EGEE User Forum

Wednesday, 1 March 2006 - Friday, 3 March 2006

CERN

Book of Abstracts

EGEE User Forum
Draft version

Contents

User Forum Plenary 1 / 1

The Grid and the LHC experiments: achievements and open issues

Author: Nick Brook[1]

[1] *CERN and Bristol University*

User Forum Plenary 1 / 2

The Grid and the Biomedical community: achievements and open issues

Author: Isabelle Magnin[1]

[1] *INSERM Lyon*

User Forum Plenary 1 / 3

Experience integrating new applications in EGEE

Author: Roberto Barbera[1]

[1] *University of Catania and INFN*

User Forum Plenary 1 / 4

Welcome

Author: Frederic Hemmer[None]

User Forum Plenary 1 / 5

Setting the scene

Author: Bob Jones[1]

[1] *CERN*

6

Summary of session 1a

7

Summary of session 1b

8

Summary of session 1c

9

Summary of session 1d

User Forum Plenary 3 / 10

Summary of parallel session 2a

Author: Harald Kornmayer[1]

[1] *Forschungszentrum Karlsruhe*

User Forum Plenary 3 / 11

Summary of parallel session 2b

Author: Johan Montagnat[1]

[1] *CNRS*

User Forum Plenary 3 / 12

Summary of parallel session 2c

Author: Cal Loomis[1]

[1] *LAL Orsay*

User Forum Plenary 3 / 13

Summary of parallel session 2d

Author: Flavia Donno[1]

[1] *CERN*

14

EGEE Training

User Forum Plenary 3 / 15

EGEE Technical Coordination group

Author: Erwin Laure[1]

[1] *CERN*

User Forum Plenary 3 / 16

Long-term grid sustainability

Author: Dieter Kranzlmueller[1]

[1] *Linz University and CERN*

Europe has invested heavily in developing Grid technology and infrastructures during the past years, with some impressive results. The EU EGEE Project (www.eu-egee.org), which provides a coordinating framework for national, regional and thematic Grids, has proved a vital catalyst and incubator for the success of establishing a working, large-scale, multi-science production Grid infrastructure that serves many sciences. As the Virtual Organizations established by scientific communities move from testing their applications on the Grid to routine and daily usage, it becomes increasingly important and necessary to ensure maintainance, reliability and adaptiveness of the Grid infrastructure. This is rather difficult with the usual (short) project funding cycles, which inhibit investment from long-term users and industry. The situation is in some ways analogous to that of scientifc networks, where independent national initiatives led to common standards and ultimately the creation of the DANTE organization. A similar evolution needs to be planned now for Grids, i.e. National Grid Initiatives to guide Grid infrastructure deployment and operation at country-level and a central coordinating body to ensure long-term sustainability and interoperability.

17

(no title)

User Forum Plenary 3 / 18

Conference summary

Author: Massimo Lamanna[1]

[1] *CERN*

User Forum Plenary 2 / 20

The EGEE infrastructure

Author: Ian Bird[1]

[1] *CERN*

User Forum Plenary 2 / 21

gLite status and plans

Author: Claudio Grandi[1]

[1] *INFN Bologna*

Poster and Demo session + cocktail / 22

HGSM Web Application

Author: Dashamir Hoxha[1]

[1] *Institute of Informatics and Applied Informatics (INIMA), Tirana, Albania*

Corresponding Author: dhoxha@inima.al

This is a web application that serves as a front-end to the database that keeps information about the grid sites (clusters), their admins, email and phone contacts, other contact people, site nodes and resources, downtimes etc. These sites are organized by country and countries are organized by regions. The admins of each site can also update the information about the site.

Summary:

Hierarchical Grid Site Managment Application

This is a web application that serves as a front-end to the database that keeps information about the grid sites (clusters), their admins, email and phone contacts, other contact people, site nodes and resources, downtimes etc. These sites are organized by country and countries are organized by regions. The admins of each site can also update the information about the site.

To summarize the design, the applications supports only one GOC (Grid Operation Center), assuming that it is used for only one grid. It can have several ROCs (Regional Operation Centers), each ROC can have several countries, and each country can have several sites. Each ROC, country and site can have one (or more) admins, which are able to modify the data of the structure (record) which they admin (ROC, country or site), and to manage substructures as well. Managing substructures means that they can also validate/appoint the admins of the substructures, e.g. a country admin can validate/appoint the site admins. The admins are recognized automatically by their certificate (without the need to use passwords).

For each structure (ROC, country, site), contact info about it are stored, including email and phone. For the sites, general info about it are stored, contact info, info about its resources and capacities, site contacts, site downtimes, and info about its nodes. For more details see the DB design: hgsm_design.sql.

In the pages where a user has edit rights (in the node where he is admin and in all the subnodes), an Edit button is displayed. Clicking in this button, the page will be displayed in the edit mode, where the fields of information can be modified, the rows of the lists can be edited or deleted, new rows can be added in the lists, etc. The id and the admins of a node can be modified only by an admin of the parent node.

The features of HGSM can be summarized like this:

- Has a database with information about a hierarchical structure (GOC–>ROCs–>Countries–>Sites–>Nodes).
- This information is published to the www by a web application.
- Each node of the tree has also one or more admins, which can modify the data of the node which they admin, and also the data of the subnodes.
- The admins of a node can appoint (set/modify/delete) the admins of the subnodes and leave up to them the modification/update of the subnode information (to make their job easier and for decentralization).
- Authentication (recognizing that somebody is admin of a node and has the rights to modify it) is done automatically by the application, using personal certificates, which are issued and verified by a certain certification authority. This means that everybody that is an admin, must have a valid certificate installed in his browser.
- The application also supports i18n and l10n (is multilingual, can be translated into several languages).

This web application can also be modified/generalized/improved easily in order to be used for any similar kind of problems, where information about a hierarchical (tree) structure has to be stored, displayed and maintained (updated/modified). E.g. it can be adopted for an eGovernment problem, where each level of the hierarchy maintains its own info/data, and the hier levels of hierarchy have access and control over the lower levels.

For more information about the application see its webpage at:
http://hgsm.sourceforge.net/

This web application is similar to:
https://goc.grid-support.ac.uk/gridsite/gocdb2/

and is intended to have the same purpose and functionality, but it is going to be used for the SEE-GRID test sites,

and it is intended to have a cleaner design and implementation,
so that if possible, it can replace it later.

1c: Earth Observation - Archaeology - Digital Library / 23

Project gridification: the UNOSAT experience

Author: Patricia Mendez Lorenzo[1]

[1] *CERN IT/PSS*

Corresponding Author: patricia.mendez@cern.ch

The EGEE infrastructure is a key part of the computing environment for the
simulation, processing and analysis of the data of the Large Hadron Collider (LHC)
experiments (ALICE, ATLAS, CMS and LHCb). The example of the LHC experiments
illustrates well the motivation behind Grid technology. The LHC accelerator will
start operation in 2007, and the total data volume per experiment is estimated to
be
a few PB/year at the beginning of the machine's operations, leading to a total
yearly production of several hundred PB for all four experiments around 2012. The
processing of this data will require large computational, storage and associated
human resources for operation and support. It was not considered feasible to fund
all of the resources at one site, and so it was agreed that the LCG computing
service would be implemented as a geographically distributed Computational Data
Grid. This means, the service will use computational and storage resources,
installed at a large number of computing sites in many different countries,
interconnected by fast networks. At the moment, the EGEE infrastructure counts 160
sites, distributed over more than 30 countries. These sites hold 15000 CPUs and
about 9PB of storage capability.
The Grid middleware will hide much of the complexity of this environment from the
user, organizing all the resources in a coherent virtual computer centre.
The computational and storage capability of the Grid is attracting other research
communities and we would like to discuss the general patterns observed in
supporting
new applications, porting their application onto the EGEE infrastructure.
In this talk we present our experiences in the porting of three different
applications inside the Grid like Geant4, UNOSAT and others.
Geant4 is a toolkit for the Monte Carlo simulation of the interaction of particles
with matter. It is applied to a wide field of research including high energy
physics
and nuclear experiments, medical, accelerator and space physics studies. ATLAS,
CMS,
LHCb, Babar, and HARP are actively using Geant4 in production.
UNOSAT is a United Nations initiative to provide the humanitarian community with
access to satellite imaginary and Geographic System services. UNOSAT is implemented
by the UN Institute for Training and Research (UNITAR) and manager by the UN Office
for Project Services (UNOPS). In addition, partners from public and private
organizations constitute the UNOSAT consortium. Among these partners, CERN
participates actively providing the computational and storage resources needed for
their images analysis.
During the gridification of the UNOSAT project, the collaboration with the
developers of the ARDA group to adapt the AMGA software to the UNOSAT expectations
was extremely important. The satellite images provided by UNOSAT have been stored
in
Storage Systems at CERN and registered inside the LCG Catalog (LFC). The files so
registered have been identified with an easy to remember Logical File Name (LFN).
The LFC Catalog is then able to map these LFN to the physical location of the
files.
Due to the UNOSAT infrastructure, their users will provide as input information the

coordinates of each image. AMGA is able to map these coordinates (considered metadata information) to the corresponding LFN of the files registered inside the Grid. Then the LFC will find the physical location of the images.
A successful model to guarantee a smooth and efficient entrance in the Grid environment is to identify an expert support to work with the new community. This person will assist them during the implementation and execution of their applications inside the Grid. He will also be the Virtual Organization (VO) contact person with the EGEE sites. This person will work together with the EGEE deployment team and with the responsible of the sites to set the services needed by the experiment or community, observing also the relevant security and access policies. Once these new communities attain a good level of maturity and confidence, a VO Manager would be identified in the users community.
This talk will report a number of concrete examples and it will try to summarize the
main lessons. We believe that this should be extremely interesting for new communities in order to early identify possible problems and prepare the appropriate
solutions. In addition, this support scheme would also be very interesting as a model, for example, for local application support in EGEE II.

1a: Life Sciences / **24**

Construction of a Mathematical Model of a Cell as a Challenge for Science in the 21 Century and EGEE project

Author: Victor Lakhno[1]

[1] *IMPB RAS, Russia*

Corresponding Author: lak@impb.psn.ru

As recently as a few years ago a possibility of constructing a mathematical model of a life seemed absolutely fantastic. However, at the beginning of 21-th century several research teams announced creation of a minimum model of life. To be more specific, not life in general, but an elementary brick of life, that is a living cell. The most well-known of them are: USA Virtual Cell Project (V-Cell), NIH (http: //www. nrcam.uchc.edu /vcellR3 /login/login.jsp); Japanese E-cell project (http://ecell. sourceforge.net/); Dutch project ViC (Virtual Cell) (http://www.bio.vu.nl /hwconf/Silicon /index.html).
The above projects deal mainly with kinetics of cell processes. New approaches to modeling imply development of imitation models to simulate functioning of cell mechanisms and devising of software to simulate a complex of interrelated and interdependent processes (such as gene networks). With the emergence of an opportunity to use GRID infrastructure for solving such problems new and bright prospects have opened up.
To develop an integrated model of more complex object than prokaryotic cell such as eukaryotic cell is the aim of the Mathematical Cell project (http://www.mathcell.ru) realized at the Joint Center for Computational Biology and Bioinformatics (www.jcbi.ru) of the IMPB RAS. Functioning of a cell is simulated based on the belief that the cell life is mainly determined by the processes of charge transfer in all its constituent elements.
Since (like in physics where the universe is thought to have arisen as a result of a Big Bang) life originated from a DNA molecule, modeling should be started from the DNA. The MathCell model repository includes software to calculate charge transfer in an arbitrary nucleotide sequence of a DNA molecule. A sequence to be analyzed may be specified by a user or taken from databanks presented at the site of the Joint Center for Computational Biology and Bioinformatics (http://www.jcbi.ru).

Presently, the MathCell site demonstrates a simplest model of charge transfer. In

the framework of the GRID EGEE project any user registered and certified in EGEE infrastructure can use both the program and the computational resources offered by EGEE.
In the near future IMPB RAS is planning to deploy in EGEE a software tool to calculate a charge transfer on inner membranes of some compartments of eukaryotic cells (mitochondria and chloroplasts) through direct simulation of charge transfer with regard to the detailed structure of biomembranes containing various molecular complexes. Next on the agenda is a software tool to calculate metabolic reaction pathways in compartments of a cell as well as the dynamics of gene networks.
Further development of the MathCell project implies integration of individual components of the model into an integrated program system which would enable modeling of cell processes at all levels –from microscopic to macroscopic scales and from picoseconds to the scales comparable with the cell lifetime. Such modeling will naturally require combining of computational and commutation resources provided by EGEE project and their merging into an integrated computational medium.

2b: Data access on the grid / 25

The AMGA Metadata Service

Author: Birger Koblitz[1]

Co-author: Nuno Santos [1]

[1] *CERN-IT*

Corresponding Author: birger.koblitz@cern.ch

We present the ARDA Metadata Grid Application (AMGA) which is part of the gLite middleware. AMGA provides a lightweight service to manage, store and retrieve simple relational data on the grid, termed metadata.

In this presentation we will first give an overview of AMGA's design, functionality, implementation and security features. AMGA was designed in close collaborations with the different EGEE user communities and combines high performance, which was very important to the high energy physics community, with fine-grained access restrictions required in particular by the BioMedical community. These access restrictions also make full use of the EGEE VOMS services and are based on grid certificates. To show to what extent the users' requirements have been met, we will present performance measurements as well as show uses-cases for the security features.

Several applications are currently using AMGA to store their metadata. Among them are the MDM (Medical Data Manager) application implemented by the BioMedical community, the GANGA physics analysis tool from the Atlas and LHCb experimens and a Digital Library from the generic applications.

The MDM application uses AMGA to store relational information on medical images stored on the grid plus information on patients and doctors in several tables. User applications can retrieve images baded no their metadata for further processing. Access restrictions are of the highest importance to the MDM application because the stored data is highly confidential. MDM therefore makes use of the fine-grained access restrictions of AMGA.

The GANGA application uses AMGA to store the status information of jobs running on the grid which can be controlled by GANGA. AMGA's simple relational database features are mainly used to ensure consistency when several GANGA clients of the same user are accessing the stored information remotely.

