Deployment of an Interactive Physics Analysis Application in the CrossGrid Project Testbed

EU-CrossGrid Project Tasks 1.3 and 4.1 Collaborators
(see list of authors at the end of the paper)

Abstract

The experience on the deployment of an Interactive Physics Analysis Application in the Grid environment provided by the Testbed of the EU project CrossGrid (IST-2001-32243) is described. The core of the application is a distributed training of an Artificial Neural Network (ANN), used by the authors in several physics fields. The process of adaptation for this application to a Grid environment ("gridification") is described, including mechanisms for distributed execution and load balancing. Preliminary results on performance measured in the CrossGrid testbed are also given. A detailed description of the tools used for development, and the Grid Environment resources, is included.

1. Introduction

The European CrossGrid project (see http://www.eu-crossgrid.org) develops new components for interactive compute and data intensive applications in a Grid [1] framework. The elaborated methodology, generic application architecture, programming tools, and new grid services are validated and tested on the CrossGrid testbed, with an emphasis on a user friendly environment. The work is done in close collaboration with the European DataGrid (EDG) project (see http://www.eu-datagrid.org) to profit from their results and experience, achieving full interoperability that has resulted in the further extension of the Grid framework across eleven European countries.

The applications considered are characterised by the interaction with a person in a processing loop: they require a response from the computing system to an action by the person in different time scales, and they are simultaneously compute- as well as data-intensive. Examples of these applications are: interactive simulation and visualisation for surgical procedures, flooding crisis team decision support systems, distributed data analysis in physics, and air pollution combined with weather forecasting. To enable an efficient development, new tools for verification of parallel source code and monitoring have been developed, as well as components for efficient distributed data access and specific resource management.

This short report details the case study for the application for Physics Analysis that has been adapted and deployed on this framework, following this schema: section 2 describes briefly the application, and is followed by the discussion in section 3 on the benefits expected from running in a Grid framework; section 4 describes in detail the Grid design pattern of the application, and sections 5 and 6 present the CrossGrid testbed and the support tools, used to obtain the results on performance shown in section 7. Finally the lessons learned are briefly summarized in the conclusions.

2. Overview of the Application

Artificial Neural Networks (ANN) have become the "de facto" standard in many physics fields to address complex multidimensional analysis problems. As a relevant example in Particle Physics, the recent search for the Higgs Boson at the Large Electron Positron Collider (LEP) at CERN, used this technique in most of the channels, obtaining a significant performance improvement compared to typical "sequential cuts" or

"likelihood" analysis approaches. Typically large simulations of the production of new elementary particles in collisions in accelerators are required. The information recorded in the detector for a single collision is called an “event”, and its size may vary up to several Mbytes, and includes a large number of measured and derived variables. It is stored in files and databases distributed over several computing centres collaborating in the experiment. Although the ANN structure itself can be moderately complex (in the Higgs search in the hadronic channel in the DELPHI collaboration the ANN used 16 input variables, two intermediate layers of 10 hidden nodes each and one final output node, so a 16-10-10-1 architecture), the training typically requires more than thousand iterations or “epochs”, running on samples of more than 100.000 events pre-selected from total simulated samples of several millions of events. Independent samples of a similar size are used to test the discrimination power of the ANN avoiding any overtraining problem. The user, a physicist, may spend long working sessions in front of the screen, first understanding all the involved variables and gathering the filtered information of interest from the distributed resources, and then building the best possible ANN, using different architectures and input variables to get the optimal selection function that when applied on the real data will select the most signal-like events.

3. Benefits of the Grid

The ANN training process demands a large computing power, requiring several hours in a dedicated processor. However, physicists see today these techniques as a direct replacement of the old "sequential selection cuts" methods, and would like to use them in an almost interactive way, without having to wait for long to try out different possibilities in the analysis.

It has been shown on a cluster environment that parallelization techniques can significantly reduce this waiting time: for an example running on a 64-nodes cluster, it was reduced from 5 hours to less than 5 minutes. However this requires dedicated local resources for interactive use, and the local availability of the data.

In a Grid framework, the user benefits from sharing distributed resources, reducing large data transfer from simulation sites and finding more easily available processors in a larger pool. Moreover the Grid environment provides a Virtual Organization framework to support the Physics Analysis community in a large distributed collaboration.

