of passenger reaction, recognition of hazards, and even help to simulate path finding algorithms. Expensive, cost-intensive environments are linked and made accessible to other research teams. The simulators available within an institute are extended by means of functionalities, and thus allow for external control.

In these areas of application, security plays a decisive role. This aspect must be taken into consideration when extending remote laboratories during the course of the BW-eLabs/NETLABS project. In addition, measures must be taken to examine

Fig. 6 Spirit of Berlin, FU Berlin, Team Rojas [30]

how hazards for both humans and material can be prevented in remote-controlled laboratories. Aspects such as working and IT security must be taken into consideration. This is especially significant in human- machine-cooperation scenarios. For this case concepts are developed to guarantee the safety of the person interacting with a remote-controlled machine.

In addition, robotics acts as a "connection discipline" within this application: actuators that control set-ups/experiments are necessary for remote-controlled laboratories. The aim to be achieved in the long run is a reconfiguration of laboratory environments. This will lead to a higher flexibility. Moreover, an infrastructure of intelligent "robots", or better said "robot arms" has to be created to control experiments and laboratories. Additional concepts are developed, prototypically implemented and fitted accordingly to BW-eLabs/NETLABS.

5 Acceptance and Usability

Usability and serviceability are decisive for the acceptance of the BW-eLabs. The usefulness has to prove itself exclusively by the increase in value it presents to researchers. The design of the user interface and interaction concepts are crucial for its operability. We will now address the latter in detail.

Both the complexity of the planned research scenarios and the complexity of the planned overall system, exceeding the possibilities of classical repositories, which realize isolated access to static data or to special isolated laboratory environments, pose unique demands towards the interaction concepts and techniques to be used. Scientists' creativity is promoted by a clear visualization of the data relevant for the experiments and is not restrained by difficulties with the interaction. Therefore

complexity reduction is necessary in many application scenarios that, however, must not lead to a restriction of possibilities of use.

Studies in psychology of perception of the last years showed that the use of close-to-reality metaphors facilitates a particularly intuitive use of software applications. A particularly complex environment as envisioned for the BW-eLabs must therefore most favorably be realized in three-dimensional representation. Such a communication environment in form of a virtual three-dimensional world on the one hand facilitates the conversion of interaction concepts that are already well-known from the real world and the integration and development of completely new approaches. On the other hand, this virtual world serves as a platform for the development and testing of new techniques of visualization of information.

At the same time, mechanisms of information-based data analysis, i.e. processing of experimental data with automatic consideration of all meta-information like physical units, metric, etc. are employed. The preparatory work of the FMF in this area already shows that this approach leads to an effective complexity reduction, so that experimentally working scientists can test-drive, develop and implement complex data analysis algorithms.

Effectiveness and efficiency of different interaction concepts and their acceptance by content providers and by the users is to be evaluated in usability tests with test subjects. With the help of the expertise won thereby, interaction concepts that encounter the broadest acceptance rate in the addressed target group are to be developed and implemented. It should in particular be examined which forms of social presence in the virtual world are essential for collaborative scenarios to be accepted by male and female scientists.

These tests will be conducted in the usability laboratory at the Stuttgart Media University that is equipped with the appropriate devices for data acquisition and analysis.

6 Roadmap

Right from the beginning, mechanisms for distributed authentication (shibboleth, and single sign on) are incorporated and usability studies for the 3D-interface are arranged. Meta data profiles and concepts for integrating digital content into intelligent web portals are developed. First concepts for the searchability of primary data are tested exemplary in the BW-eLabs.

During the starting year of the project the 3D-infrastructure is established, embedding the already existing virtual laboratories and determining the intial connection to the eResearch environment. To this end the important components such as experiments, their life cycle, groups, and so on are defined, open standards are evaluated, and the first prototype of a remotely-controllable laboratory is implemented. Additionally, models of the implementation of digital holography into BW-eLabs are examined and evaluated.

The second year is devoted to the extension and the development of the infrastructure, due to the newly collected requirements, and to the development of remote

experiments as well as to the implementation of the usability studies. The first nanotechnological virtual laboratories and remote experiments are integrated and the entire process of data generation, filing, and access is reproduced and evaluated. In order to visualize real objects, initial methods of digital holography are included. A first prototype to semantic searchability of primary raw data is prepared and existing information resources are connected. For that purpose, a sophisticated policy-management for the integration of licensed data and open access concepts is prepared in co-operation with the university libraries and the partners. Evaluations of test users will take place.

The third year is dedicated to consolidating services and infrastructure so that they can be sustainably integrated into production systems. By incorporating more institutions, additional scenarios are ascertained and if necessary further components will be supplied. Activities to widen the user-base that began in the starting year as well as community-building actions are strengthened (conferences, workshops, etc.). Technical support documents are provided to facilitate the integration of further experiments. Based on the evaluation, a plan is provided, not only for securing the results, but also containing a business model for forthcoming developments.