Finally, the Digital Library project makes similar use of AMGA as the MDM application but provides many different schemas to store not only images but information on texts, movies or music. Another difference is that there is only a central librarian updating the library while for MDM updates are triggered by the many image acquisition systems themselves.

This presenation will also discuss future developments of AMGA, in particular its features to replicate or federate metadata. They will mainly allow users to make use of a better scaling behaviour but could also allow better security by using federation to physically seperate metadata. The replication features will be compared to current proprietary solutions.

AMGA provides a very lightweight metadata service
as well as basic database access functionality on the Grid.
After a brief overview of AMGA's design, functionality, implementation
and security features we will show performance comparisons of AMGA with
direct database access as well as other Grid catalogue services. Finally
the replication features of AMGA are presented and a comparison done
with proprietary database replication solutions.

1c: Earth Observation - Archaeology - Digital Library / 26

Requirements of Climate applications on Grid infrastructures; C3-Grid and EGEE

Author: Joachim Biercamp[1]

Co-authors: Kerstin Ronneberger [1]; Stephan Kindermann [1]

[1] *DKRZ*

Corresponding Author: ronneberger@dkrz.de

Human made climate change and its impact on the natural and socio-economic
environment is one of todays most challenging problems of mankind. To understand and
project processes, changes and impacts of the natural and socio-economic system a
growing community of researchers from various disciplines investigates and analyses
the earthsystem by means of computer simulation and analysis models.
These models are usually computational demanding and data intensive as they need to
compute and store high resolved 4-dimensional fields of various parameters. Moreover,
the required close collaboration in interdisciplinary and often also international
research projects involves intensive community interactions.
To support climate workflows the community established proprietary, mostly national
or regional solutions, which are normally grouped around centralized high performance
computing and storage resources. Homogeneous discovery of and access to climate data
sets residing in distributed petabyte climate archives as well as distributed
processing and efficient exchange of climate data are the central components of
future international climate research. Thus, the EGEE infrastructure potentially
offers a highly suitable environment for such applications.

However, existing grid infrastructures - including EGEE - do not yet meet the requirements of the climate community essential for prevalent workflows. Hence, to port existing applications and workflows on the EGEE infrastructure, a stepwise extension of the infrastructure to community specific services is needed. Moreover, the identification and demonstration of feasibility and added value is essential to convince the community to change their established habits. The Collaborative Climate Community Data and Processsing Grid (C3-Grid [1]) is an application driven approach towards the deployment of GRID techniques for climate data analysis. Solutions currently developed in this project offer a potentially fruitful basis to improve the suitability of the EGEE infrastructure as a platform for data analysis within climate research.

Within EGEE climate is part of the Earth Science Research (ESR) VO. We evaluated and tested the use of the EGEE infrastructure for climate applications [4]. As part of this prototypes of simulation as well as analysis software were tested on the EGEE infrastructure. We identified 3 different accesspoints for pilot applications, that can demonstrate the potential benefit of the EGEE infrastructure for climate research: Ensemble simulations with models of intermediate complexity, coupling experiments on a common platform and data sharing and analysis.

Ensembles of simulations performed with the same model but different future scenarios and different parameterisations are required to quantify the uncertainty and possible variety of future climate predictions. EGEE offers a good infrastructure for such ensemble simulations with models of intermediate complexity, which do not need the performance of a supercomputer. Ensembles can be submitted as DAG, parametric or collection job and results could be directly stored, analysed and reduced to the required information on the grid.

The coupling of diverse models of different disciplines is essential to understand the interaction and feedback between the different climate and earth system components, as e.g. the human impact on future climate development. In corresponding projects partners from different institutes of different nations are collaborating on a common modeling framework. The EGEE infrastructure would be a valuable platform for such coupling approaches. Data, models and output could be easily shared, different access and user rights can be established via VOMS. Currently different coupling tools are explored to assess their "grid-suitability".

Data sharing and analysis is a central aspect in climate research. The enormous amounts of data, produced by the model simulations need to be analysed, visualised and validated against observations or other data sources to be correctly interpreted. This involves a multiplicity of statistical calculations carried out on samples of different large data files. Currently such data analysis is centred around the heterogeneous database systems, which are accessed via non-standardised metadata. Thus, the establishment of a common data exchange and management infrastructure bridging the existing heterogeneous community datamanagement solutions with the EGEE data management system would add great value to such applications.

Especially for the realisation of climate data sharing and analysis workflows on EGEE the following components need to be developed:

1) a common agreed upon metadata schema for discovery of climate data sets stored in grid file space as well as in external community datacenters
2) a common community metadata catalogue based on this schema
3) common interfaces to reference and access grid external data resources (mainly databases)

All of these aspects are addressed within the recently introduced national German C3Grid [1] project within the German e-science (D-Grid [2]) initiative which aims to develop a grid middleware specific for the needs of the climate research community. Within this project a common metadata schema is defined. A community metadata catalogue and information system is established and a common data access interface will be defined.

To promote EGEE as a climate data handling (and postprocessing) infrastructure based on these developments we propose a stepwise approach:

- establishment of an international standards based climate metadata catalog (e.g. based on AMGA plus a common push/pull metadata exchange to grid external metadata catalogues via established metadata harvesting protocols

- establishment of data access to (initially free) climate datasets in climate data centers: As intial starting point we need an easy way to access data in climate data centers and copy/register them on grid storage, e.g. by using proprietary access clients or OGSA-DAI.
- adaptation of commonly used climate data processing toolkits on EGEE such as e.g. cdo [3]

[1] http://www.c3grid.de
[2] http://www.d-grid.de
[3] http://www.mpimet.mpg.de/˜cdo/
[4] Stephan Kindermann, EGEE infrastructure and Grids for Earth Sciences and Climate Research, Technical report DKRZ (available under http://c3grid.dkrz.de/moin.cgi/PublicDocs)

Poster and Demo session + cocktail / 27

Demonstration of the P-GRADE portal

Author: Peter Kacsuk[1]

Co-author: Gergely Sipos [1]

[1] *MTA SZTAKI*

Corresponding Author: sipos@sztaki.hu

The P-GRADE portal plays more and more important role in the EGEE community. After its successful demos in the previous EGEE conferences (Athens and Pisa) the representatives of several EGEE VOs have approached us with the request to support their users by the P-GRADE portal that is already the official portal of two EGEE VOs: VOCE (Virtual Organization Central Europe) and HunGrid (Hungarian VO of EGEE). Besides, P-GRADE portal is the official portal of SEEGRID which is a 100% EGEE-based Grid infrastructure serving all the countries of the South-East European region (even those countries that were not members of EGEE-1). After the Pisa demo the EGRID VO established a P-GRADE portal to support their activity and the biomed community showed interest to connect the portal to their workflow management engine. Besides the EGEE community, the portal is successfully used as service for the UK National Grid Service (NGS) and it was also successfully connected to the GridLab testbed as well as to the Hungarian ClusterGrid. After its successful demonstration at the Supercomputing'05 exhibition representatives of the US Open Science Grid also expressed their interest to connect the portal to their Grid.

Why is P-GRADE portal so successful? The main reason is that it is a generic workflow-oriented portal that can support all the important features the typical end-users would like to have:

1. Hidden low-level Grid details but at the same time enabling the access of any important feature of the underlying Grid
2. Easy porting of the applications to the Grid
3. User-friendly, graphical environment to control and observe the execution of the Grid application
4. Enabling the usage of MPI programs in the Grid
5. Enabling the usage of legacy codes in the Grid
6. Developing and executing workflow applications in the Grid
7. Combining MPI and legacy programs in workflows
8. Developing and executing parametric study applications (both at job and workflow level) in the Grid

9. Providing parallel execution mechanisms for the workflows at various levels a. intra-job b. inter-job c. pipe-line d. multi-thread
10. Supporting multi-Grid access mechanism and inter-Grid parallelism
11. Providing a secure and robust Grid application development and execution service for end-users (including certificate management, quota management and resource management)
12. Providing user-centric error messages and workflow recovery mechanism in case of erroneous job and workflow execution.
13. Providing autonomous error correction facilities
14. Supporting collaborative workflow development and execution
15. Tailoring the portal to specific user needs

The current version of P-GRADE portal (version 2.3) can provide features 1-4, 6, 9/a, 9/b,10-12, 15. The UK NGS extension of the portal can provide features 5 and 7. Feature 14 is already prototyped and demonstrated at the Supercomputing'05 exhibition. This feature will be available as service by November 2006. Features 8, 9/c and 9/d are under development as a joint work with the bioscience EGEE community and will be available in version 3.0 by April 2006. Version 3.0 will also support feature 13.

P-GRADE portal is based on the JSR168 compliant GridSphere 2 framework and hence it supports the easy extension and tailoring of the portal according to specific user needs. There are two examples for such extension of the portal. For the UK NGS, University of Westminster developed and added a new portlet that supports the definition and invocation of legacy code services. For the EGRID community, researchers of the Abdus Salam International Centre for Theoretical Physics have developed and now add a new portlet that enables file transfer among Grid computational and storage resources. In fact the further development of the portal is going on as a joint activity of several universities and institutes in Europe. Besides the above mentioned two collaborating partners, Univ. of Reading contributes to the creation of the collaborative version of the portal while CNRS collaborates with SZTAKI in creating the parametric study version of the portal. The Boskovic research institute in Zagrab developes specific application oriented portlets.

The goal of the demonstration of the P-GRADE portal is to demonstrate the features mentioned above. We shall use four portal installations during the demonstration. The VOCE portal (version 2.3) that runs as a service for VOCE will be used to demonstrate the robustness and scalability of the P-GRADE portal as a VO service. This demo tries to convince the audience that the current version of P-GRADE portal is robust and scalable and hence it can be used for any VO of EGEE as a stable service for end-users. This portal will be used to demonstrate features 1-4, 6, 9/a, 9/b,10-12.

The UK NGS portal (version 2.2) that runs as a service for UK NGS will be used to demonstrate how the portal can be extended with legacy code services as well as with application-specific portlets. Moreover we shall demonstrate the multi-Grid access mechanism of the portal showing that both the UK NGS and the HunGrid (EGEE) sites can be accessed by the same portal within a workflow in a simultaneous way realizing Grid interoperability and multi-Grid parallelism. This portal will be used to demonstrate features 5, 7, 10. Two experimental portals (prototypes) will also be demonstrated to show the future features of the portal (features 8, 9/c, 9/d and 14).

We hope that by continuing the successful series of portal demonstrations more and more EGEE user community will recognize the obvious advantages of using the portal instead of the low-level command-line user interface. The mass usage of Grid technology cannot be achieved by low-level commands, only high-level, graphical user interfaces can attract and convince the end-users that Grid is usable for them. P-GRADE portal is a step towards this direction.

Summary:

The current and new features of the P-GRADE portal will be presented. The current features are already built in to those portal releases that work as production services for several EGEE related VO (VOCE, HunGrid, EGRID) and SEEGRID as well as for other type of Grids like the UK NGS. The new features are under preparation and demonstrated as prototype systems. They will be put into service during this year.

1a: Life Sciences / **28**

Application of the Grid to Pharmacokinetic Modelling of Contrast Agents in Abdominal Imaging

Authors: Bernardo Celda[1]; Damià Segrelles[2]; Daniel Monleón[1]; David Moratal[3]; Ignacio Blanquer[2]; José Carbonell[2]; Luis Martí-Bonmatí[4]; Montserrat Robles[2]; Vicente Hernández[2]

[1] *Universidad de Valencia Estudi General*

[2] *Universidad Politécnica de Valencia*

[3] *David Moratalb*

[4] *Hospital Universitario Dr. Peset*

Corresponding Author: iblanque@dsic.upv.es

The liver is the largest organ of the abdomen and there are a large number of lesions affecting it. Both benign and malignant tumours arise within it. The liver is also the target organ for most solid tumours metastasis. Angiogenesis is quite an important marker of tumour aggressiveness and response to therapy. The blood supply to the liver is derived jointly from the hepatic arteries and the portal venous system. Dynamic Contrast Enhanced Magnetic Resonance Imaging (DCE-MRI) is extensively used for the detection of primary and metastatic hepatic tumours. However, the assessment of early stages of the malignancy and other diseases like cirrhosis require the quantitative evaluation of the hepatic arterial supply. To achieve this goal, it is important to develop precise pharmacokinetic approaches to the analysis of the hepatic perfusion. The influence of breathing, the large number of pharmacokinetic parameters and the fast variations in contrast concentration in the first moments after contrast injection reduce the efficiency of traditional approaches. On the other hand, the traditional radiological analysis requires the acquisition of images covering the whole liver, which greatly reduces the time resolution for the pharmacokinetic curves. The combination of all these adverse factors makes very challenging the analytical study of liver DCE-MRI data.
The final objective of the work we present here is to provide the users with a tool to optimally select the parameters that describe the farmacokinetic model of the liver. This tool will use the Grid as a source of computing power and will offer a simply and user-friendly interface.
The tool enables the execution of large sets of co-registration actions varying the values of the different parameters, easing the process of transferring the source data and the results. Since Grid concept is mainly batch (and the co-registration is not an interactive process due to its long duration), it must provide with a simply way to monitor the status of the processing. Finally the process must be achieved in the shorter time possible, considering the resources available.

Summary:

1. The Medical Scenario

1.1. Pharmacokinetic Modelling
The pharmacokinetic modelling of the images obtained after a quick administration of a bolus of extracellular gadolinium chelates contrast can have a deep impact on the

diagnosis and the evaluation of different pathogen entities.
Pharmacokinetic models are designed to forecast the evolution of an endogenous or exogenous component on the tissues. To follow-up the evolution of the contrast agent a sequence of MRI volumetric images is obtained at different times following the injection of contrast. Each of these images comprises a series of image slices that cover the body part explored. Since the whole process takes a few minutes, images are obtained in different break-hold periods. This movement of the patient produces artefacts that make images directly incomparable.
The study of pharmacokinetic models for the analysis of hepatic tumours is an outstanding example of the above. A prerequisite for the computation of the parameters that govern the model is the reduction of the deformation of the organs in the obtained images. This process can be performed by co-registering all the volumetric images with respect to the first one.