4. Grid Design Pattern

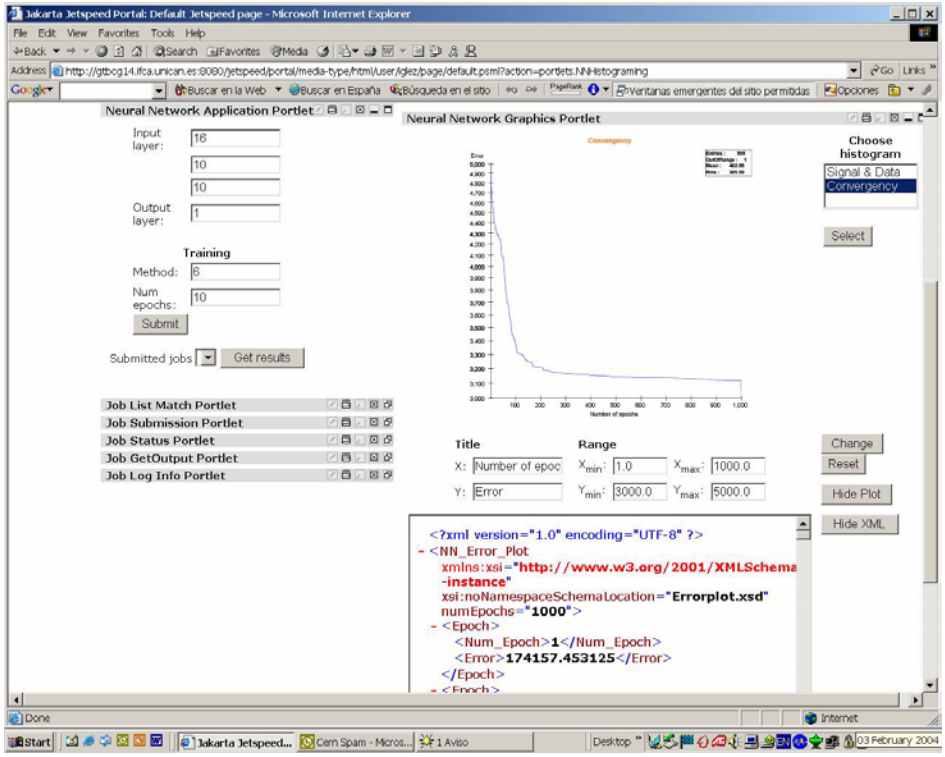
The application is based on the MultiLayer Perceptron package (MLP) that includes a well known algorithm, BFGS[2] that suits a basic parallelization scheme with a Master and N slaves that was implemented using MPI in a local cluster configuration. As described in detail in previous work, this scheme allows the distribution of the work among several machines where the corresponding stripped data is locally available and processed in parallel. Latency issues are well under control as messaging between the master and slaves nodes is reduced to the ANN node connection weights and the returned estimated errors, that are added by the master to estimate the gradient and calculate the next minimization step.

A basic step towards the Grid framework is the use of the Globus[3] aware MPICH-G2[4]. Results on the local cluster showed no penalty on efficiency, when compared to the MPICH-P4 version, and first tests on distributed machines across the network using directly the Globus RLS (Resource Language Specification) were also encouraging.

However one of the first issues observed was the need for load balancing to take into account the different processing power of distributed resources. The naïve approach for achieving load balancing via adaptation of the total size of each slave input dataset has worked remarkably well, allowing the improvement of global performance as described in detail below.

Another Grid issue was the data management. The input data is prepared by scanning the event databases and files corresponding to collisions, and converting the selected data into XML files, a process being also parallelized. Each event dataset, is identified by a couple of tag numbers labelled as "run" and "event" in these files, and an attribute is used to mark the event as signal (1) or background (0) like. The registration of these files in the Grid framework returns a logical file name (lfn) that can be used by the computing resources via the Replica Location Service to transfer them locally.

The new CrossGrid Resource Broker provides parallel job submission to MPICH-G2 enabled resources. In fact the CrossGrid Roaming Access Server (RAS) front-ends, either the Migrating Desktop or the Interactive Portal, hides the complexity of this process to the end user, that simply registers using her/his digital certificate, selects the application parameters including the number of desired nodes for distributed execution, and receives in his desktop screen the graphical output showing in real time the convergence of the ANN training (i.e., the total classification error) as can be seen in the Portal snapshot shown below.