References

1. S. Jeschke, S. Cikic, Ludwig, N. Sinha, U. and C. Thomsen. Virtual and remote experiments in cooperative knowledge spaces, in grid enabled remote instrumentation, Part IV: Grid resource allocation, QoS, and security, F. Davoli, N. Meyer, R. Pugliese, and S. Zappatore., (eds), DOI 10.1007/978-0-387-09663-6_22, pp. 329–343, Springer, New York, NY, 2009
2. S. Jeschke, N. Natho, T. Richter. Integration technology for virtual laboratories – individualization and flexibility through computer algebra systems. Proceedings of the 2nd International Conference on Engineering Education & Training – Preparing engineers to meet the challenges of the future (ICEET – 2), April 9–11, Kuwait, 2007
3. S. Jeschke, T. Richter, H. Scheel, and C. Thomsen. On remote and virtual experiments in e-Learning in statistical mechanics and thermodynamics. Journal of Software (JSW), 6(2), 76–85, 2007
4. S. Jeschke, O. Pfeiffer, and C. Thomsen. Vernetzung experimenteller ressourcen in forschung und ausbildung für nanotechnologien und nanowissenschaft In: INFORMATIK 2006, Informatik für Menschen, Band 1, Beiträge der 36. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 2.–6. Oktober 2006 in Dresden, C. Hochberger and R. Liskowsky. (eds), pp 85–89, 2006, Gesellschaft für Informatik, Bonn, 2006
5. S. Jeschke, N. Natho, O. Pfeiffer, and C. Thomsen. Networking resources for research and scientific education in nanoscience and nanotechnologies, Proceedings of the 2008 International Conference on Nanoscience and Nanotechnology. Casual Productions, Melbourne, VIC, IEEE Press, New York, NY, pp. 234–237, 2008
6. eSciDoc Project webpage, http://www.escidoc.org/, last access 31.3.2010 (2010)
7. M. Riede, R. Schueppel, K.O. Sylvester-Hvid, M. Kuhne, M.C. Rottger, K. Zimmermann, and A.W. Liehr. On the communication of scientific data: The full-metadata format, Computer Physics Communications, 181(3), March 2010, 651–662, ISSN 0010-4655, DOI: 10.1016/j.cpc.2009.11.014, http://www.sciencedirect.com/science/article/B6TJ5-4XV06XY-1/2/2f48f25ec4ab09a1fff899dfe322c018)

8. W. Stephan, B. Graf, A. Kirchgässner, S. Laschat, W. Schüz. Das Zeitschriften-Paradoxon oder: Wer verfügt über wissenschaftliche Information? Universitätsbibliothek Stuttgart, Bibliothek Bildung und Fortschritt 3, 2004
9. W. Osten, P. Ferraro. Digital holography for the inspection of microsystems. W. Osten., (ed), Optical inspection of microsystems, pp. 351–426, CRC Taylor & Francis, Boca Raton, FL, 2006
10. B. Eßmann, P. Bleckmann, T. Hampel, and R. Elsässer. Distributed persistence in CSCW applications. tr-ri-05-268, December 2005
11. B. Eßmann, T. Hampel, F. Götz, and A. Elsner. Embedding collaborative visualizations into virtual knowledge spaces. Proccedings of the 7th International Conference on the Design of Cooperative Systems, pp. 33–40, Carry-le-Rouet, 2006
12. A. Haake. CURE Das CSCL-Portal der FernUniversität in Hagen – Benutzungshandbuch. Hagen, Germany: FernUniversität Gesamthochschule. http://teamwork.fernuni-hagen.de/CURE/doc/manual.pdf , last access 31.3.2010), 2005
13. J.M. Haake, T. Schümmer, A. Haake, M. Bourimi, and B. Landgraf. Supporting flexible collaborative distance learning in the cure platform. Proceedings of the Hawaii International Conference on System Sciences (HICSS-37). Washington, DC, USA,. http://teamwork.fernuni-hagen.de/CURE/, last access 31.3.2010, IEEE Press, 2004
14. F. Scholze, W. Stephan. Retrieval on the grid. Results from the European Project GRACE (Grid Search and Categorization Engine) Zugang zum Fachwissen – ODOK '05, pp. 118–127, Feldkirch: Neugebauer , 2006
15. Wonderland (Sun) Project webpage, https://lg3d-wonderland.dev.java.net/, last access 6.8.2009
16. G. Schneider. Identity management in der praxis, (together with D. v. Suchodoletz), In: Lecture notes in informatics vol P-73, Jan v. Knop et al. (eds), pp. 255–264, Springer, Berlin, 2005
17. G. Schneider. The black-forest-grid-initiative (together with R. Backofen, H.-G- Borrmann, W. Deck, A. Dedner, L. de Raedt, K. Desch, M. Diesmann, M. Geier, A. Greiner, W.R. Hess, J. Honerkamp, St. Jankowski, I. Krossing, A.W. Liehr, A. Karwath, R. Klöfkorn, R. Pesché, T. Potjans, M.C. Röttger, L. Schmidt-Thieme, B. Voß, B. Weibelt, P. Wienemann, V.-H- Winterer. PIK, 29, , 81–87, 2006
18. G. Schneider. Neue Medien als strategische Schrittmacher an der Universität Freiburg, in: Neue Medien als strategische Schrittmacher an der Universität Freiburg, Universitätsbibliothek Freiburg, 2007, pp. 13–24, ISBN 978-3-928969-20-8, http://www.freidok.uni-freiburg.de/volltexte/3228/pdf/Neue_Medien_Universitaet_Freiburg.pdf, last access 31.3.2010, (2007)
19. F.-S. Riehle, R. Bienert, R. Thomann, G.A. Urban, M. Krüger. Blue luminescence and superstructures from magic size clusters of CdSe, Nano Letters, 9(2), 514–518, http://pubs.acs.org/doi/abs/10.1021/nl080150o, last retrieved 31.03.2010, (2009)
20. P.-J. Wu, K.-D. Tsuei, K.-H. Wei, K.S. Liang. Energy shift of photoemission spectra for organics-passivated CdSe nanoparticles: The final state effect, Solid State Communications 2007, 141(1), 6–11, 2007
21. Y. Yuan, F.-S. Riehle, H. Gu, R. Thomann, G.A. Urban, M. Krüger. Journal of Nanoscience and Nanotechnolgy, accepted (2009)
22. M.L. del Puerto, M.L. Tiago, J.R. Chelikowsky. Ab initio methods for the optical properties of CdSe Clusters, Physical Review B, 2008, 77, 045404 ,2008
23. D.J. Norris, A.L. Efros, S.C. Erwin. Doped nanocrystals, Science 2008, 319, 1776–1779, 2008
24. L. Yang, J. Chen, X. Zhong, K. Cui, Y. Xu, Y. Kuang. Au@Pt nanoparticles prepared by one-phase protocol and their electrocatalytic properties for methanol oxidation; Colloids and Surfaces A, 2007, 295, 21–26, 2007
25. L. Demarconnay, C. Coutanceau, J.-M. Leger. Oxygen electroreduction at nanostructured PtBi catalysts in alkaline medium, Electrochimica Acta, 2008, 53, 3232–3241, 2008
26. R.H. Rich, J.S. Hauger, E.G. Derrick. The role of the national science foundation in supporting advanced network infrastructure: Views of the research community, report of the