1.2. Co-registration
The co-registration of images consists on aligning the voxels of two or more images in the same geometrical space by using the necessary transformations to make the floating images as much as possible similar to the reference image.
In general terms, the registration process could be rigid or deformable. Rigid registration only uses affine transformations (displacements, rotations, scaling) to the floating images. Deformable registration enables the use of elastic deformations on the floating images. Rigid registration introduces fewer artefacts, but it can only be used when dealing with body parts in which the level of internal deformation is lower (e.g. the head). Deformable registration could introduce unrealistic artefacts, but is the only one that could compensate the deformation of elastic organs (e.g. in the abdomen). Registration in 3D is necessary in this case, since the deformation happens in the three axes. This is an extremely time-consuming process.

1.3 Post processing
Although the co-registration of images is a computationally complex process which must be performed before the analysis of the images, it is not the only task that needs high performance computing. Extracting the parameters that define the model and computing the transfer rates for each voxel in the space will require large computing resources.
The process of identification of the parameters consists on solving an over-determined non-linear system of equations for each voxel on the image. This process is computing intensive considering the amount of voxels of the images (on the order of 6 millions per study). This process is highly parallel.

1. Grid Application In order to provide the necessary computing power to solve the co-registration and the identification of the parameters, an application has been developed. This application has been implemented considering the following points: • Provide high performance. Be prepared to use a large Grid infrastructure to provide the computational power. • Usability. Reduce the complexity of the use of Grids by means of a user-friendly interface. This interface must be open to its integration in other applications. The choice is to implement a web-services based portal. • Security. The program must deal with the risks of using remote resources. Anonymisation is required and access control is very important. • Reliability. The application must provide production capability, so assistance to the Resource Broker must be provided on selecting the sites.

This Grid application uses a graphical user interface that calls web services that implement the different tasks. This application is based on previous developments [2][3].
• Creating the proxy on the grid. Users have a certificate stored in the UI. The private key is provided by the user and a proxy is created remotely on the UI.
• Transferring the data into the Grid. The user gets the images from local disks or a scanner, transferring them to the FTP server of the UI. The data is copied to the SEs.
• Select the ranges of the parameters to test. The users can select the range of the values of the input parameters (Maximum step length for the gradient descent optimisator, Maximum number of iterations for the optimiser, Initial scaling factor and Initial angle for deformation) that will be used for the jobs to be launched.
• Create the JDLs and define the arguments for the scripts of each job (One job per

registration and per combination of parameters).
• Run the jobs and retrieve the data to the Working Nodes.
• Monitoring of the evolution of a set of jobs.
• Downloading the output of all the jobs in a group with a single click. This implies downloading the results, which were uploaded in the Storage Resources by the jobs, to the UI, and from it back to the FTP server and the application.

1. Results
   The results obtained can be considered in terms of performance and scientific results. The results presented in this section are related to the images from a clinical trial with 20 patients obtained at the Hospital Dr. Peset for this work.
   Considering the performance, the required time to perform a registration of a volumetric image in a PIII at 866 Mhz with 512 MB of RAM is approximately 1 hour and 27 minutes. Considering that the complete study performed involved 20 patients the total cost would be 2331h 22m. Using a 20-procs computing farm the complete process took 132h 50m. The computational cost using the Grid was 17h 35m.
   The overhead of Grids is due to the use of secure protocols, remote and distributed storage resources and the scheduling overhead, which is in the order of minutes due to the monitoring policies which are implemented in a poll fashion.
   Regarding the scientific results obtained.

2. Conclusions and Future Plans
   The computing requirements for a reduced clinical trial of 20 patients exceeds the conventional computational capabilities of either a hospital or a research team.
   Moreover, the need for computing is not constant and only after the clinical trials.
   Thus, there is a need for a production platform with a high degree of reliability, such as EGEE [1]. On the other side, the use of gLite is justified by the need for access control in data and metadata and the improved metadata management.
   The work is being completed with the implementation of the model parameters computation on the Grid, which is also a time-consuming process and which could be easily speed-up by the use of the Grid.

3. References

[1] EGEE. Enabling Grids in E-SciencE. http://www.eu-egee.org
[2] I. Blanquer, V. Hernandez, D. Segrelles, M. Robles, J. M. Garcia, J. V. Robledo, "Clinical Decision Support Systems (CDSS) in Grid Environments". From Grid to HealthGrid (ISBN 1-58603-510-X), Vol: 112, pp: 80-89, 2005 IOS Press.
[3] Ignacio Blanquer, Vicente Hernández, Ferran Mas, Damià Segrelles, "A Framework Based on Web Services and Grid Technologies for Medical Image Registration". Lecture Notes in Bioinformatics, ISSN 0302-9743, vol 3745, pp 22-33, Sptringer 2005.

Poster and Demo session + cocktail / 29

Scientific data audification within GRID: from Etna volcano seismograms to text sonification

Author: Domenico Vicinanza[1]

[1] *Univ. of Salerno + INFN Catania*

Corresponding Author: domenico.vicinanza@cern.ch

Data audification is the representation of data by sound signals; it can be considered as the acoustic counterpart of data graphic visualization, a mathematical mapping of information from data sets to sounds.
Data audification is currently used in several fields, for different purposes: science and engineering, education

and training, in most of the cases to provide a quick and effective data analysis and interpretation tool.
Although most data analysis techniques are exclusively visual in nature (i.e. are based on the possibility of
looking at graphical representations), data presentation and exploration systems could benefit greatly from
the addition of sonification capacities. In addition to that, sonic representations are particularly useful when
dealing with complex, high-dimensional data, or in data monitoring tasks where it is practically impossible
to use the visual inspection. More interesting and intriguing aspects of data sonification concern the
possibility of describing patterns or trends, through sound, which were hardly perceivable otherwise. Two
examples, in particular, will be discussed in this paper, the first one coming from the world of geophysics and
the second one from linguistics.

Poster and Demo session + cocktail / 30

Meteorology and Space Weather Data Mining Portal

Author: Mikhail Zhizhin[1]

Co-authors: Alexey Poyda [2]; Dmitry Mishin [3]

[1] *Geophysical Center Russian Acad. Sci.*

[2] *Moscow State University*

[3] *Institute of Physics of the Earth Russian Acad. Sci.*

Corresponding Author: jjn@wdcb.ru

We will demonstrate an environmental data mining project Environmental Scenario Search Engine (ESSE) including a secure web application portal for interactive searching for events over a grid of environmental data access and mining web services hosted by OGSA-DAI containers. The web services are grid proxies for the database clusters with terabytes of high-resolution meteorological and space weather reanalysis data over the past 20-50 years. The data mining is based on fuzzy logic to make it possible to describe the searching events in natural language terms, such as "very cold day". The ESSE portal allows parallel data mining across disciplines for correlated events in space, atmosphere and ocean. The ESSE data web-services are installed in the USA, Russia, South Africa, Australia, Japan, and China. The EGEE infrastructure facilitates sharing of the environmental data and grid services with the European environmental sciences community. The work is done in cooperation with the National Geophysical Data Center NOAA and supported by the grant from the Microsoft Research Ltd.

2b: Data access on the grid / 31

Space Physics Interactive Data Resource - SPIDR

Authors: Eric Kihn[1]; Mikhail Zhizhin[2]

Co-authors: Dmitry Medvedev [2]; Dmitry Mishin [3]; Rob Redmon [1]

[1] *National Geophysical Data Center NOAA*

[2] *Geophysical Center Russian Acad. Sci.*

[3] *Institute of Physics of the Earth Russian Acad. Sci.*

Corresponding Author: jjn@wdcb.ru

SPIDR (Space Physics Interactive Data Resource) is a de facto standard data source on solar-terrestrial physics, functioning within the framework of the ICSU World Data Centers. It is a distributed database and application server network, built to select, visualize and model historical space weather data distributed across the Internet. SPIDR can work as a fully-functional web-application (portal) or as a grid of web-services, providing functions for other applications to access its data holdings.

Currently SPIDR archives include geomagnetic variations and indices, solar activity and solar wind data, ionospheric, cosmic rays, radio-telescope ground observations, telemetry and images from NOAA, NASA, and DMSP satellites. SPIDR database clusters and portals are installed in the USA, Russia, China, Japan, Australia, South Africa, and India.

SPIDR portal combines functionality from the central XML metadata repository with two levels of metadata, descriptive and inventory, with a set of distributed data source web services, web map services, and raw observations data files collections. A user can search for data using metadata inventory, use persistent data basket to save the selection for the next session, and to plot and download in parallel the selected data in different formats, including XML and NetCDF. A database administrator can upload new files into the SPIDR databases using either the web services or the web portal. SPIDR databases are self-synchronising. User support on the portal includes discussion forum, i-mail, data basket for metadata bookmarks and selected data subsets, and usage tracking.

SPIDR technology can be used for environmental data sharing, visualization and mining, not only in space physics, but also in seismology, GPS measurements, tsunami warning systems, etc. All grid data services in SPIDR share the same Common Data Model and compatible metadata schema.

Poster and Demo session + cocktail / 32

Secured Medical Data Management on the EGEE grid

Authors: Akos Frohner[1]; Birger Koblitz[1]; Christophe Pera[2]; Daniel Jouvenot[2]; Johan Montagnat[2]; Nuno Santos[1]; Peter Kunszt[1]

[1] *CERN*

[2] *CNRS*

Corresponding Author: johan@i3s.unice.fr

** Clinical data management versus computerized medical analysis

The medical community is routinely using clinical images and associated medical data for diagnosis, intervention planning and therapy follow-up. Medical imagers are producing an increasing number of digital images for which computerized archiving, processing and analysis are needed.

DICOM (Digital Image and COmmunication in Medicine) is today the most widely adopted standard for managing medical data in clinics. DICOM is including both the image content and additional information on the patient and the acquisition. DICOM was exclusively designed to respond clinical requirements. The interface with computing infrastructures for instance is completely lacking.

Grids are promising infrastructures for managing and analyzing the huge medical databases. However, the existing grid middlewares are often only providing low level data management services for manipulating files, making difficult the gridification of medical applications. Medical data often have to be manually transferred and transformed from hospital sources to grid storage before being processed and analyzed. To ease applications development there is a need for a data manager that: (i) shares access to medical data sources for computing without interfering with the clinical practice; (ii) ensures transparency so that accessing medical data does not require any specific user intervention; and (iii) ensures a high data protection evel to respect patients privacy.

** MDM: a grid service for secured medical data management

To ease medical applications devlopment, We developed a Medical Data Manager (MDM) service with the support of the EGEE uropean IST project. This service was developped on top of the new generation middleware release, gLite.

The data management in the gLite middleware is based on a set of Storage Elements which are exposing a same standard Storage Resource Manager SRM) interface. The SRM is handling local data at a file level. Additional services such as GridFTP or gLiteIO are coexisting on storage elements to provide transfer capabilities. In addition to storage resources, the gLite data management system includes a File Catalog (Fireman) offering a unique entry point for files distributed on all grid storage elements. Each file is uniquely identified through a Global Unique IDentifier (GUID).

The Medical Data Management service architecture is diagrammed in figure 1. On the left, is represented a clinical site: various imagers in an hospital are pushing the images produced on a DICOM server. Inside the hospital, clinicians can access the DICOM server content through DICOM clients. In the center of figure 1, the MDM internal logic is represented. On the right side, the grid services interfacing with the MDM are shown. To remain compatible with the rest of the grid infrastructure, the MDM service is based on an SRM-DICOM interface software which translates SRM grid requests into DICOM transactions addressed to the medical servers. Thus, medical data servers can be transparently shared between clinicians (using the classical DICOM interface inside hospitals) and image analysis scientists (using the SRM-DICOM interface to access the same data bases) without interfering with the clinical practice. An internal scratch space is used to transform DICOM data into files that are accessible through data transfer services (GridFTP or gLiteIO). For enforcing data protection, a highly secured and fault tolerant encryption key catalog, called hydra, is used. In addition, all DICOM files exported to the grid are anonimized. A metadata manager is in charge of holding the metadata extracted from DICOM headers and to ease data search. The AMGA ervice is used for ensuring secured storage of these very sensitive data. The AMGA server holds a relation between each DICOM slice and the image metadata.

The security model of the MDM relies on several components: (i) file access control, (ii) files anonymization, (iii) files encryption, and (iv) secured access to metadata. The user is coherently identified through a single X509 certificate for all services involved in security. The file access control is enforced by the gLiteIO service which accepts Access Control Lists (ACLs). The hydra key store and the AMGA metadata service both accept ACLs. To read an image content, a user needs to be authorized both to access the file and to the encryption key. The access rights to the sensitive metadata associated to the files are administrated independently. Thus, it is possible to grant access to an encrypted file only (e.g. for replicating a file without accessing to the content), to the file content (e.g. for processing the data without revealing the patient identity), or to the full file metadata (e.g. for medical usage). Through ACLs, it is possible to implement complex use cases, granting access rights to patients, physicians, healthcare practitioners, or researchers independently.

** Medical image analysis applications

On the client side, three levels of interfaces are available to access and manipulate the data hold by the MDM: (1) the standard SRM interface, can be used to access encrypted images provided that their GUID is known; (2) the encryption middleware layer can both fetch and decrypt files; (3) the fully MDM aware client provides access to the metadata associated to files in addition.

The Medical Data Manager has been deployed on several sites for testing purposes. Three sites are actually holding data in three DICOM servers installed at I3S (Sophia Antipolis, France), LAL (Orsay, France) and CREATIS (Lyon, France). An AMGA catalog has also been set up in CREATIS (Lyon) for holding all sites' metadata, and an hydra key store is deployed at CERN (Geneva, Switzerland).

The testbed deployed has been used to demonstrate the viability of the service by registering and retrieving DICOM files across sites. Registered files could be retrieved and used for computations from EGEE grid nodes transparently. The next important milestone will be to experiment the system in connection with hospitals by registering real clinical data freshly acquired and registered on the fly from the hospital imagers.

The Medical Data Manager is an important service for enabling medical image processing applications on the EGEE grid infrastructure. Several existing applications could potentially use the MDM such as the GATE, CDSS, gPTM3D, pharmokinetics, and Bronze Standard applications currently deployed on the EGEE infrastructure.