The interactivity also allows job cancellation and re-submission if the convergence is not as desired. The final output, i.e. the collection of ANN weights and total error, can be stored in the Virtual Directory included in the RAS, giving access to local desktop space, or remote storage managed thanks to the Replica Manager or accessed directly under "grid-ftp".

The support for this Grid enabled activity is provided by the Crossgrid testbed described in detail in the next section.

5. The CrossGrid Testbed

The CrossGrid international distributed testbed shares resources across sixteen European sites and this is one of the challenging points of the CrossGrid project. The sites range from relatively small computing facilities in universities to large computing centres, offering an ideal mixture to test the possibilities of the Grid framework. National research networks and the high-performance European network, Géant, assure the interconnectivity between all sites. The network includes usually three steps: the local step (typically inside a University or Research Centre, via Fast or Gigabit Ethernet), the jump via the national network provider (at speeds that will range from 34 Mbits/s to 622 Mbits/s or even Gigabit) to the national node, and finally the link to the Géant network (155 Mbits/s to 2.5 Gbits/s). The figure below shows a map with the different nodes, including the major network links.

The CrossGrid testbed largely benefits from the DataGrid (EDG) experience on testbed setup. The basic testbed middleware is composed of EDG and Globus middleware distributions, and in particular the current installation is based in the package named LCG-1 assembled for the LHC Computing Project at CERN (LCG).

The testbed includes two different types of services: global collective services and distributed computing services. Regarding computing resources, all sites include:

- An installation server using the LCFG-ng software
- A Computing Element (CE) machine
- An Storage Element (SE) machine
- A User Interface (UI) machine
- At least two Worker Nodes (WN) or a dual CPU system

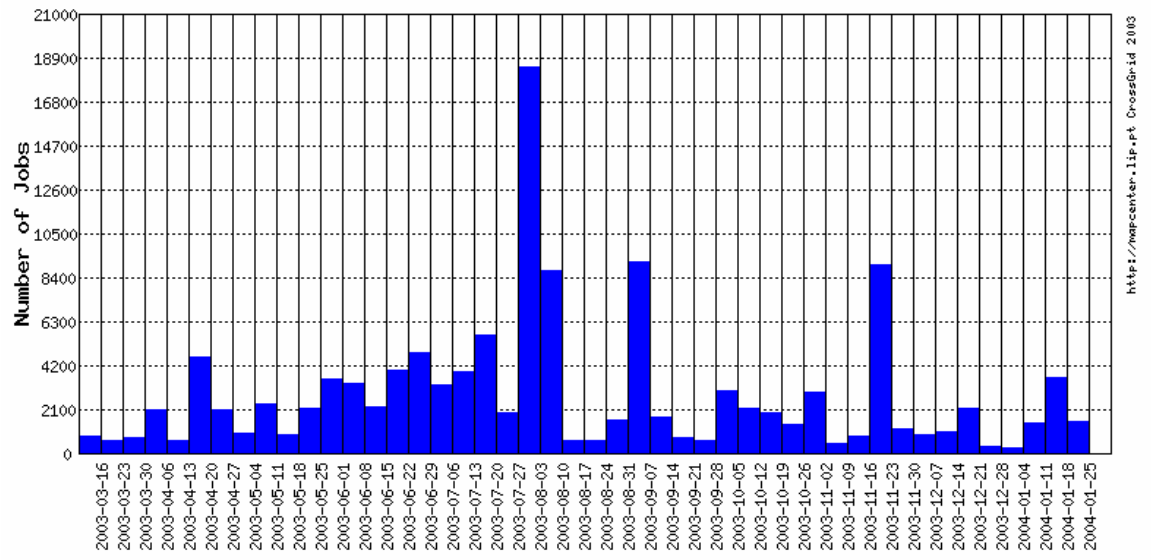
providing a total of 104 CPUS (64 WN) and an storage volume above 4400 GB (18 SE).