AAAS, October 1999, available online at: http://www.aaas.org/spp/rcp/netpolicy/reporth.htm, last access 31.3.2010, (1999)
27. A. Friedrichs, P. Meisen, Henning, K. A generic software architecture for a driver information system to organize and operate truck platoons. In: Vortragsveröffentlichung, International Conference on Heavy Vehicles (HHVT2008), Paris, 19–22.05.2008, (2008)
28. P. Meisen, K. Henning, T. Seidl. A data-mining technique for the planning and organization of truck platoons. In: Proceedings of the International. Conference on Heavy Vehicles. Paris, 2008
29. K. Henning, E. Preuschoff., (eds), Einsatzszenarien für Fahrerassistenzsysteme im Strassengueterverkehr und deren Bewertung. VDI, Reihe 12, Nr. 531, Düsseldorf, 2003
30. Spirit of Berlin, FU Berlin, Team Raúl Rojas, http://robotics.mi.fu-berlin.de/pmwiki/, last access 31.03.2010

"The SIP Pod" – A VoIP Student Lab/Playground

Máté J. Csorba, Prajwalan Karanjit, and Steinar H. Andresen

Abstract Student laboratory work is usually carried out within the university premises. In our lab, students are required to familiarise themselves with the principles of SIP entities and protocols, to configure and manage SIP UAs and a (virtual) proxy. The services of the system and also the management functions should be accessed not only at the campus, but also in an environment that encompasses the area of "Wireless Trondheim" (the downtown of Trondheim) and the student living quarters. A range of functions, from the most basic ones to location based routing for (fictive) emergency calls are to be demonstrated. The lab is built on open source software and products and it is a part of our course TTM4130 "Service Intelligence and Mobility". We believe that the exercises will foster creativity and enthusiasm among our students. The program framework of the lab can be made available to interested parties.

1 Background/Main Philosophy

At our department we have lectured networking principles and also software engineering/programming of real time systems in a number of years.[1] The lab and exercises have mainly been carried out at the university premises. The course "Service Intelligence and Mobility" deals with signalling systems, service control in modern networks, and also gives an introduction to mobility issues. It is presented to the students in their 6th semester (in a 5 year/10. semester integrated MSc study plan). The course originally (in 2000) emphasized IN (Intelligent Networks) architecture and principles, but has evolved to use the IMS/NGN architecture as "the

M.J. Csorba (✉)
Department of Telematics, Norwegian University of Science and Technology, N-7491 Trondheim, Norway
e-mail: csorba@item.ntnu.no

[1] A"pod" may mean a social group of whales, dolphins, etc. As we have named our SIP agent "ORCA", the notion of a SIP pod reflects a social playing ground of different instances of this agent.

F. Davoli et al. (eds.), *Remote Instrumentation Services on the e-Infrastructure*,
DOI 10.1007/978-1-4419-5574-6_21,

grand example" in recent years. In this period it has also become clear that SIP (the Session Initiation Protocol) is going to play a very important role in future systems. The course has had a mandatory lab exercise that should familiarize the students with SIP. This exercise was carried out on equipment that was configured and prepared in a lab. Typically 8 pairs of students where working in the lab simultaneously, for half a day, with the exercise.

However, considering that nearly every student today has his/her own PC/laptop and that SIP UAs may be easily configured, it was decided to try to run a more extensive set of exercises in the course, extending the lab to the students living premises and also to places out in town. This last opportunity is facilitated by an initiative, "Wireless Trondheim" that now covers most of the city of Trondheim.[2]

With these opportunities given, we are implementing a set of exercises that can be carried out and demonstrated outside the campus (out in town, even in the homes of friends and family).