Summary:

This abstract describes the effort to deploy a Medical Data Management service on top of the EGEE grid infrastructure. The most widely accepted medical image standard, DICOM, was developed for fulfilling clinical practice. It is implemented in most recent medical image acquisition and analysis devices. The EGEE middleware is using the SRM standard for handling grid files. Our prototype is exposing an SRM compliant interface to the grid middleware, transforming on the fly SRM requests into DICOM transactions. The prototype ensures user identification, strict file access control and data protection through the use of relevant grid services. This Medical Data Manager is easing the access to medical databases needed for many medical data analysis applications deployed today. It offers a high level data management service, compatible with clinical practices, which encourages the migration of medical applications towards grid infrastructures. A limited scale testbed has been deployed as a proof of concept of this new service.

2a: Workload management and Workflows / 33

MOTEUR: a data intensive service-based workflow engine enactor

Authors: Johan Montagnat[1]; Tristan Glatard[1]; Xavier Pennec[2]

[1] *CNRS*

[2] *INRIA*

Corresponding Author: johan@i3s.unice.fr

** Managing data-intensive application workflows

Many data analysis procedures implemented on grids are not only based on a single processing algorithm but rather assembled from a set of basic tools dedicated to process the data, model it, extract quantitative information, analyze results, etc. Given that interoperable algorithms packed in software components with a standardized interface enabling data exchanges are provided, it is possible to build complex workflows to represent such procedures for data analysis. High level tools for expressing and handling the computation flow are therefore expected to ease computerized medical experiments development.

Workflow processing is a thoroughly researched area. Grid enabled application often need to process large datasets made of e.g. hundreds or thousand of data to be processed according to a same workflow pattern. We are therefore proposing a workflow enactment engine which:
- Makes the description of the application workflow simple from the application developer point of view.
- Enables the execution of legacy code.
- Optimizes the performances of data-intensive applications by exploiting the potential parallelism of the grid infrastructure.

** MOTEUR: an optimized service-based workflow engine

MOTEUR stands for hoMe-made OpTimisEd scUfl enactoR. MOTEUR is written in Java and available under CeCILL Public License (a GPL-compatible open source license) at http://www.i3s.unice.fr/~glatard.
The workflow description language adopted is the Simple Concept Unified Flow Language (Scufl) used by the Taverna and that is currently becoming a standard in the e-Science community.

Figure 1 shows the MOTEUR web interface representing a workflow that is being executed. Each service is represented by a color box and data links are represented by curves. The services are color coded depending on their current status: gray services have never been executed; green services are running; blue services have finished the execution of all input data available; and yellow services are not currently running but waiting for input data to become available.

MOTEUR is interfaced to the job submission interfaces of both the EGEE infrastructure and the Grid5000 experimental grid. In addition, lightweight jobs execution can be orchestrated on local resources. MOTEUR is able to submit different computing tasks on different infrastructures during a single workflow execution. MOTEUR is implementing an interface to both Web Services and GridRPC application services.

By opposition to the task-based approach implemented in DAGMan, MOTEUR is service-based. The services paradigm has been widely adopted by middleware developers for the high level of flexibility that it offers. Application services are similarly well suited for composing complex applications from basic processing algorithms. In addition, the independent description of application services and the data to be processed make this paradigm very efficient for processing large data sets. However, this approach is less common for application code as it requires all codes to be instrumented with the common service interface.

To ease the use of legacy code, a generic wrapper application service has been developed. This grid submission service is exposing a standard web interface and is controlling the submission of any executable code. It releases the user from the need to write a specific service interface and recompile its application code. Only a small executable invocation description file is required to enable the command line composition by the generic wrapper.

To enact different data-intensive applications, MOTEUR implements two data composition patterns. The data sets transmitted to a service can be composed pairwise (each input of the first input data set is processed with each input of the second one). This correspond to the case where the two input data sets are semantically connected. The data sets can also be fully composed (all inputs of the first set are processed with all inputs of the second one). The use of these two composition strategies significantly enlarges the expressiveness of the workflow language. It is a powerful tool for expressing complex data-intensive processing applications in a very compact format.

Finally MOTEUR enables 3 different levels of parallelism for optimizing workflow application code execution:
- workflow parallelism inherent to the workflow topology;
- data parallelism: different input data can be processed independently in parallel;
- services parallelism: different services processing different data are independent and can be executed in parallel.

To our knowledge, MOTEUR is the first service-based workflow enactor implementing all these optimizations.

** Performance analysis on an image registration assessment application

Medical image registration algorithms are playing a key role in a very large number of medical image analysis procedures. They are fundamental processings often needed prior to any subsequent analysis. The Bronze Standard application (http://egee-na4.ct.infn.it/biomed/BronzeStandard.html) is a statistical procedure aiming at assessing the precision and accuracy of different registration algorithms. The complex application workflow is illustrated in figure 1. This data-intensive application requires the processing of as much input image pairs as possible to extract relevant statistics.

The Bronze Standard application has been enacted on the EGEE infrastructure through the MOTEUR workflow execution engine. A 126 image pairs data base, courtesy of Dr Pierre-Yves Bondiau (cancer treatment center “Antoine Lacassagne”, Nice, France), was used for the computations. In total, the workflow execution resulted in 756 job submissions. The different levels of optimization implemented in MOTEUR permitted a speed-up higher than 9.1 when compared to a naive execution of the workflow.

Such data intensive applications are common in the medical image analysis community and there is an increasing need for compute infrastructure capable of efficiently processing large image databases. MOTEUR is a generic workflow engine that was designed to efficiently process data intensive workflows. It is freely available for download under a GPL-like license.

Summary:

In this abstract we introduce MOTEUR, a service-based workflow engine optimized for dealing with data intensive applications. MOTEUR eases the enactment of applications with complex data flow patterns on the EGEE production infrastrucutre. It is a generic workflow engine, based on current standards and freely available, that can be used to instrument legacy application code at low cost. Performances are demonstrated on a real medical image analysis application.

1c: Earth Observation - Archaeology - Digital Library / 34

International Telecommunication Union Regional Radio Conference and the EGEE grid

Author: Andrea Manara[1]

Co-authors: D. Botha ; M. Cosic ; P.N. Hai ; T. Gavrilov

[1] *ITU BR*

The Radiocommunication Bureau of the ITU (ITU-BR) manages the preparations for the ITU Regional Radio Conference RRC06 to establish a new frequency plan for the introduction of digital broadcasting (band III and IV/V) in Europe, Africa, Arab States and former-USSR States. During the 5 weeks of the RRC06 Conference (15 May to
16 June 2006) delegations from 119 Member States will negotiate the frequency plan.

The frequency plan will be established in an iterative way. During week time at the RRC06 administrations will negotiate and submit their requirements to the ITU-BR, which will conduct over the subsequent weekend all the calculations (analysis and synthesis) that would result in assigning specific frequencies for the draft plan. The output of the calculations will be the input for negotiations in the subsequent week, with the last iteration constituting the basis for the final frequency plan. In
addition, partial calculations are envisaged for parts of the planning area in between two global iterations (for the entire planning area).

For obtaining optimum planning of the available frequency spectrum, two different software processes have been developed by the European Broadcasting Union and they are run in sequence: compatibility assessment and plan synthesis. The compatibility assessment (which is very CPU demanding and can be run on a distributed infrastructure) calculates the interference between digital requirements, analogue broadcasting and other services stations. The plan synthesis assigns channels to requirements which could share the same channel.

The limited time to perform the calculation calls for the optimization of the process. The turnaround time to provide a new set of results would be a critical factor for the success of the Conference. The EGEE grid will greatly enhance the ITU-BR available resources allowing better serving the Conference. The grid infrastructure will complement the client-server distributed system developed within the ITU-BR, which has been used for the first exercises. In addition, the

possibility
to perform faster calculations could improve the efficiency of the negotiation (for example, giving preliminary results during the negotiation weeks themselves or allow extra quality checks and compatibility studies).

The compatibility assessment consists in running a large number of jobs (some tens of
thousands). Each job is basically the same application running on different datasets representing the parameters of radio-stations. One should note that the execution time varies by more than 3 orders of magnitudes (the majority of jobs needs only few seconds but few jobs require many hours) depending on the input parameters and cannot
be completely predicted. To cope with this situation we decided to use a client-server system called DIANE that allows run-time load balancing, access to heterogeneous resources (Grid and local cluster at the same time) and a robust infrastructure to cope with run-time problems. In the DIANE terminology, a job is defined as a "task". DIANE allows using in the most effective way the available resources since each available worker nodes asks for the next task: while a long task
will "block"a node, in the mean time the short tasks (the large majority) will flow through the other nodes.

We have already demonstrated to be able to perform the required calculations on the EGEE/LCG infrastructure (in the first tests, we have run with a parallelism of the order of 50, observing the expected speed-up factor) and we are preparing, in close collaboration with CERN, to use these techniques during the Conference later this year. The EGEE infrastructure does not only enable us to give the adequate support for an important international event but, in addition, the substantial speed-up already observed opens the possibility to allow faster and more detailed studies during the Conference. The technical improvement gives the possibility to provide a better service and technical data to the Conference's delegates.

The present set up is well suited for the foreseen application. The possibility to access resources from the grid and corporate resources (which we are not yet exploiting) is very appealing and should be interesting for other users. The possibility to describe and execute more complex workflow (presently we are using the
system to execute independent tasks in parallel) could increase the interest for the tools we are currently using.

Poster and Demo session + cocktail / 35

Internal Virtual Organizations in the RDIG-EGEE Consortium

Authors: Andrey Demichev[1]; Elena Tikhonenko[2]

Co-authors: Mikhail Zhizhin [3]; Slava Ilyin [1]; Vladimir Dobretsov [4]; Vladimir Voznesensky [4]

[1] *SINP*

[2] *Joint Institute for Nuclear Research (JINR)*

[3] *GC RAN*

[4] *RCC KI*

Corresponding Author: elena.tikhonenko@cern.ch

In the beginning of 2005 the formal procedures and the proper administrative structures for creation and registration of the internal RDIG-EGEE virtual organizations were established in the Russian Data Intensive Grid (RDIG)

consortium.
The Service Center of Registration of the Virtual Organizations is accessible through the URL: http://rdig-registrar.sinp.msu.ru/newVO.html . All the documents and rules, the basic document, in particular - "Creation and Registration of Virtual
Organizations in the frames of the RDIG-EGEE: Rules and Procedure"(in Russian), and
the Questionnaire examples can be found there (http://rdig-registrar.sinp.msu.ru/VOdocs/newVOinRDIG.html). The Council on RDIG-EGEE extension has been formed. The Council inspects all the new requests for new virtual organizations to be created.
The aim of the creation of the RDIG-EGEE virtual organizations is to serve the
national scientific projects and to test new application areas prior to including them into the global EGEE infrastructure. Nowadays we have 6 RDIG-EGEE internal virtual organizations with 42 members in them. Brief information on the Fusion VO for ITER project activities in Russia, eEarth VO for geophysics and cosmic research tasks (http://www.e-earth.ru/), and PHOTON VO for PHOTON and SELEX experiments (http://egee.itep.ru/PHOTON/index29d5en.html) is presented in poster.

1c: Earth Observation - Archaeology - Digital Library / 36

On-line demonstration of Flood application at EGEE User Forum

Author: Viet Tran[1]

[1] *Institute of Informatics, Slovakia*

Corresponding Author: viet.ui@savba.sk

The flood application has been successfully demonstrated at EGEE second review in December and we would demonstrate it at EGEE User forum for Grid application developers and Grid users.

Flood application consists of several numerical models of meteorology, hydrology and hydraulics. A portal is developed for comfortable use of flood application. The portal has four main modules:
• Workflow management module: for managing execution of tasks with data dependences
• Data management module: allows users to search and download data from storage elements
• Visualization module: show the output from models in several forms: text, picture, animation and virtual reality
• Collaboration module: allows users to communicate with each other and cooperate on flood forecasting

The demonstration will be done on GILDA demonstration testbed. Job execution in the Grid tested will be performed using gLite middleware. The aim of the demonstration is to show how to implement complicate grid applications with many models and support modules and also the FloodGrid portal, that allows users to run the application without knowledge about grid computing

Summary:

The flood application has been successfully demonstrated at EGEE second review in December and we would demonstrate it at EGEE User forum for Grid application developers and Grid users. The demonstration will be done on GILDA demonstration testbed. Job execution in the Grid tested will be performed using gLite middleware. The aim of the demonstration is to show how to implement complicate grid applications with many models and support modules and also the FloodGrid portal,

that allows users to run the application without knowledge about grid computing

2d: VO tools - Portals / 37

Supporting legacy code applications on EGEE VOs by GEMLCA and the P-GRADE portal

Author: Peter Kacsuk[1]

Co-authors: Gergely Sipos [1]; Tamas Kiss [2]

[1] *MTA SZTAKI*

[2] *University of Westminster*

Corresponding Author: sipos@sztaki.hu

Grid environments require special grid-enabled applications capable of utilising the underlying middleware services and infrastructures. Most Grid projects so far have either developed new applications from scratch, or significantly re-engineered existing ones in order to be run on their platforms. This practice is appropriate only in the context where the applications are mainly aimed at proving the concept of the underlying architecture. However, as Grids become stable and commonplace in both scientific and industrial settings, a demand will be created for porting a vast legacy of applications onto the new platform. Companies and institutions can ill afford to throw such applications away for the sake of a new technology, and there is a clear business imperative for them to be migrated onto the Grid with the least possible effort and cost.

Grid computing has reached the point where reliable infrastructures and core Grid services are available for various scientific communities. However, not even the EGEE Grid contains any tool to support the turning of legacy applications into Grid services that provide complex functions on top of the core Grid layer. The Grid Execution Management for Legacy Code Architecture (GEMLCA), presented in this paper, enables legacy code programs written in any source language (Fortran, C, Java, etc.) to be easily deployed on the EGEE Grid as a Grid service without significant user effort. GEMLCA does not require any modification of, or even access to, the original source code. A user-level understanding, describing the necessary input and output parameters and environmental values –such as the number of processors or the job manager required –is all that is needed to port the legacy application binary onto the Grid. Moreover, since GEMLCA has been integrated with the P-GRADE Portal, end-users can publish legacy applications as Grid services and can invoke legacy code services as a special kind of job (node) inside their workflows by an easy to use graphical portal interface.