Integration of these resources is done by the global collective services:

- Resource Broker (RB): the heart of the testbed workload management system. The RB receives job requests sent by users and finds computing resources suitable to run the jobs. The CrossGrid RB supports single and parallel jobs (inside a cluster via MPICH-P4 or distributed using MPICH-G2)
- Information Index (II): the root entry point for the MDS information tree that contains the resources information published by the CE and SE systems.
- MyProxy: repository of certificate proxies for long-lived jobs and portals.
- Virtual Organization server: repository of authorization information used by the testbed systems to generate the authorization databases. The CrossGrid VO server contains the list of all CrossGrid users authorized to use the testbed.
- Replica Location Services (RLS): a service that stores information about the location of physical files in the grid. The RB uses the RLS to make scheduling decisions based on the location of the files required by the jobs. The users can also use the RLS service through the Replica Manager (RM). Although this is an EDG service CrossGrid is developing components for its access optimization functionality.
- Mapcenter: an EDG tool enhanced by CrossGrid to monitor the connectivity of systems and services.
- Roaming Access Server (RAS): CrossGrid specific service that provides support for the Migrating Desktop (MD) and portal applications.
- Roaming Access Server Job Submission Service (RAS JSS): a component of the RAS server that performs the actual job submission.

The production central services are located mainly in four sites. Resulting from the collaboration between LIP and the Portuguese Academic Network (FCCN) on Grid technologies the systems at the LIP/FCCN site are physically located in the FCCN Network Operations Center in Lisbon. The systems enjoy a bandwidth connection of 100Mbit/s to the multi-Gigabit Pan-European network backbone Géant. The remaining collective services are located at different sites under control of the corresponding project partners. It should be noticed the presence of two Roaming Access Servers (RAS) in Poznan and Nicosia. These two servers have been replicated to increase the reliability of the RAS service that supports both the CrossGrid Migrating Desktop and the CrossGrid portals. In the future these RAS servers will become part of a RAS network that will include automatic balancing and fault tolerance between the RAS servers.

A graphical view of the global testbeds usage can be obtained from the Computing Element (CE) statistics available from the CrossGrid Mapcenter. The graphic shows the total amount of job submissions in the last 45 weeks. The statistics are collected directly from the CEs and include direct globus job submissions and job submissions through the Resource Brokers.



The authentication of users and systems is performed through a public key infrastructure based on X.509 certificates. Since certificates must be issued by trusted certification authorities (CAs) CrossGrid choose to trust the national CAs already established by EDG, and to coordinate the deployment of new national CA's where necessary. New grid certification authorities have been established in the countries where they were not previously available namely: Poland, Germany, Slovakia, Greece and Cyprus. A strong effort was made to make the new CAs recognized by DataGrid creating new opportunities for sharing resources between CrossGrid and DataGrid and therefore extending the grid coverage in Europe.

6. Development and Support Tools

The Crossgrid GridPortal, (http://gridportal.fzk.de), built using the Savannah (SourceForge) tool, provides central support for testbed sites manager and developers, including CVS code repository, mailing list-forums, bug-trackers, etc.

The testbed site managers use the LCFGng installation and configuration software, providing the RedHat 7.3 operating system including security patches, the LCG-1 distribution and the CrossGrid specific software. A wrapping of the LCG-1 configuration profiles and RPM lists to enable a better management and inclusion of CrossGrid specific software and configurations was implemented, giving origin to a common installation maintained centrally. This installation process is extensively documented in the CrossGrid “Cluster Installation Manual”.

Developers benefit also of specific CrossGrid tools. In particular, for the ANN application, an MPI Profiling Interface (MARMOT) was used to verify the correctness of this parallel, distributed application using the MPI paradigm, to make it portable, reproducible and reliable. In addition, a “Developers Guide” describes all necessary steps to deploy software in the CrossGrid testbed. A testing and validation process is performed in a dedicated testbed before any new middleware is installed in all sites. All the deployment process of a new software into the CrossGrid testbed is “tracked” by the CrossGrid Integration Team, providing personal advice to each developer, and helping to keep a coherent and complete set of middleware installed.

Finally, users are supported through a CrossGrid Help Desk, a web-based system customized from the OneOrZero initiative, and a Tutorial is available specifically for the testbed use and the ANN application.