The features, it is thought, will make it more fun for the students, as they can demonstrate/show to others what they are doing. It is also assumed that this environment will foster a much higher degree of creativity, thus further enhancing the interest and enthusiasm within the area. Further, we hope that the knowledge and familiarity obtained will be a good background for activities throughout the remaining 2 years of their study.

Our lab scheme and the program/system environment are sketched below. The exercise will be run under this new scheme for the first time in spring semester 2009. "The project" is run as an open source project, and interested parties (other universities) that may be interested are invited to join and share the experiences.

In this lab we emphasis SIP as a signalling protocol for VoIP. However as the protocol and the scheme is quite general, and already is supported by many development tools and methods, it of course can be developed to cover other purposes too.

2 Procedure/Work Description

The major goal of the laboratory work and the semester assignment for the students taking part in the course is to get acquainted with distributed telecom services from multiple aspects. There are multiple laboratory exercises during which students are required to install and configure a SIP proxy on their dedicated server workstation running Linux. After obtaining a working proxy instance the students may test it with some proprietary SIP User Agents (UAs) such as SJPhone or XLite. The students are grouped into pairs and have to enable multi-domain support for their proxies to be able to conduct SIP calls not only between the members of a pair but between users of the other domains established by the different groups. During the first laboratory session students gain insight into the modular setup of

[2]The on campus network as well as Wireless Trondheim adheres to the principles of the "Eduroam" initiative, which assures open ports for the SIP protocol.

the open-source SIP proxy openSIPS [1]. They install and configure the necessary modules that will allow them to have, among others, multi-domain support, user administration using MySQL databases, presence service, etc. Their setup is tested by establishing SIP sessions between users they create and register using SIP UAs. The test traffic that is created during the exercise is monitored using network traffic monitoring tools and as part of the exercise the students have to draw and comment Message Sequence Charts (MSCs) to show their understanding of various SIP sessions.

After obtaining basic operational state of the proxy each group studies Electronic Numbering (ENUM), as defined in [2]. Adding the capability of ENUM translation to the proxy is a relatively easy task, after which students are required to test their configuration with internationally available and free ENUM test numbers. During this task it is also required to capture the messages exchanged and visualize them in MSCs. In addition to the freely available numbers the students have to experiment with ENUM numbers configured by them. For this purpose a Domain Name Server (DNS) is installed in the laboratory, which has a configuration that can be extended by each group of students. This task teaches the students how to use Naming Authority Pointer (NAPTR) records [3] by requiring them to add a record to the configuration of the DNS server. Usually, this means that each of them adds his or her cellular number to the database this way establishing redirection of their own cellular number to one of the registered SIP clients within the laboratory if a the number is used as callee address in a UA. In the subsequent tasks the students have to take a deeper insight into the configuration and the message routing mechanisms of the proxy. This is tested by first printing out some of the message headers from SIP sessions routed by the proxy assigned to the group, second by diverting SIP calls based on given decision criteria. For example, criteria based on caller information subtracted from the SIP header diverting the call to specific endpoints based on database entries.

In the second part of the laboratory exercises the students get insight into the mechanics of a SIP UA by developing basic functionalities based on the JAIN SIP API [4]. Participants of the exercise are given a skeleton of a SIP UA that is built upon classes provided by JAIN SIP. The first task is to build a simple program that can send and receive SIP queries and responses and it is required to be able to register itself with the proxy as a first step. For this purpose message factories can be used provided by the UA skeleton. The groups continue then with implementing the SIP call establishment protocol to be able to talk to other UAs using their client application. In this case the capability of establishing calls is tested with proprietary UAs and that of developed by other groups of students.

During the semester attendees of the course have to submit a term project work also. The project work lasts approximately 2 weeks, during which students solve a series of tasks at home. These tasks, upon completion, result in a simple SIP-based communication network. In the project work phase the students are expected to understand the code and the basic classes and the event-driven state machine architecture they are provided with. One of the main goals is to implement a softphone extended with presence support. This involves implementing the mechanisms

handling the SIP messages PUBLISH, SUBSCRIBE and NOTIFY. In addition, supplementary features can be implemented by the students such as a contacts list for the UA. It is important to note that students are allowed to make modifications to the template in many ways, thus there is no limit to what extent the groups can get involved with the soft-phone and there is space for creativity in this assignment.

3 Laboratory Setup

The laboratory and the exercises are designed to enable experiments based on SIP in a distributed execution environment. The possible locations to interact with elements of the laboratory are not restricted to NTNU's premises. It was an important design criterion that interaction with the workstations and servers the students are allowed to use shall be available from both the city center and the student homes.

This concept builds heavily on the project called Wireless Trondheim, which aims for building a world-class laboratory for testing nomadic broadband wireless services [5]. Thus, network connectivity between students and laboratory equipment is maintained throughout the city and also covering the student homes via wireless IP connections (cf. Fig. 1.). Even though the network of Wireless Trondheim provides connectivity to the Internet, requests from outside the NTNU domain are filtered inside the domain. Security measures, such as authentication of users are achieved by Eduroam that is a roaming infrastructure used by the international research and education community [6].