The GEMLCA - P-GRADE Portal has been operating for the UK NGS community as a service since September 2005. Recently, the researchers of the University of Westminster and MTA SZTAKI have developed the EGEE-specific version of this tool. The EGEE-specific GEMLCA P-GRADE Portal offers the same legacy code management and workflow-oriented application development and execution facilities for EGEE research communities that have been provided on the UK NGS for more than six months now.

On top of the JSR-168 compliant portlets of the P-GRADE Portal (credential management, workflow enactment, etc) the GEMLCA-specific version contains an additional portlet that can be used to turn legacy applications into Grid services and to offer these services to other users of the portal. These users can invoke the legacy code services with their own custom input data, moreover, they can integrate legacy code services with newly developed codes inside their workflows. The portal environment contains a GEMLCA-specific editor to help users define such workflows. The workflow enactment service integrated into the Portal is

capable to forward job submission and legacy code service invocation requests to appropriate providers. While the core EGEE sites are responsible for job execution, the "legacy code repository"component of the portal server handles legacy code invocation requests.
This centralised repository provides opportunity for portal users to share applications with each other. The facility is a natural step to extend the concept of Virtual Organizations (VO). While the storage services of the EGEE Grid provide storage space for VO members in order to share data with each other, the code repository component of the GEMLCA P-GRADE Portal provides facility for VO members to share applications with each other. Moreover, since the P-GRADE Portal can be connected to multiple VOs at the same time, application sharing among the members of different VOs can take place through the Portal.
According to the current notion of EGEE the Grid is separated into research domain specific VOs, each of them containing relatively small number of resources. This concept simply prohibits two scientists working on two different scientific domains to collaborate with each other. Because these researchers are members of two different VOs there is no way for them to share applications with each other. However, by publishing their applications in the "legacy code repository"component of the GEMLCA P-GRADE Portal they can share these codes with other members of the whole EGEE community. This facility paves the way for revolutionary results in interdisciplinary research.

Besides the GEMLCA P-GRADE Portal the presentation will introduce an urban traffic simulation application developed on the EGEE Grid using this tool.
The traffic simulation is based on a workflow consisting of three types of components. The Manhattan legacy code (component 1) is an application to generate inputs for the MadCity simulator: a road network file and a turn file. The MadCity road network file is a sequence of numbers, representing a road topology of a road network. The MadCity turn file describes the junction manoeuvres available in a given road network. Traffic light details are also included in this file. MadCity (component 2) is a discrete-time microscopic traffic simulator that simulates traffic on a road network at the level of individual vehicles behaviour on roads and at junctions. After completing the simulation, a macroscopic trace file, representing the total dynamic behaviour of vehicles throughout the simulation run, is created. Finally a traffic density analyser (component 3) compares the traffic congestion of several runs of the simulator on a given network, with different initial road traffic conditions specified as input parameters. The component presents the results of the analysis graphically.
The lecture will use this application to describe how portal users can integrate their domain-specific applications into a large distributed program to solve the complex problem of traffic simulation. This example will present the benefits of portal-based collaborative work on the EGEE.

Summary:

Integrating GEMLCA and the P-GRADE portal enables EGEE users to use their exisiting legacy codes as Grid services without writing any wrapper or modification for the existing legacy codes. More than that such legacy codes can be used as components of workflows, combining them into complex Grid applications. Furthermore, the integrated GEMLCA/P-GRADE portal enables the transformation of any executed Grid job into a legacy service that can be easily used by other members of the VO community.

Poster and Demo session + cocktail / 38

Demo: LHCb data analysis using Ganga

Author: Andrew Maier[1]

[1] *CERN*

Corresponding Author: andrew.maier@cern.ch

The ARDA-LHCb prototype activity is focusing on the GANGA system (a joint ATLAS-LHCb project). The main idea behind GANGA is that the physicists should have a simple interface to their analysis programs. GANGA allows preparing the application, to organize the submission and gather results via a clean Python API. The details needed to submit a job on the Grid (like special configuration files) are factorised out and applied transparently by the system. In other words, it is possible to set up an application on a portable PC, then run some higher-statistics tests on a local facility (like LSF at CERN) and finally analyse all the available statistics on the Grid just changing the parameter which identifies the execution back-end.

1a: Life Sciences / **39**

Application of GRID resource for modeling charge transfer in DNA

Author: Nadezhda Fialko[1]

Co-authors: Alexandr Zaytsev [1]; Victor Lakhno [2]

[1] *research fellow*

[2] *director*

Corresponding Author: fialka@impb.psn.ru

Recently, at the interface of physics, chemistry and biology, a new and rapidly developing research trend has emerged concerned with charge transfer in biomacromolecules. Of special interest to researchers is the electron and hole transfer along a chain of base pairs, since the migration of radicals over a DNA molecule plays a crucial role in the processes of mutagenesis and carcinogenesis. Moreover, understanding the mechanism of charge transfer is necessary for the development of a new field, concerned with charge transfer in organic conductors and their possible application in computing technology.
To use biomolecules as conductors, one should know the rate of charge mobility. We calculate theoretical values of charge mobility on the basis of a quantum-classical model of charge transfer in various synthesized polynucleotides at varying temperature T of the environment. To take into account temperature fluctuations, a random force with specified statistical characteristics was added in the classical equations of site motion (Langevin force). (See e.g.: V.D.Lakhno, N.S.Fialko. Hole mobility in a homogeneous nucleotide chain // JETP Letters, 2003, v.78 (5), pp.336-338; V.D.Lakhno, N.S.Fialko. Bloch oscillations in a homogeneous nucleotide chain // Pisma v ZhETF, 2004, v.79 (10), pp.575-578).
As is known, the results of most biophysical experiments are averaged (for example, in our case, over a great many DNA fragments in a solution) values of macroscopic physical parameters. When modeling charge transfer in a DNA at finite temperature, calculations should be carried out for a great many realizations so that to find average values of macroscopic physical parameters. This formulation of the problem enables paralleling of the program by realizations such as "one processor –one realization".
A sequential algorithm is used for individual realizations. Initial values of site velocities and displacements are preset randomly from the requirement of equilibrium distribution at a given temperature. In calculating individual realizations, at each step a random number with specified characteristics is generated for the Langevin term.
To make the problem of modeling of the charge transfer in a given DNA sequence at a prescribed temperature suitable to be calculated using GRID resource, the original program was divided into 2 parts.
The first program calculates one realization for given parameters. At the input it

receives files with parameters and initial data. The peculiarity of the task is that we are interested in dynamics of charge transfer, so at the program output we get several dozens Mb results.
Using a special script, 100-150 copies of the program run with the same parameters and random initial data. Upon completion of the computations, the files of results are compressed and transmitted to a predefined SE.
When an appropriate number of realizations is calculated, the second program runs once. It must calculate average values for charge probabilities, for site displacements from the equilibrium, etc.
A special script is sent to calculate this program on WN. This WN takes from SE files with results of realizations in series of 10 items. For each series the averaging program runs (at the output one gets the data averaged over 10 realizations). If the output file of a current realization is absent or defective, it is ignored, and the next output file is taken. The files obtained are processed by this averaging program again. This makes our results independent of chance failures in calculations of individual realizations.
Using GRID resource by this method, we have carried out calculations of the hole mobility at different temperatures in the range from 10 to 300 K for (GG) and (GC) polynucleotide sequences (several thousands realizations).

1a: Life Sciences / 40

SIMRI@Web : An MRI Simulation Web Portal on EGEE Grid Architecture

Author: Fabrice BELLET[1]

Co-authors: Christophe PERA [1]; Hugues BENOIT-CATTIN [1]; Ioana NISTOREANU [1]

[1] *CREATIS - UMR CNRS 5515 - U630 Inserm*

Corresponding Author: yougz@creatis.insa-lyon.fr

In this paper, we present a web protal that enables simulation of MRI images on the grid. Such simulations are done using the SIMRI MRI simulator that is implemented on the grid using MPI. MRI simulations are useful for better understanding the MRI physics, for studying MRI sequences (parameterisation), and validating image processing algorithms. The web portal client/server architecture is mainly based on a java thread that screens a data base of simulation jobs. The thread submits the new jobs to the grid, and updates the status of the running jobs. When a job is terminated, the thread sends the simulated image to the user. Through a client web interface, the user can submit new simulation jobs, get a detailed status of the running jobs, have the history of all the terminated jobs as well as their status and corresponding simulated image.
As MRI simulation is computationally very expensive, grid technologies appear to a real added value for the MRI simulation task. Nevertheless the grid access should be simplified to enable final user running MRI simulations. That is why we develop a tis specific web portal to propose a user friendly interface for MRI simulation on the grid.

Summary:

*** Application context ***
The simulation of Magnetic Resonance Imaging (MRI) is an important counterpart to MRI acquisitions [1]. Simulation is naturally suited to acquire theoretical understanding of the complex MR technology. It is used as an educational tool in medical and technical environments. By offering an analysis independent of the multiple parameters involved in the MR technology, MRI simulation permits the investigation of artifact causes and effects. Simulation may also help in the development and optimization of MR sequences. Finally MRI simulator provides an interesting assessment tool of image processing techniques since it generates 3D

realistic images from medical virtual objects perfectly known.
The SIMRI simulator is a recent 3D MRI advanced simulator [1] that integrates in a unique simulator most of the simulation features that are offered in different simulators. It takes into account the main static field value and enables realistic simulations of the chemical shift artifact including off-resonance phenomena. It also simulates the artifacts linked to the static field inhomogeneity like those induced by susceptibility variation within an object. It is implemented in the C language and distributed under the CECILL license. The MRI sequence programming is done using high level C functions with a simple programming interface. To manage large simulations, the magnetization kernel is implemented in a parallelized way that enables simulation on PC grid architecture [2].

*** Grid added value ***
Since simulation of the MR physics is computationally very expensive, parallel implementation is mandatory to achieve performances compatible with the target applications. As an example it takes 12 hours to simulate a 512^2 image on a recent PC. This time has to be multiplied by 16 for a 1024^2 image. In 3D, simulation of a 5123 volume would require 100 years !
Thanks to the linearity property of the main computation task, the simulation job can be distributed easily with almost no communication between nodes during simulation [2]. As a consequence, the computation time is reduced in proportion with the available computation nodes. In this context, by offering a virtually unlimited computing power, grid technologies appear to a real added value for the MRI simulation task. Nevertheless the grid access should be simplified to enable final user running MRI simulations. That is why we develop a specific web portal to propose a user friendly interface for MRI simulation on the grid.

*** Experience, results and perspectives ***
The end user functionalities of the MRI web portal are the following:
- Full access to the MRI simulation parameters.
- Access to MRI simulation on a local cluster as well as on the EGEE grid.
- User authentication.
- Enhanced user job history.
- Enhanced running job status.
- Simulation results sent by mail.

The client interface has been developed in PHP5, HTML. The server side is running on a web server Apache V. 2.0.54. It has been developed using :
- PHP5 including the libraries libssh2.so et mysql.so.
- MySql v.4.
- Java 1.4.2 including the libraries jsch.jar,mysql-connector-java-3.jar.

The web server add each new job submission in a database that contains all the parameters of the jobs, and all the user description. Iteratively, a Java thread screens the job table of the data base. For each new job, the thread submits the job to the required target using LCG2 for EGEE grid or PBS for the local cluster. For all the running jobs, the thread asks the job status and updates the data base. For all the terminated jobs, the thread gets the simulated images from the corresponding platform, saves it in the data base, and sends it by mail to the user.
Each user has access to in personnal account on the web and gets the status of his running jobs as well as the history of all his simulations.
This portal is effective since september 2005. At the moment ,it is opened only to the 6 persons involved in the SIMRI project. After 300 simulations, we observed a job failure rate on the grid of about 20 %.
Our main perspective for this year is to develop a new web portal architecture that would use the web service functionalities of Glite middleware. We target a versatile and open architecture to be able to add easily in the portal new simulation target like the CINES machines and to add other MRI simulation codes like the one linked to susceptibility effect.

*** Key issue of the grid for the application area ***
In the context of this simulation application, the main key issues of the grid are the following:

- Efficient MPI implementation on the grid clusters.
- Improvement of the robustness of the grid to guarantee almost 100% of job success.
- Facilitation of the server certificate usage.
- Enabling in a transparent way job submission to multiple clusters in order to be able to use a large number of nodes for 3D simulation.

*** References ***
[1] H. Benoit-Cattin, F. Bellet, J. Montagnat,C. Odet, "Magnetic Resonance Imaging (MRI) simulation on a grid computing architecture,"Proc. IEEE CGIGRID'03-BIOGRID'03, Tokyo, 2003.
[2] H. Benoit-Cattin, G. Collewet, B. Belaroussi, H. Saint-Jalmes,C. Odet, "The SIMRI project: A versatile and interactive MRI simulator," Journal of Magnetic Resonance, vol. 173, pp. 97-115, 2005.

1b: Astrophysics/Astroparticle physics - Fusion - High-Energy physics / 41

Benefits of the MAGIC Grid

Author: Harald Kornmayer[1]

Co-authors: Andreu Pacheco [2]; Ariel Garcia [3]; Ciro Bigongiari [4]; Esther Accion [2]; Gonzalo Merino [2]; Jose A,. Coarasa [5]; Manuel Delfino [2]; Mirco Mazzucato [6]

[1] *FORSCHUNGSZENTRUM KARLSRUHE (FZK)*

[2] *PIC*

[3] *FZK*

[4] *INFN*

[5] *MPPMU*

[6] *CNAF/INFN*

Application context and scientific goals

The field of gamma-ray observations in the energy range between 10 GeV and 10 TeV developed fast over the last decade.
From the first observation of TeV gamma rays from the Crab nebula using the atmospheric Cerenkov imaging technique in 1989 [1] to the discovery of new gamma ray sources with the new generation telescopes like the HESS observation of a high-energy particle acceleration in the shell of a supernova remnant [2], a new observation window to the universe was opened.
In the future other ground based VHE γ-ray observatories (namely MAGIC [3], VERITAS [4] and KANGAROO [5]) will significantly contribute to the exploitation of this new observation window.
With the new generation Cerenkov telescopes the requirements for the analysis and Monte Carlo production computing infrastructure will increase due to a higher number of camera pixels, faster FADC systems and a bigger mirror size.
In the future the impact of VHE gamma-ray astronomy will increase by joined observations of different Cerenkov telescopes.