7. Performance

Understanding performance issues requires a detailed description of the implementation of the parallel training algorithm, described below:

i) The master node starts the work reading execution parameters like the number of reserved nodes, its processing power, the input parameters for the ANN architecture, and the description of the training dataset, i.e. the relation of all logical file names and number of events for each one.

ii) Using this information, the master determines an optimal splitting of the input data set between all slave nodes, and sends to them the corresponding logical file names. Optionally, the decision can also take into account the cost of the data transference from the Storage Elements to the slaves (Worker Nodes) using the Crossgrid Data Access Optimization module. The master also initializes all ANN weights to random values and transmits them through MPI to the slaves.

iii) The Worker Nodes (slaves) distributed across the different testbed sites transfer locally the assigned input datasets, using the EDG Replica Manager.

iv) For each iteration (epoch), the master sends the weight values to the slaves, which compute the error and the gradient on the partial dataset and return them to the master; as both are additive, total errors are calculated by the master that prepares the new weights along the corresponding direction in the multidimensional space and update the weights. The error computed in a separated test dataset is used to monitor the convergence and displayed also in the graphical output.

v) The previous step is repeated until a convergence criteria is fulfilled or a predetermined number of epochs is reached. The master returns as output the ANN weights, describing the multidimensional classification function.

vi) The whole training process can be repeated with different initial random weights to prevent incorrect results due to local minima. The final error can be used to select the best final ANN function if different minima are found.

The performance can be measured by the time employed to train an ANN with a given architecture (in our case 16-10-10-1) on a fixed dataset (corresponding to about 400K events) for a fixed number of epochs (100). The final result is expected to depend upon the processing power of the selected master node, the number and power of slaves nodes, the effect of latency between the master and slaves for MPICH-G2 traffic over the network, and the initial time to transfer locally the input files. A good load balancing should take into account all these factors.

First tests[5] in a cluster environment were done at IFCA centre in Santander, using up to 32 slaves in a cluster of dual Pentium III IBM x220 servers (1.26 GHz, 512 Kb cache, 640 MB RAM with 36 GB SCSI and 60 GB IDE disks) running Linux RH7. With all machines being identical, no load balancing was needed. The time reduction went from 11250 s of wait time (more than 3 hours) when using only one slave, down to 400 s. when using 32 slaves. For later comparison with the Grid environment, the time employed using 8 slaves was 1450s.

The study of the performance on the Grid environment has been started on the CrossGrid testbed. The first study was done using only few nodes distributed at several sites in Spain, Portugal and Poland in order to reduce the complexity of understanding the results obtained, while preserving the distributed nature of the experiment.

An important issue, as already indicated, is the availability of distributed replica for the training data, as its proximity in access time to the computing resources is a key

point to reduce the waiting time. The Grid framework provides several interesting possibilities on data replication and distributed data access. For this first tests the data have been copied and registered into Crossgrid Replica Catalogue, and further replicated into several testbed Storage Elements. When choosing a single cluster to perform the ANN training, it is enough that the master node downloads the input data files to the common directory used by all slaves. In case of multi-site training, both the master and the slaves should -in a first step- download this input data.

The second issue is load balancing. The used dataset contains a total of 400K events with the following components: 5K signal (Higgs bosons) events and two different kinds of background events: 300K WW events, and 100K QCD events. For our tests we further divided each of the subsets into 80 slices, and replicated them across the participating nodes. In a configuration using a master node (PIII@1.26 GHz), and 5 slaves (2.4 GHz P4 Xeon in a cluster in Poland and 1.7 GHz P4 in Portugal), we observed that an equal distribution of events between them resulted in a heavy performance penalization due to the slowest slaves, where each training epoch took almost 40% more time than in the fastest ones. Introducing a weighting factor (in a first approach the CPU speed ratio) on the data size distributed to the slaves, this difference was reduced to only 5%. So, the training on the same total number of events, 400K, that took around 1.07 s per epoch, was reduced to 0.85 s per epoch after load balancing.

For the complete execution, a first example of the tests shows that using a comparable master-slave configuration (1 master and 8 slaves) distributed across the testbed, the total time employed was 1630 s., to be compared with the 11250 s. using only one node, and the 1450 s. running in a local cluster environment.

These first numbers show the good perspectives in a distributed computation scheme for this application, and will be extended to situations where larger datasets are employed and the transfer time may play a key factor.

Additional information is being obtained now from two new CrossGrid tools taking into account the application kernel: the Grid-Benchmarking application, and the Performance Prediction Component. It will be reported in a future paper.

8. Conclusions and Lessons Learned

The experience on the deployment of an Interactive Physics Analysis Application in the Grid environment provided by the Testbed of the EU project CrossGrid has been described. Even for an application that could be typically classified as HPC-like, the training of an Artificial Neural Network (ANN), the adaptation to a real Grid environment has provided satisfactory results, thanks to the distributed execution and load balancing mechanisms implemented. Preliminary results on performance measured in the CrossGrid testbed available resources show that even without taking yet into account the benefit of avoiding large data transfers to a central site, the application execution time is comparable to the one obtained using resources from a dedicated local cluster.