An additional service that is available to the students while roaming in the Wireless Trondheim network is the GeoPos server [1]. This server can be used to determine physical location of the host of a single user if it is connected to the network via Wireless Trondheim. The positioning service provided by the GeoPos

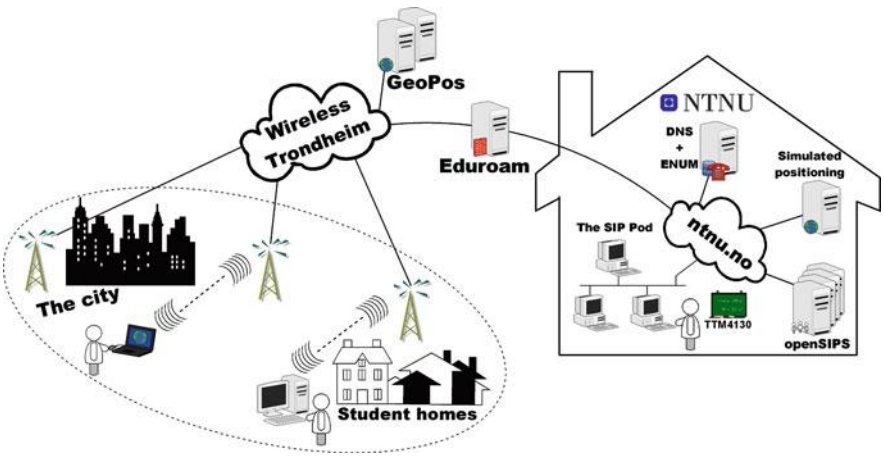


Fig. 1 The "SIP Pod" lab setup

server is basically available as a web-service and can be queried using XML encoded messages. Positioning services can be incorporated into the UA developed by the students during the semester by implementing an additional HTTP module that queries a configurable URL, i.e. it can be used to communicate via XML messages with the service. The XML responses contain, among others, latitude and longitude information revealing the position of the hosts initiating the query.

Location information can be used during the semester assignment in various tasks. One example is the scenario of emergency call re-routing. After obtaining the location of the soft-phone location information can be sent to the proxy by extending the SDP part of an INVITE message. This enables the students to embellish their UA with emergency features by capturing the emergency calls initiated by the UA (e.g. if a callee address starts with sos@) and adding location information to it, which can be extracted and used at the proxy-side for call routing. The idea behind sending location information during an emergency call establishment provides the possibility of routing the call to the nearest hospital, fire brigade or other authority that is supposed to take action. To implement the emergency call re-routing example the students have to extend the configuration of their proxy as well. Adding a database of SIP addresses of emergency services and their geographical location can be used by the proxy. However, students need to add the functionality (i.e. adding scripts to the routing mechanisms) that extracts user location information from SIP INVITE messages at the proxy too.

A more advanced task related to SIP-based presence is encapsulating location information into presence status messages. Implementation of this mechanism is an optional task for the students. If implemented, this feature can be optional and be enabled on demand by the user of the UA. If configured to do so the soft-phone automatically attaches location information to the SIP PUBLISH message, which can be read by another end-user that is subscribed to the presence updates of this particular soft-phone. On the other end of the connection a listening agent can be implemented that separates presence status and location information of the user and can further be used to, for example, deliver data towards the UA triggered by some arbitrary criterion. This setup, if implemented by the students, can be used to develop creative services during the semester based on the capability of monitoring the contacts' geographical location. A simple service example that can be given here is a push-based service that can for example deliver photos and sightseeing information to portable devices that are carried by students wandering within the city center and maintaining a connection to the laboratory network.

4 System/Requirements Description

The experimental setup of our lab consists of the SIP UA and OpenSIPS [7] SIP proxies. We focus on giving an insight of programming to the students, along with a general idea of SIP. Thus, after an introductory laboratory exercise, the rest of the tasks concentrate on our own SIP UA in form of "ORCA", a soft phone. The ORCA is actually a complete SIP UA. However, for the purpose of the exercise and as a

student project only a template of ORCA is provided to the students. This template has basic code providing a foundation and a skeleton for additional features. The students will then contribute with their code to make the template more functional and build their own "ORCA" UA almost as complete as the original soft phone built by us beforehand. One of the crucial ideas behind the design of this laboratory is to allow the students to freely roam around the premises of the university and also to certain locations of the city and yet be able to complete their tasks. In fact, one of the fundamental features of the soft phone is to use this location information. So, by introducing mobility we aim to provide the students a practical experience of a location aware system. On top of that, this system will be built by the students themselves.

In order to ease the development of SIP applications, programmers typically use Application Programming Interfaces (APIs) that encapsulate specific aspects of SIP functionality thus enabling programmers to concentrate on the application service logic. There are several SIP APIs available. Some of them are CPL (Call Processing Language) [8], SIP-CGI [9], JAIN SIP and JAIN SDP [4], SIP Servlets [10], SIP API for J2ME [11], IMS API [12, 13] etc. In addition to these, there are many open source implementations of SIP entities – such as proxies, User Agents, and so on – that offer proprietary interfaces for building applications or extending the functionality in the product. For instance, the OpenSIPS [7] project offers an open source implementation of a SIP server, and provides low-level, nonstandard APIs for SIP application development.

In our case, we have chosen to use JAIN SIP and JAIN SDP for the development of the soft phone. Our soft phone system (SIP UA) is written in Java using JAIN SIP API [4]. The underlying design is inspired by the sample soft-phone from [14], including the event driven approach. The soft phone presented in [14] has basic features like SIP signal handling and receiving and transmitting the audio/video content over RTP. Our system extends these features by adding presence support (server based), a contact list, custom presence status, location awareness and SOS call support.