In 2003 the national Grid centers in Italy (CNAF), Spain (PIC) and Germany (GridKA) started together with the MAGIC collaboration an effort to build a distributed computing system for Monte Carlo generation and analysis on top of existing Grid infrastructure.

The MAGIC telescope was chosen due to the following reasons:
o The MAGIC collaboration is international, with most partners from Europe
o main partners of the MAGIC telescope are located close to the national Grid centers
o The generation of Monte Carlo data is very compute intensive, specially to get enough statistics
in the low energy range.
o The analysis of the fast increasing real data samples will be done in different institutes. The collaborators need a seamless access to the data while reducing the number of
replicas to a minimum.
o The MAGIC collaboration will build a second telescope in 2007 resulting in a doubled data rate.

The idea of the MAGIC Grid [6] was presented to the EGEE Generic Application Advisory Panel (EGAAP).
In June 2004 EGEE accepted the generation of Monte Carlo data for the MAGIC telescope as one of the generic applications of the project.

Grid added value

By implementing the MAGIC Grid over the last two years, the MAGIC collaboration benefit in many aspects. These aspects are described in this chapter.

o Collaboration of different institutes
By combining the resources of the MAGIC collaborators and the reliable resources from the national Grid centers the MAGIC collaborators will be empowered to use their computing infrastructure more efficiently. The time to analyse the big amount of data to solve specific scientific problems will be shortend.

o Cost reduction
By using the EGEE infrastructure and the EGEE services the effort for MAGIC collaboration to build a distributed computing system for the Monte Carlo simulations was significantly reduced.

o Speedup of Monte Carlo production
As the MAGIC Monte Carlo System was build on top of the EGEE middleware the integration of new computing resources is very easy. By getting support from many different EGEE resource providers the production rate for the Monte Carlos can be increased very easily.

o Persistent storage of observation data
The MAGIC telescope will produce a lot of data in the future. These data are currently stored on local resources including disk systems and tape libraries. The MAGIC collaboration recognized that this effort is not negligible especially concerning man power. Therefore the observation data will be stored by the spanish Grid center PIC.

o Data availability improvements
By importing the observation data to the Grid, the MAGIC collaboration expect that the availablitly of data will be increased with the help of Grid data management methods like data replication, etc. As the main data services will be provided in the future by the national Grid centers instead of research university groups at universities, the overall data availablitly is expected to increase.
o Cost reduction
By using the EGEE infrastructure and the EGEE services the effort for MAGIC collaboration to build a distributed computing system for the Monte Carlo simulations was significantly reduced.

Experiences with the EGEE infrastructure

The experiences of the developers during the different phases of the realisation of the MAGIC Monte Carlo production system on the EGEE Grid infrastructure are described in this chapter. As the MAGIC virtual organisation was accepted as one of the first generic EGEE application, the development process was influenced by general developments within the EGEE project too like changed in the middleware versions, etc.

o Prototype implementation

The migration of the compute intensive MMCS program from a local batch system to the Grid was done by the definition of a template JDL form. This template sends all needed input data together with the executable to the Grid. The resources are chosen by the resource broker. The automatic registration of the output file as a logical file on the Grid was not very reliable at the beginning, but improved to production level within the EGEE project duration.

o Production MAGIC Grid system

The submission of many production system needed the implementation of a graphical user interface and a database for metadata. The graphical user interface was realised with the JAVA programming language. The execution of the LCG/gLite commands is wrapped in JAVA shell commands. A MySQL database holds the schema for the metadata. As mentioned above the "copy and register" process for the output file was not realiable enough an additional job status "DONE (data available)" was invented. With the help of the database, jobs that did not reach this job status within two days are resubmitted. The job data are keeped in a seperate database table to analyse them later.

o Reliability of EGEE services

The general services like resource brokers, VO management tools and Grid user support was provided by the EGEE resources providers. The MAGIC Grid is setup on top of this services. A short report of the experiences with this production services will be given.

Key issues for the future of Grid technology

The MAGIC collaboration is currently evaluating the EGEE Grid infrastructure as the backbone for a distributed computing system in the future including the data storage on Grid data centers like PIC. Furthermore the discussion with other projects like the HESS collaboration has started to move towards "Virtual Very High energetic Gamma ray observatory" [7]. The problems and challenges that needs to be solved on the track to a sustainable Grid infrastructure will be discussed from the user perspective

References:

[1] T. Weekes et al., The Astrophysical Journal, volume 342 (1989), p. 379
[2] F. A. Aharonian et al., Nature 432, 75 - 77 (04 November 2004)
[3] E. Lorenz, 1995, Fruehsjahrtagung Deutsche Physikalische Gesellschaft, March 9-10
[4] T. Weekes et al., Astropart. Phys., 17, 221-243 (2002)
[5] Enomoto, R. et al., Astropart. Phys. 16, 235-244 (2002)
[6] H. Kornmayer et al., "A distributed, Grid-based analysis system for the MAGIC telescope",
Proceedings of the CHEP Conference , Interlaken, Switzerland, 2004
[7] H. Kornmayer et al., "Towards a virtual observatory for high energetic gamma rays", Cherenkov 2005,
Paris, 2005

2d: VO tools - Portals / 42

Experience Supporting the Integration of LHC Experiments Software Framework with the LCG Middleware

Authors: Andrea Sciaba[1]; Antonio Delgado Peris[1]; Flavia Donno[2]; Massimo Lamanna[1]; Patricia Mendez Lorenzo[1]; Simone Campana[1]; roberto santinelli[1]

[1] *CERN/IT/PSS*

[2] *CERN/IT/GD*

Corresponding Author: roberto.santinelli@cern.ch

The LHC experiments are currently preparing for data acquisition in 2007 and because of the large amount of required computing and storage resources, they decided to embrace the grid paradigm. The LHC Computing Project (LCG) provides and operates a computing infrastructure suitable for data handling, Monte Carlo production and analysis.
While LCG offers a set of high level services, intended to be generic enough to accommodate the needs of different Virtual Organizations, the LHC experiments software framework and applications are very specific and focused on the computing and data models.
The LCG Experiment Integration Support team works in close contact with the experiments, the middleware developers and the LCG certification and operations teams to integrate the underlying grid middleware with the experiment specific components. The strategical position between the experiments and the middleware suppliers allows EIS team to play a key role at communications level between the customers and the service providers.
This activity is the source of many improvements on the middleware side, especially by channelling the experience and the requirements of the LHC experiments.

The scope of the EIS activity encompasses several areas:

1) Understanding of the experiment needs
2) Identify open issues and possible solutions
3) Develop specific interfaces, services and components (when missing in or not yet satisfactory)
4) Provide operational support during Data Challenges, Service Challenges and massive productions.
5) Provide and maintain the user documentation;
6) Provide tutorial for the users community

In the last year, the focus has been extended also to non High-Energy Physics communities like Biomed, GEANT4 and UNOSAT. In this work we discuss the EIS experience, describing the issues raising in the organization of the Virtual Organization support and the achievements, together with the lessons learned. This activity will continue in the framework of EGEE II, and we believe could be an example for several users communities on how to optimise their uptake of grid technology in the most efficient way.

Poster and Demo session + cocktail / 43

MEDIGRID: Mediterranean Grid of Multi-risk data and Models

Author: Ladislav Hluchy[1]

Co-author: Branislav Simo [1]

[1] *Institute of Informatics, Slovakia*

We present an IST project of the 6th Framework Programme, aimed to create a distributed framework for multi-risk assessment of natural disasters that will integrate various models for simulation of forest fire behavior and effects, flood modeling and forecasting, landslides and soil erosion simulations. Also, a distributed repository with earth observation data, combined with field measurements is being created, which provides data to all models using data format conversions when necessary. The entire system of models and data will be shaped further as a multi-risk assessment and decision support information platform.

There are 6 partners in the project from Greece, Portugal, France, Spain, United Kingdom and Slovakia.

The system targets both Linux and Windows based simulation models. The Linux based models are meteorological, hydrological and hydraulics models of the flood forecasting application, with meteorology and hydraulics being a parallel MPI tasks. Other applications - forest fire behaviour and effects, landslides and soil erosion - are sequential Windows jobs. These simulations are being merged into one system that uses common distributed data warehouse containing data for pilot areas in France, Portugal and Spain. User should be able to transparently run these simulations from the application portal, reuse data between models and store the results annotated with metadata back to the data warehouse.

In order to create a virtual organization (VO) for multi-risk assessment of natural disasters a grid middleware had to be chosen to be used on computing resources. Because each of the partners provides some of the services on his own resources that run both Linux and Windows, we could not use available middleware toolkits like LCG or Globus as they are focused on Unix/Linux platform. For example, they build their data services on the GridFTP standard for data transfer. However, there are stable implementations of GridFTP just for Unix based systems, ignoring the world of Windows. Therefore, we have decided to implement our own data transfer and job submission services. In order to keep some compatibility with the established grid infrastructures, we have chosen the Java implementation of the WSRF specification by the Globus alliance as a base for our services. It is an implementation of core web (grid) services with security, notifications and other features and it is capable of running on both Windows and Linux. Each of the system components - simulation models, data providers, information services or other supporting services - is exposed as a web service. We use WSRF as a standard basic technology that both serves as an implementation framework for individual services and also enables to glue the individual components together.

The whole system will be accessible via a web portal. We have chosen GridSphere portal framework for its support of portlet specification. Application specific portlets will allow users to invoke all the simulation services plugged into the system in application specific manner; for example using maps for selection of a target area or an ignition points for forest fire simulations. There will be portlets for browsing results, metadata describing those results, testbed monitoring and others.

So far, two services have been implemented on top of the WSRF: Data Transfer service and Job Submission service.

Data Transfer service serves as a replacement for widely used GridFTP tools. The main disadvantage of GridFTP is that implementations are available just for the UNIX platforms. In Medigrid, Windows is a platform of several models and porting them to UNIX world was not an option for developers.

Data Transfer service provides data access policies definition and enforcement in terms of access control lists (ACLs) defined for each data resource - a named directory serving as a root directory for given directory tree accessible via the service. It has been integrated with central catalog services we have deployed: Replica Location Service - a service from Globus toolkit for which we had to implement WSRF wrapper - and Metadata Catalog Service - a service from Gryphyn project that is just a plain web service.

Job Submission service provides the ability to run the executable associated to it with parameters provided with job submission request. Currently, jobs are started locally using the “fork” mechanism on both Linux and Windows. Requests are queued by the service and run in the “first come first served” manner in order not to overload the computer. In near future we plan to add job submission forwarding from the service to a Linux cluster and later on to a classical grid.A base of the project’s portal has been set up based on the Gridsphere portal framework. Thus far portlets have been developed for browsing the contents of the metadata catalog service and a portlet for generic job submission.

As it can be seen in this project, the world of simulations is not limited to the Unix platform and support for Windows applications is desired but missing.Therefore we think it may be important for the EGEE project to try to suppport Windows users in order to widen its reach and appeal.

Summary:

We present an IST project of the 6th Framework Programme, aimed to create a distributed framework for multi-risk assessment of natural disasters that will integrate various models for simulation of forest fire behavior and effects, flood modeling and forecasting, landslides and soil erosion simulations. Also, a distributed repository with earth observation data, combined with field measurements is being created, which provides data to all models using data format conversions when necessary. The entire system of models and data will be shaped further as a multi-risk assessment and decision support information platform.

2a: Workload management and Workflows / 44

K-Wf Grid: Knowledge-based Workflows in Grid

Author: Ladislav Hluchy[1]

Co-author: Ondrej Habala [1]

[1] *Institute of Informatics, Slovakia*

We present an IST project of the 6th Framework Programme, aimed towards intelligent grid middleware and workflow construction. The project’s acronym K-Wf Grid stands for “Knowledge-based Workflow System for Grid Applications”. The project itself employs ontologies, artificial reasoning, Petri nets and modern service-oriented architectures in order to simplify the use of grid infrastructures, as well as integration of applications into the grid. K-Wf Grid system is composed of a set of modules. The most visible one is the collaboration portal, from which a user can control the infrastructure and manage his/her application workflows. Behind this portal are hidden services doing the workflow management, monitoring of applications and infrastructure, knowledge extraction, management, and reuse. The project is behind its prototype phase and a successful review by the Commission. The idea of the project is based in the observation, that users often have to learn not only how to use the grid, but also how to best take advantage of its components, how to avoid problems caused by faulty middleware, application modules and the inherent dynamic behavior of the grid infrastructure as a whole. Additionally, with the coming era of resources virtualized as web and grid services, dynamic virtual organizations and widespread resource sharing, the variables that are to be taken into account are increasing in number. Therefore we tried to devise a user layer above the infrastructure, that would be able to handle as much of the learning and remembering as possible. This layer should be able to observe what happens during application execution, infer new knowledge from these

observations and use this knowledge the next time an application is executed. This way the system would - over time - optimize its behavior and use of available resources.
The realization of this idea has been split into several tasks and formed into the architecture, that became the K-Wf Grid project.
The main interaction of users with the system occurs through the Web Portal. Through it, users can access the grid, its data and services, obtain information stored in the knowledge management system, add new facts to it, construct and execute workflows. The portal consists of three main parts, the Grid Workflow User Interface (GWUI), the User Assistant Agent (UAA) interface, and the portal framework based on GridSphere, including collaboration tools from the Sakai project and interfaces to other K-Wf Grid modules. GWUI is a Java applet visualization of a Petri net-modeled workflow of services, in which the user can construct a workflow, execute it and monitor it. UAA is an advisor, which communicates to the user all important facts about his/her current context –the services he/she considers to use, the data he/she has or needs. Apart from automatically generated data, the displayed information contains also hints entered by other users, which may help anyone to select better data or services or avoid problems of certain workflow configurations. This way the users may collaborate together and share knowledge.
Under the Web Portal lies the Workflow Orchestration and Execution module, composed of several components. These components together are able to read a definition of an abstract workflow, expand this definition into a regular workflow of calls to service interfaces, map these calls to real service instances and execute this workflow to obtain the expected results, described in the original abstract workflow. This way the user does not need to know all the services that are present in the grid and he/she is required only to state what result is required.
To be able to abstract the grid in such a way as described in previous paragraph, the system has to know the semantics of the grid environment it operates on, and so we need to employ serious knowledge management, computer-based learning and reasoning. This is the area of the Knowledge module, which is split into the storage part –Grid Organization Memory (GOM), and the learning part –Knowledge Assimilation Agent (KAA). KAA takes observed events from the monitoring system, maps them to the context of the performed operation and extract new facts from them. These facts are then stored into GOM, as well as used in later workflow composition tasks in order to predict service performance. GOM itself stores all information about the available application services in a layered ontology and new applications may be easily added into its structure by describing their respective domains in an ontology, connected to the general ontology layer developed in K-Wf Grid.
The monitoring infrastructure is integrated into the original grid middleware, with the Grid Performance Monitoring and Instrumentation Service (GPMIS) as a processing core. GPMIS receives information from a network of sensors, embedded into the middleware, application services (where it is possible to instrument the services) and into the other K-Wf Grid modules. Apart from collecting observations for the learning modules, the monitoring infrastructure is also a comprehensive tool for performance monitoring and tuning, with comfortable visual tools in the user portal.
At the bottom of the architecture lies the grid itself –the application services, data storage nodes and communication lines. K-Wf Grid has three distinct and varied pilot applications, which it uses to test the developed modules. One of them is a flood prediction suite, developed from a previous effort in the CROSSGRID project. It consists of a set of several simulation models for meteorology, hydrology and hydraulics, as well as support and visualization tools, all instantiated as WSRF services. The second application is from the business area –a web service-based ERP system. The third application is a system for coordinated traffic management in the city of Genoa.