Several lessons have been learned along the execution of the project:

- The selection of a basic grid middleware with reasonable support and including most of the features needed to support the gridification of an application is a key point, and once done, stability is highly demanded by developers and users.
- Developer support tools are important, in particular a modern project development environment is mandatory, but it should be complemented by adapted installation and deployment tools, detailed guides, and overall a good integration team with experience on a real Grid testbed. This will be needed yet for some time as developers are overwhelmed by the current complexity in

details of the framework, and to keep a good coherency until a more "standard" set of basic services is well established. For daily use, a friendly environment like the one provided by CrossGrid Migrating Desktop or Portal is very convenient.

- Testbed installation and maintenance starts to be better understood. Several tools simplify the delicate configuration process, and more powerful collective services help to better benefit from the distributed resources, keeping a satisfactory monitoring level.
- Data management tools are very useful, and should be further enhanced to profit from the possibility of handling local data at local computing resources. This needs the combined use of access cost functions and the flexibility to mix Computing and Storage services in the same machine.

As a final conclusion, the deployment of a realistic complex distributed application over a real Grid testbed covering many different sites has been successfully performed. Further work will allow the exploitation in a scientific environment, in particular for the Interactive Physics Analysis over large distributed data samples that will be collected in the new LHC accelerator experiments at CERN.

Acknowledgements

This work has been mainly supported by the European project CrossGrid (IST-2001-32243). We would like to thank in particular all our colleagues in this project.

Bibliography

[1] I. Foster and C. Kesselman, editors.
The Grid: Blueprint for a Future Computing Infrastucture, Morgan Kaufmann Publishers, 1999.

[2] Broyden, Fletcher, Goldfarb, Shanno (BFGS) method.
For example in Practical Methods of Optimization, R.Fletcher. Wiley (1987)

[3] The Globus Project.
http://www.globus.org

[4] N. Karonis, B. Toonen, and I. Foster,
MPICH-G2: A Grid-Enabled Implementation of the Message Passing Interface},
Journal of Parallel and Distributed Computing (JPDC), to appear, 2003.

[5] O. Ponce et al.
Training of Neural Networks: Interactive Possibilities in a Distributed Framework.
In D. Kranzlmuller et al. (Eds.)
9th European PVM/MPI, Springer-Verlag, LNCS Vol. 2474, pp. 33-40, Linz, Austria, September 29-October 2, 2002.

List of authors

C. Martinez-Rivero, J. Marco, D. Rodriguez, R. Marco, I.González, D. Cano, I. Diaz
Instituto de Fisica de Cantabria (CSIC-UC), Santander, Spain

A. Fernández, S.Gonzalez, J.Sánchez, J.Salt, F. Fassi, V.Lara
Instituto de Física Corpuscular(CSIC-UV), Valencia, Spain

J.Gomes, M.David, J.Martins, L.Bernardo
Laboratorio de Instrumentacao e Fisica de Particulas, Lisbon, Portugal

M.Hardt and A.García
Forschungszentrum Karlsruhe GMBH, Germany

P. Nyczyk and A.Ozieblo
Akademickie Centrum Komputerowe CYFRONET, Krakow, Poland

P.Wolniewicz
Poznan Supercomputing and Networking Center, Poznan, Poland

M.Bluj, K. Nawrocki, A.Padee, Wojciech Wislicki
A.Soltan Institute for Nuclear Studies, Interdisciplinary Centre for Mathematical and Computational Modelling, University of Warsaw, and Instytut Radioelektroniki PW, Warsaw, Poland

C.Fernández, J.Fontán, A.Gómez, I.López
CESGA, Centro de Supercomputacion de Galicia, Santiago de Compostela, Spain

Y.Cotronis, E.Floros
National Center for Scientific Research "Demokritos", National and Kapodistrian University of Athens, Dep. of Informatics and Telecommunications, Greece

G.Tsouloupas, W.Xing, M.Dikaiakos
University of Cyprus, Cyprus

J.Astalos
Ustav Informatiky Slovenska Akademia Vied, Bratislava, Slovakia

B.Coghlan
Trinity College Dublin, Ireland

E.Heymann, M.Senar
Universitat Autonoma de Barcelona, Spain

C.Kanellopoulos
Aristotle University of Thessaloniki, Greece