The idea of Instant Messaging (IM) has become highly successful, particularly because of the possibility for both (or even multiple) parties to communicate in near real-time. Initially, IM was limited to stationary workstations but these days mobile devices such as mobile phones are becoming equipped with such capabilities too. In fact, the concept of IM is highly suited to mobile users as they are likely to roam around and not stay at fixed places. Even though SIP is originally a signaling protocol, it also provides foundation to support IM [15]. Presence is an accompanying technology to IM. Presence allows an entity to indicate its status to other entities. Such status may be related to its availability, for example, "Busy", "Away", "Out to lunch" etc. The Internet Engineering Task Force's SIMPLE working group [16] aims to design a set of backwards compatible extensions to SIP known as SIMPLE (SIP Instant Messaging and Presence Leveraging Extensions). These extensions are integrated well within SIP to provide the capabilities of today's instant messaging and presence systems. SIMPLE also manages the distribution and permission rights

to private information so that the amount of information that is published can be controlled.

Presence Information Data Format (PIDF) [17] is the standard for describing the messages sent for presence. Our system is compatible with the PIDF format. All the presence related SIP messages namely, PUBLISH, SUBSCRIBE and NOTIFY are built using PIDF. As such, the system is compatible with any other SIP phone with PIDF based presence support. Presence related activities in our system are as follows. As the soft-phone is turned ON, it sends a PUBLISH message to the proxy. It always sends status as "Online" during this stage. While the soft-phone is running, a similar PUBLISH message is sent each time, the user changes his status. The user can add or remove his custom statuses also. Some status like "Online", "Offline", "Busy", "Away", "Be Right Back" and "Invisible" are considered as keywords and hence, the user cannot delete them or add them. When the soft-phone is turned OFF, or the user exits, it sends a PUBLISH message with "Offline" status. Similar to the PUBLISH message, the soft-phone also sends SUBSCRIBE message for the saved contacts, as it is turned ON. The SUBSCRIBE message is then sent for each new contact added. Also a timer is set when starting off the soft-phone that sends SUBSCRIBE in a periodic manner. The soft-phone automatically adjusts the timer so that the message is sent before the previous SUBSCRIBE expires. A NOTIFY message is sent by the proxy if the contact updates it presence status. For example, suppose Alice has Bob and Trudy in her buddy list. Then when Bob or Trudy updates the presence status, Alice receives the NOTIFY message with this updated status. On receiving the NOTIFY message the contact list is automatically updated. Fig. 2 shows an example of a PUBLISH message sent by our system. The message in the figure publishes the presence status "Online" for 'alice@somedomain.com'.

Fig. 3 shows an example of a SUBSCRIBE message sent by 'alice' for subscription of presence status of 'bob'.

Fig. 4 shows a NOTIFY sent by the SIP proxy to 'alice'. The message carries status of 'bob'.

Location Awareness enables an application to take advantage of geographic location information such as longitude, latitude, altitude etc. The application can receive geographic information from several sources such as GPS, etc. Such geographic location information can be implemented in different ways by different applications. During a phone call, the soft-phone can send its geographic location along with the INVITE message and the proxy can then redirect the call to some pre-specified destination based on the location information in the body of the message. This idea is used in our system to implement SOS calls. Our soft-phone receives geographic location information through a web based service called "Geographical Position Service" (GeoPos) [1]. In our implementation, the URL for retrieving the location information can be configured through a configuration dialog box. A sample screenshot is shown in Fig. 5.

When the user makes any SOS calls, the geographic latitude and longitude information are embedded in the INVITE message as an attribute in its SDP content. Fig. 6 shows a sample snapshot of a typical SOS call made from our soft-phone.

```
PUBLISH sip:alice@somedomain.com:5060 SIP/2.0
Call-ID: f9051a85c1d8b899158ed8da8b4c4617@someotherdomain.com
CSeq: 1 PUBLISH
From: <sip:alice@somedomain.com:5060>;tag=56438
To: <sip:alice@somedomain.com:5060>
Via: SIP/2.0/UDP
someotherdomain.com:5566;branch=z9hG4bK1cda419
Max-Forwards: 70
Contact: <sip:alice@someotherdomain.com:5566>
Expires: 300
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY, MESSAGE,
SUBSCRIBE, INFO
Event: Presence
Content-Type: application/pidf+xml
Content-Length: 440

<?xml version='1.0' encoding='UTF-8'?>
<presence
xmlns='urn:ietf:params:xml:ns:pidf'
xmlns:dm='urn:ietf:params:xml:ns:pidf:data-model'
xmlns:rpid='urn:ietf:params:xml:ns:pidf:rpid'
xmlns:c='urn:ietf:params:xml:ns:pidf:cipid'
entity='sip:alice@somedomain.com'>
<tuple id='alice'>
  <status>
    <basic>open</basic>
  </status>
</tuple>
<dm:person id='alice'>
  <rpid:activities>
    <rpid:Online/>
  </rpid:activities>
  <dm:note>Online</dm:note>
</dm:person>
</presence>
```