Summary:

We present an IST project of the 6th Framework Programme, aimed towards intelligent grid middleware and workflow construction. The project's acronym K-Wf Grid stands for "Knowledge-based Workflow System for Grid Applications". The project itself employs ontologies, artificial reasoning, Petri nets and modern service-oriented architectures in order to simplify the use of grid infrastructures, as well as

integration of applications into the grid. K-Wf Grid system is composed of a set of modules. The most visible one is the collaboration portal, from which a user can control the infrastructure and manage his/her application workflows. Behind this portal are hidden services doing the workflow management, monitoring of applications and infrastructure, knowledge extraction, management, and reuse. The project is behind its prototype phase and a successful review by the Commission.

1c: Earth Observation - Archaeology - Digital Library / 45

ArchaeoGRID, a GRID for Archaeology

Authors: Giuliano Pelfer[1]; Juan Antonio Barcelò Alvarez[2]; Pier Giovanni Pelfer[3]

Co-authors: Alfredo Maximiano Castillejo [2]; Andrea Toselli [2]; Antonio Politi [4]; Assumció Vila i Mitjà [5]; Igor Bogdanovic [2]; Jordi Estevez Escalera [6]; Jordi Pijoan-López [2]; Jose Antonio Esquivel [7]; Laura Mameli [2]; Oriol Vicente Campos [2]; Raquel Piqué Huerta [2]

[1] *CSDC, University of Florence, Italy*

[2] *Department de Prehistòria, UAB, Barcelona, Spain*

[3] *Dept. Physics, University of Florence and INFN, Italy*

[4] *ISC-CNR, Florence, Italy*

[5] *Institució Milà i Fontanals , CSIC, Barcelona, Spain*

[6] *Department de Prehistòria, UAB, Barcelona*

[7] *IAG, Universidad de Granada, Spain*

Corresponding Author: pelfer@tin.it

Modern archaeology, between the historical, anthropological and social sciences, is the more suitable and mature for the application of the Grid technologies. In fact, archaeology is a multidisciplinary historical science, using data and methods from many of the natural and social sciences. Archaeological research do and has done large use of computers and digital technologies for data acquisition and storage, for quantitative and qualitative data analysis, for data visualisation, for mathematical modeling and simulation. The Web also is intensively used for results exchange, for communication and for accessing to large database by the Web Services technology. The interest of archaeologist for such methods is today more than a temporal interest. There are many computational archaeologists through the world and specialised quantitative archaeology laboratories experimenting new methods in spatial analysis, geostatistics, geocomputation, artificial intelligence applications to archaeology, etc.
In fact any material remains, artifacts and ecofacts, macro and microscopic, present on the earth surface, representing the material culture of the past societies is relevant for the archaeology, independently from its esthetical or economical value. Remains should be described according to their basic properties (shape, size, texture, composition, spatial and temporal location), which implies the use of sophisticated procedures for its computer representation: 3D geometry and realistic rendering, among them.
Furthermore, data should be related spatially and temporally in complex ways. In so doing, an archaeological site should be understood as a complex sequence of finite states of a spatio-temporal trajectory, where an original entity (ground surface) is modified successively, by accumulating things on it, by deforming a previous accumulation or by direct physical modification (building, excavation). This spatio-temporal representation must be considered as continuum made up of discrete, irregular, discontinuous geometrical shapes (surfaces, volumes) defined by additional characteristics (shape, texture, composition, as dependent variables of the model) which in turn influence the variation of every archaeological feature.

The idea is that interfacial boundaries represent successive phases, and are dynamically constructed. Within them, there should be some statistical relationship between the difference in value of the dependent regionalised variable which defines the discontinuity at any pair of points and their distance apart.
The complexities of archaeological data processing are more demanding when we consider that archaeological analysis cannot be constrained to the study of a single site. In recent years archaeological research teams are very much interested in doing extended projects involving the study of many different sites at very large geographic regions during very long time spans. This work is specially relevant in the case of the study of paleoclimatic human adaptations, hunter-gatherer societies mobility and the study of the origins of cities and early state formation. In these cases, archaeological data produced by excavation and field survey or retrieved from different types of available archives, are not only huge in quantity but also in diversity and complexity, and the computing power needed for their analysis, simulation and visualisation is very large. The purpose is then working towards a landscape archaeology which should reconstruct the evolution of settlement organization on the studied region with a low or high spatio-temporal resolution in relation with the analysed level, intersite, intrasite or regional.
For such a precise reconstruction of geomorphology, hydrology, climate, landcover and landuse of the region, based on known data, must be done using models and simulation. Moreover, as a social and historical science, such a simulation cannot stops at the physical elements, but it should include the study of demographic variation, including demographic models, settlement and urban dynamics and production and exchange models.

All that means that archaeology is a computer intensive discipline. Model building is time consuming and resource intensive, and archaeological data are huge. They also are unique in character, so they cannot be substituted, because they need care to preserve. Everything in our analysis has to be preserved and stored, but also the information about them. The results of simulated data must be preserved for a long time because they represent the status of the data interpretation at some date and will be useful for future analysis.("Crisis of Curation"). For the previous reasons the archaeology need to exploit the GRID technology for data access, storage and management, for data analysis, for simulation, for archaeological knowledge circulation : from WEB to GRID. ArchaeoGRID will offer the unique opportunity to share data, processing and model building opportunities with other branches of science and create synergy with other GRID projects.(Earth Sciences, Digital Library, Astrophysics GRID projects, etc.)
The starting project proposes to begin with the study of the origin of the city in Mediterranean area between XI and VIII Centuries B.C. using the GILDA t-Infrastructure. The study will provide a functional framework for broad studies of the interactions of humans in ancient urban societies and with the environment .
During the past fifteen years, archaeologists in the Mediterranean have accumulated large amounts of computerized data that have remained trapped in localized and often proprietary databases. It is now possible to change that situation. ArchaeoGRID will be made to facilitate ways in which such data might be brought together and shared between researchers, students, and the general public. Archaeological data always includes an intrinsic geographic component, and the compilation and sharing of geographic data through GIS has become increasingly important in the governmental, private sector and academic worlds during the past years. New GRID technologies for spatial data, expansion of the Web Services and development of open GIS technology now make it possible to share geographic information quickly, widely and effectively.
The first application running on the GILDA be will be related with paleoclimate and weather simulation in the regions where the urban centers originate around the IX and VIII centuries B.C. In fact weather phenomena, climate and climate changes produced effects on individuals and societies in the past. In the next future, GILDA will be used to explore the possibilities of different computational methodologies insiting of the tools for the analysis of spatio-temporal data. Classical statistical analysis of spatio-temporal series will be used, but also we intend to develop new methods for the analysis of longitudinal analysis, based on neural networks technology.
Simulation programs and data available on the web and free will be used for application. Such data could be integrated with data from archaeological excavation

and survey. The complexity and the dimension of program code and data require the use of MPI library for parallel calculation on GILDA computers using Linux OS.
Open source GRASS GIS and package R for statistical analysis installed on GILDA will give the possibility to prepare the input data for the full Mediterranean area and for the territories of the urban centers.
A schematic architecture of the ArchaeoGRID showing the relevant parts and their links will be presented. Given the intrinsic nature of archaeological field work, the communication and the information exchange between groups on site and groups working in distant laboratories, museums and universities need fast and efficient communication ways. Telearchaeology lies at the real nature of archaeological endeavor and could be very useful also for education and for diffusion of the archaeological knowledge. A multicast architecture for advanced videoconferencing specially tailored for large scale persistent collaboration could be used.
The added value, linked with new perspectives of the archaeological and historical research, with the management of the archaeological heritage, with the media production, with the territory management and with tourism, will be discussed.

1b: Astrophysics/Astroparticle physics - Fusion - High-Energy physics / 46

FUSION ACTIVITIES IN THE GRID

Author: Francisco Castejon[1]

[1] *CIEMAT*

Corresponding Author: francisco.castejon@ciemat.es

The future Magnetic confinement Fusion energy research will be mainly based upon large international
facilities with the participation of a lot of scientist belonging to different institutes. For instance, the large
device ITER (International Tokamak Experimental Reactor) that will be built in Cadarache (France) is
participated by six partners: Europe, Japan, USA, Russia, China, and Korea. India is presently involved in
negotiations to join the project and Brazil is also considering the possibility of joining the project. Besides
ITER, the Fusion community has a strong collaboration structure devoted both to the tokamak and the
stellarator research. As a result of this structure, there exists a network of groups and Institutes that are
sharing facilities and/or results obtained on those facilities.
Magnetic Fusion facilities are constituted by large devices devoted to study Plasma Physics that produce a
large amount of data to be analysed (the typical rhythm of data production is about 1 GBy/s for a conventional
device that can reach 10 times larger value in ITER). The analysis and availability of those data is a key point
for the scientific exploitation of those devices.
Also, large computations are needed for understanding plasma Physics and developing new calculation
methods that are very CPU time consuming. A part of this computation effort can be performed in a
distributed way and Grid technologies are very suitable to perform those calculations. Several Plasma Physics
applications are being envisaged for adapting into the grid, those that can be distributed in different processors.
The first kind of applications is In particular, Monte Carlo codes are suitable and powerful tools to perform

transport calculations , especially in those cases like the TJ-II stellarator that present radially extended ion
orbits, which has strong influence on confinement: The fact that orbits are wide makes that ions perform large
radial excursions during a collision time, which will enhance outward heat flux. The usual transport calculations based on local plasma characteristics that give local transport coefficients are not suitable for this
kind of geometry in the long mean free path regime. The suitable way to estimate transport is to follow
millions of individual particles that move in a background plasma and magnetic configuration. The interaction
with other particles is simulated by a collision operator, which depends on density and temperature, and by a
steady state electric field, caused by the unbalanced electron and ion fluxes. This tool will be also useful to
take into account other kinetic effects on electron transport, like those related to heating and current drive.
This transport tool is now working in a Supercomputer and is being prepared to be ported to the grid, where
will run soon. The capability of performing massive kinetic transport calculations will allow us to explore
transport properties in different heating conditions and collisionalities, as well as with different electric field
profiles.
Another application that requires distributed calculations is the massive ray tracing. The properties of
microwave propagation and absorption are estimated in the geometrical optics (or WKB) approximation by
simulating the microwave beam by a bunch of rays. Those rays are launched and followed inside the plasma
and all the necessary quantities are estimated along ray trajectories. Since all the rays are independent, they
can be calculated separately . The number of rays needed in a normal case is typically 100 or 200, and the
time needed for every ray estimate is about 10-20 minutes. This approximation works when the waist of the
beam is far from any critical layer in the plasma. Critical layers are those where mode conversion, absorption,
or reflection of microwaves happens. When the waist of the beam is closed to critical layers, a much higher
number of rays is needed to simulate the beam. The typical number can be of the order of 10000, which is
high enough to make it necessary to run the application in the grid. Massive ray tracing calculations could
also be useful to determine the optimum microwave launching position in a complex 3D device like a real
stellarator.
These two former applications require that a common file with stellarator geometry data is distributed in all
the processors as well as individual files with the initial data of every ray and trajectory.

Stellarator devices present different magnetic configurations with different confinement properties. It is
necessary to look for the magnetic configuration that present the best confinement properties, considering
the experimental knowledge of confinement and transport in stellarators. Therefore, stellarator optimization
is a very important topic to design the future stellarators that have to play a role in Magnetic confinement
fusion. The optimization procedure has to take into account a lot of criteria that are based on the previous

stellarator experience: neoclassical transport properties, viscosity, stability, etc. A possible way to develop this
procedure is to parametrize the plasma by the Fourier coefficients that describe the magnetic field. Every set
of coefficients is considered as a different stellarator with different properties. The optimization procedure
has to take into account the desired characteristics for a magnetic configuration to be suitable for an optimised stellarator. The optimization criteria are set through functions that take into account the properties
that favour plasma confinement . Every case can be run in a separate node of the grid in order to explore the
hundreds of parameters that are involved in the optimization.
Presently, other applications are being considered to be run in the grid in order to solve efficiently some
problems on Plasma Physics that are needed for the future magnetic confinement devices. For instance,
transport analysis is a key instrument in Plasma Physics that gives the transport coefficients that fit the
experimental data. Transport analysis is performed using transport codes on the real plasma discharges. A
plasma confinement device can perform tens of thousands of discharges along its life and only a few of them
are analysed. It would be possible to install a transport code in the grid that performs automatic transport
analysis on the experimental shots. In this way, the dependence of local transport coefficients on plasma
parameters like magnetic configuration, density, temperature, electric field, etc. can be extracted. And, finally
the tokamak equilibrium code EDGE2D can be installed in the grid to obtain equilibrium parameters in the
edge, which is basic to estimate the exact plasma position and the equilibrium properties in the plasma edge.