Fig. 2 PUBLISH message

```
SUBSCRIBE sip:bob@somedomain.com SIP/2.0
Call-ID: d12fecc1a76f00e2097cc73536da7cbe@someotherdomain.com
CSeq: 1 SUBSCRIBE
From: <sip:alice@somedomain.com:5060>;tag=56438
To: <sip:bob@somedomain.com>
Via: SIP/2.0/UDP someotherdomain.com:5566;branch=z9hG4bK2b267a3
Max-Forwards: 70
Contact: <sip:alice@someotherdomain.com:5566>
Expires: 300
Event: Presence
Content-Length: 0
```

Fig. 3 SUBSCRIBE message

The format of this entry is as below.

$$a = sos :< geographic\ latitude > - < geographic\ longitude >$$

In Fig. 6, latitude is 63.4301... while the longitude is 10.4012....

```
NOTIFY sip:alice@someotherdomain.com:5566 SIP/2.0
Via: SIP/2.0/UDP somedomain.com;branch=z9hG4bKc7e4.58927231.0
To: <sip:alice@somedomain.com>;tag=56438
From: <sip:bob@somedomain.com>;tag=10.6172.1217066055.596
CSeq: 2 NOTIFY
Call-ID: 15a0c8f563c41647e96a7783b1365557@129.241.131.40
Max-Forwards: 70
Event: presence
Contact: <sip:somedomain.com:5060>
Subscription-State: active;expires=234
Content-Type: application/pidf+xml
Content-Length: 488

<?xml version="1.0" encoding="UTF-8"?>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
xmlns:dm="urn:ietf:params:xml:ns:pidf:data-model"
xmlns:rpid="urn:ietf:params:xml:ns:pidf:rpid"
xmlns:c="urn:ietf:params:xml:ns:pidf:cipid"
entity="sip:bob@somedomain.com">
  <tuple id="bob">
    <status>
      <basic>open</basic>
    </status>
  </tuple>
  <dm:person id="bob">
    <rpid:activities>
      <rpid:Online/>
    </rpid:activities>
    <dm:note>Online</dm:note>
  </dm:person>
</presence>
```

Fig. 4 NOTIFY message

The SIP proxy on receiving the message above can invoke a script, e.g. in Perl, and it can run through a list or a database to find the nearest hospital, fire brigade or police station based upon the geographic coordinates. The nature of the destination i.e. hospital or fire brigade or others can be determined by the SIP URI or additional description field as an additional attribute line in SDP body. Our system sends different URIs, for example, sos@somedomain for general SOS calls, fire@somedomain for SOS call to a fire brigade and so on. An important advantage of including the geographic coordinates is that the precise location of the one in need of help can be located by the authority, which may be the police or other similar bodies. Later investigations can also be supported, but of course, for this the SIP proxy must keep a record of the SOS calls. In future we will make an effort to model these functions according to proposals currently worked upon by the IETF working group ECRIT [18].

5 Conclusions

While the first wave with UAs on PCs/Laptops is gratifying, opportunities for future work remain. It would be interesting to see the UAs running on smart mobile devices

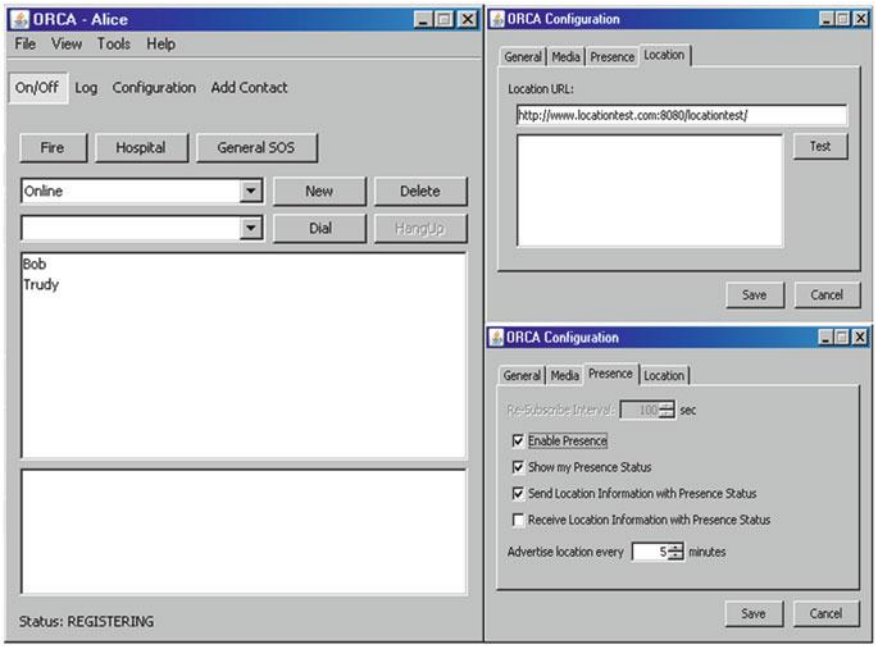


Fig. 5 A sample screenshot of the SIP soft phone and two major configuration panes, for Location and for Presence

```
INVITE sip:sos@somedomain.com:5060 SIP/2.0
Call-ID: e7068350277ce5a817d14f9ab0b7a15c@someotherdomain.com
CSeq: 1 INVITE
From: <sip:alice@somedomain.com:5060>;tag=56438
To: <sip:sos@somedomain.com:5060>
Via: SIP/2.0/UDP
someotherdomain.com:5566;branch=z9hG4bKa92abca860e26690c0ba07fe
Max-Forwards: 70
Contact: <sip:alice@someotherdomain.com:5566>
Content-Type: application/sdp
Content-Length: 183

v=0
o=- 3426055765 3426055765 IN IP4 129.X.X.X
s=-
c=IN IP4 129.X.X.X
t=0 0
a=sos:63.430106668738 - 10.401261146356
m=audio 40000 RTP/AVP 3
m=video 50000 RTP/AVP 26
```

Fig. 6 Sample SOS call INVITE with Location Information

such as mobile phones, PDAs. This would require rewriting the UAs with a different API that is particularly designed for mobile devices. An implementation API, for JSR-180 SIP API for J2ME [11] specification developed under the Java Community Process, is provided by Nokia. Nokia also provides APIs for several other forms

of SIP and VoIP development [19]. Another worthy idea would be maintaining a collection of services created by students and making it an open resource for the research, development and educational community. This way anybody with interest in the SIP Pod can contribute further.

As the current implementation of "ORCA" is targeted for SIP laboratory purposes, we did not consider any security related aspects. But it will be of a great value if future releases of "ORCA" address the security risks and provide built-in countermeasures for them. An interesting way of making it possible might be by integrating SIP laboratory exercises with security related courses at our department. Students studying in security courses will, then, get an opportunity to implement the security measures on a practical system. It is well known that SIP is vulnerable to wide varieties of threats [20]. Assignments related to security can also be further divided into several categories. All the work need not be related only to the development and implementation of security measures. Some of the work might be related to testing as well. Several testing tools are available either as open source or closed source [21]. These tools can not only be helpful in finding the vulnerabilities in the systems developed by students, but also provide them an idea of how an adversary launches different attacks.

The lab has had its first run in spring 2009. With the introduction of the SIP Pod in our courses at NTNU we expect to inspire students to explore the possibilities of intelligent networks and to create new services on their own; maybe demonstrate such services to their friends and families (as the services can be accessed from outside the campus). Moreover, we invite other educational institutes to collaborate with us on further developments related to the SIP Pod.

References

1. The Geographical Positioning Service (GeoPos) at NTNU: http://dellgeopos.item.ntnu.no/, accessed 30.01.2009
2. P. Faltstrom, and M. Mealling. The E. 164 to uniform resource identifiers (URI) dynamic delegation discovery system (DDDS) application (ENUM), RFC 3761, IETF, pp.4–11, April 2004
3. M. Mealling, and R. Daniel. The naming authority pointer (NAPTR) DNS resource rec, RFC 2915, IETF, September 2000
4. JAIN SIP: Java API for SIP Signaling: https://jain-sip.dev.java.net/, accessed 30.01.2009
5. Wireless Trondheim: http://tradlosetrondheim.no/index.php?la=en, accessed 30.01.2009
6. Eduroam: http://www.eduroam.org/, accessed 30.01.2009
7. OpenSIPS project: http://www.opensips.org/, accessed 30.01.2009
8. J. Lennox, Call processing language (CPL): A language for user control of internet telephony services, RFC 3880, IETF, pp.4–8, October 2004
9. J. Lennox, H. Schulzrinne, and J. Rosenberg. Common gateway interface for SIP, RFC 3050, IETF, pp.12–30, January 2001
10. SIP Servlet Developer's Homepage: http://www1.cs.columbia.edu/~ss2020/sipservlet/
11. SIP API for J2ME: http://developers.sun.com/mobility/apis/articles/sip, accessed 30.01.2009
12. IMS API: http://jcp.org/en/jsr/detail?id=281, accessed 30.01.2009
13. An IP Multimedia Subsystem (IMS API/Framework for Java SE: http://code.google.com/p/ims-api/, accessed 30.01.2009
14. R.M. Perea. Internet multimedia communications using SIP, a modern approach including java practice, Elsevier, pp.541–543 2008

15. B. Campbell, J. Rosenberg, H. Schulzrinne, C. Huitema, and D. Gurle. Session Initiation Protocol (SIP) Extension for Instant Messaging, RFC 3428, IETF, pp.2–5 December 2002.
16. IETF Working Group "SIMPLE": http://www.ietf.org/html.charters/simple-charter.html, accessed 30.01.2009
17. H. Sugano, S. Fujimoto, G. Klyne, A. Bateman, W. Carr, and J. Peterson. Presence information data format (PIDF), RFC 3863, IETF, pp. 5–18, August 2004
18. IETF Work Group "Emergency Context Resolution with Internet Technologies (ECRIT)", http://www.ietf.org/html.charters/ecrit-charter.html, accessed 30.01.2009
19. Forum Nokia – VoIP Documentation: http://www.forum.nokia.com/Resources_and_Information/Documentation/VoIP/, accessed 30.01.2009
20. S. Sawda, and P. Urien. SIP security attacks and solutions: A state-of-the-art review. IEEE Network, pp.3187–3191 2006
21. E. Cha, H.K. Choi, and S.J. Cho. Evaluation of security protocols for the session initiation protocol. IEEE Network, pp.611–616, 2007

Index

F. Davoli et al. (eds.), *Remote Instrumentation Services on the e-Infrastructure*,
DOI 10.1007/978-1-4419-5574-6