1b: Astrophysics/Astroparticle physics - Fusion - High-Energy physics / 47

Genetic Stellarator Optimisation in Grid

Author: Vladimir Voznesensky[1]

[1] *Nuclear Fusion Inst., RRC "Kurchatov Inst."*

Corresponding Author: vovic@nfi.kiae.ru

Computational optimisations can be found in a wide area of natural, engineering and economical sciences. They may be carried out by different methods, that include classical gradient-based, genetic algorithms, etc.

Stellarator facilities optimisation may be noted as an example of such task. Stellarators are the toroidal devices for magnetic confinement of plasma. In contrast to tokamak (ITER facility, for example), no toroidal current is required here, so that stellarators are principally stationary devices. As a payment for stationary working, stellarators are principally three-dimensional (non axisymmetric) configurations. This can lead to enhanced losses of fast particles - to an enhancement of losses of fast particles - the product of fusion reaction- and plasma.

The plasma equilibrium in stellarator can be found if the shape of the boundary plasma surface and the radial profiles of plasma pressure and toroidal current are prescribed. During the last decades it was shown that the properties of the stellarators can be significantly improved by appropriate choice of the shape of the boundary magnetic surface. Because of the large variety of stellarators the optimisation is still under way.

Boundary surface may be characterised by a set of Fourier harmonics that give the shape of the surface, the magnetic field, and the electric current. The Fourier coefficients compose a multidimensional space of optimisation (free) parameters and their number may exceed a hundred.

The quality parameters are functions depending on optimisation parameters and describing the properties of the considered configuration. As soon as the stellarator plasma equilibrium is found, quality parameters such as stability of different modes, fast particle long time collision-less confinement, neoclassical transport coefficients, bootstrap current, etc. can be computed.

In the optimisation task, the measure of optimum, so called a target function, is based on quality parameters and may be, for example, a weighted sum of such parameters. Computation of a stellarator quality parameters set and target function values for a given optimisation parameters vector takes about 20 minutes on conventional PC.

Such computation may form a single grid job. The technique presented in this work may be useful for tasks having target function calculation large enough for a job.

Splitting each gradient-based optimisation step into several independent grid jobs may be ineffective in case of numerical gradient computation due to hardly asynchronous jobs completion.

For such reason, genetic algorithms have been chosen as optimisation methods. Such method treats parameter vector of a variant as a "genome" and imply three activities in each iteration. The activities are selection of "parents", their breeding and computation of target function values for each "child" genome.

Initial pool of genomes can be generated randomly inside the optimisation parameters variation domain defined by a user. Genetic method iterations enrich genome pool with new better genomes.

Genetic algorithms behave well for grid computations, because genome pool may be appended by grid jobs results sporadically, so aborting or delaying several jobs completion would not affect the overall optimisation process hardly.

During the selection, genome with better target function value should have a preference among genomes pool. The following algorithm has been used for choosing "mothers" and "fathers" of a new stellarator generation.

Genomes pool is arranged according to target function values, so the better genomes go first. Then, iterations over the pool are carried out until a "father" is chosen. On every iteration, a uniform random number is generated, so current genome is chosen with some user-predefined probability, say 2% or 3%. A "mother" is chosen in the same manner.

Such selection algorithm have no direct influence from target function derivatives, so it suppresses fast appearing of "super genome" (i.e. "inbreeding") that may constrain other potentially fruitful genomes.

Genetic algorithm breeding in case of continual optimisation domain should not change statistical mean and dispersion of genome pool, because there is no reason to shift, disperse or collect optimisation space points in the breeding activity. Only selection activity should put such changes. The following method preserving such statistical parameters have been used for stellarators.

Two coefficients f and m for each Fourier harmonic from every parent vectors pair were bred separately. Every new coefficient was a random number of Gaussian distribution. The distribution had the mean (f+m)/2 and the standard deviation |f-m|/2.

A set of scripts realising the technique in Python language have been developed. One of them generates an initial genome pool, another one spawns new jobs for quality parameters computation, the third gathers already computed results from the grid and the fourth generates new part of genome pool depending on the existing one. The number of concurrently spawned jobs is kept below a given threshold. New, running and complete jobs' genomes and quality parameters are stored in files of special directory hierarchy.

The iteration is realised by a Bash script. The script implies spawning, gathering, genetic generation scripts and scheduling a new iteration using "at" command. The scripts are intended to run controlled by user commands on LCG-2 user interface host.

A test example of stellarator optimisation task have been computed. About 7.500 variant jobs have been spawn, about 1.500 of them were discarded since no equilibria were found. In other 6.000, a set of quality parameters based on the fields and target function values were computed.

Histograms representing distribution of target function values in first, second, third, fourth, fifth and sixth thousands of results in order of appearance show that the sets of best values converge to the believed optimum value with the linear order.

This technique can be employed fruitfully in developing new stellarator concepts with different optimization criteria. Moreover, the proposed technique based on genetic algorithms and grid computing that works for the stellarator optimisation task can be employed in a wide spectrum of applications, both scientific and practical.

REFERENCES
1. ESA Genetic Algorithms Tutorial by Robin Biesbroek, http://www.estec.esa.nl/outreach/gatutor/Default.htm
2. M.I.Mikhailov, V.D.Shafranov, A.A.Subbotin, et.al. Improved alpha-particle confinement in stellarators with poloidally closed contours of the magnetic field strength. // Nuclear Fusion 42 (2002) L23-L26

Poster and Demo session + cocktail / 48

Sustainable management of groundwater exploitation using Monte Carlo simulation of seawater intrusion in the Korba aquifer (Tunisia)

Author: Jawher Kerrou[1]

Co-authors: Giuditta Lecca [2]; Giuseppe La Rocca [3]; Philippe Renard [1]; Roberto Barbera [3]

[1] *University of Neuchatel*

[2] *CRS4*

[3] *University of Catania*

Corresponding Author: philippe.renard@unine.ch

Worldwide, seawater intrusion and salinisation of coastal aquifers and soils is a major threat for food production. While the physico-chemical processes triggering the transport and accumulation of salts in these regions are relatively well known and well described by a set of partial differential equations, often it is extremely difficult to model accurately these phenomena because of the lack of an accurate data set. On one hand the physical parameters (porosity, permeability, dispersivity) that control groundwater flow are extremely variable in space within geological media and are only measured at some specific locations, on the other hand the forcing terms (pumping, precipitation, etc.) are often not measured directly in the field. The result is a high level of uncertainty. The problem is how to take rational decision toward sustainable water management in such a context ?

One possibility explored within this work is to run a large set of model simulations with stochastic parameters by means of the EGEE GRID infrastructure and to define robust and sustainable water management decisions based on probabilistic analysis of the resulting simulation outputs. This approach is currently being investigated in the Cape Bon peninsula, located 50 km South-East of Tunis, one of the most productive agricultural areas in Tunisia. In this plain the World Bank has shown that major water resources problem could occur in the next decade. One of the major sources of uncertainty in the Cap Bon aquifer system are the pumping rates and their time evolution. To investigate the impact of this source of uncertainty, first a geostatistical model of the spatial distribution of the pumping has been constructed and then the GRID has been used to run a 3D density-dependent groundwater flow and salt transport model in a Monte Carlo framework.

While these results are still preliminary, GRID computing paradigm offers clearly a huge potential within this field. One particularly interesting aspect offered by this methodology to Tunisian water managers, not having access to local computing technology, is to be able in a near future to run directly, via a web portal to the GRID, their groundwater flow simulation and uncertainty analysis. This option has not been tested yet and requires further development.

1b: Astrophysics/Astroparticle physics - Fusion - High-Energy physics / 49

Experiences on Grid production for Geant4

Author: Alberto Ribon[1]

[1] *CERN*

Corresponding Author: alberto.ribon@cern.ch

Geant4 is a general purpose toolkit for simulating the tracking and interaction of particles through matter. It is currently used in production in several particle physics experiments (BaBar, HARP, ATLAS, CMS, LHCb), and it has also applications in other areas, as space science, medical applications, and radiation studies.
The complexity of the Geant4 code requires careful testing of all of its components, especially before major releases (which happens twice a year, in June and December).
In this talk, I will describe the recent development of an automatic suite for testing hadronic physics in high energy calorimetry applications. The idea is to use a simplified set of hadronic calorimeters, with different beam particle types, and various beam energies, and comparing relevant observables between a given reference version of Geant4 and the new candidate one. Only those distributions that are statistically incompatible are then printed out and finally inspected by a person to look for possible bugs.
The suite is made of Python scripts, and utilizes the "Statistical Toolkit" for the statistical tests between pair of distributions, and runs on the Grid to cope with the large amount of CPU needed in a short period of time. In fact, the total CPU time required for each of these Geant4 release validation productions amounts to about 4 CPU-years, which have to be concentrated in a couple of weeks.
Therefore, the Grid environment is the natural candidate to perform this validation production. We have already run three of them, starting in December 2004. In the last production, in December 2005, we run as Geant4 VO, for the first time, demonstrating the full involvement of Geant4 inside the EGEE communities. Several EGEE sites

have provided us with the needed CPU, and this has guaranteed the
success of the production, arriving to an overall efficiency rate
of about 99%.
In the talk, emphasis will be given on our experiences in using
the Grid, the results we got from it and possible future
improvements. Technical aspects of the Grid framework that have
been deployed for the production will only be mentioned; for more
details see the talks of P.Mendez and J.Moscicki.

2b: Data access on the grid / 51

GDSE: A new data source oriented computing element for Grid

Authors: Edgardo Ambrosi[1]; Giuliano Taffoni[2]

Co-authors: Andrea Barisani [2]; Antonia Ghiselli [1]; Fabio Pasian [2]; Vuerli Claudio [2]

[1] *INFN - CNAF*

[2] *INAF - SI*

Corresponding Author: taffoni@oats.inaf.it

1. The technique addressed in connection with concrete use cases
   In a GRID environment the main components that manages the jobs life are the Grid Resource Framework
   Layer, the Grid Information System Framework and the Grid Information Data Model. Since the job life is
   strongly coupled with its computational environment then the Grid middleware must be aware of the specific
   computing resources managing the job. Until now, only two types of computational resources, the hardware
   machines and some batch queueing systems, have been taken into account as a valid Resource Framework
   Layer instances. However different types of virtual computing machines exist such as the Java Virtual Machine,
   the Parallel Virtual Machine and the Data Source Engine (DSE). Moreover the Grid Information System and Data
   Model have been used for representing hardware computing machines, never considering that a software
   computational machine is even a resource that can be well represented. This work addresses the extension of the Grid Resource Framework Layer, of the Information System and of the Data Model so that a
   software virtual machine as a Data Source Engine is a valid instance for a Grid computing model, namely the
   so called Grid-Data Source Engine (G-DSE). Once the G-DSE has been defined, a new Grid element, namely the
   Query Element (QE) can be in turn defined; it enables the access to a Data Source Engine and Data Source,
   totally integrated with the Grid Monitoring and Discovery System and with the Resource Broker.
   The G-DSE has been designed and set up in the framework of the GRID.IT project, a multidisciplinary Italian
   project funded by the Ministry of Education, University and Research; the Italian astrophysical community
   participates to this project by porting on Grid three applications, one of them addressed to the extraction of
   data from astrophysical databases and their reduction by exploiting resources and services shared on the

available INFN Grid infrastructure whose middleware is LCG based. The use case we envisaged and sketched
out for this application reflects the typical way astronomers work with. Astronomers typically require to 1)
discover astronomical data that reside on astronomical databases spread worldwide; this discovery process is
driven through a set of metadata fully describing the data the user looks for; 2) if data are found in some
archive on the network they are retrieved and processed through a suite of appropriate reduction software
tools; data can also be cross-correlated with similar data residing elsewhere or just acquired by the
astronomer; 3) if data the user looks for are not found, the astronomer can decide to acquire them through a
set of astronomical instrumentation or generate them on the fly through proper simulation software tools; 4)
at the end of the data processing phase the user typically saves the results in some database reachable on the
network.
In the framework of our participation to GRID.IT project we realized that the LCG Grid infrastructure based on
Globus 2.4 is strongly computing centric and does not offer any mechanism to access databases in a
transparent way for final users. For this reason, after having evaluated a number of possible solutions like
Spitfire and OGSA-DAI, it was decided to undertake a development phase on the Grid middleware to make it
able to fully satisfy our application demands. It is worth to note here that a use case like that described above
is not peculiar of the astrophysical community only, rather it is applicable to other disciplines where access to
data stored in complex structures like database represent a factor of key importance.
Within the GRID.IT project the extended LCG Grid middleware has been extensively tested proving that the
solution under development makes the Grid technology able to fully meet the requirements of typical
astrophysical application.
The G-DSE is currently in a prototypal state; further work is needed to refine it and bring it in a production
state. Once the Grid middleware has been enhanced through the inclusion of the G-DSE, the new QE can be
set up. The QE is a specialized CE able to interact, making use of G-DSE capabilities, with databases looking
them as embedded resources within the Grid, like a computing resource or a disk resident file. The QE is able
to process and handle complex workflows that foresee both the usage of traditional Grid resources as well as
the new ones; database resources in particular may be seen and used as data repository structures and even
as virtual computing machines to process data stored within them.

2. Best practices and application level tools to exploit the technique on EGEE
A suite of tools are currently in the process of being designed and set up to make easy for applications to use
the functionalities and capabilities of a G-DSE enabled Grid infrastructure. Such tools are mainly thought to
help users in preparing the JDL scripts able to exploit the G-DSE capabilities and, ultimately, the functionalities offered by the new Grid QE. The final goal however is to offer to final users graphical tools to
design and sketch out their workflows to be passed on to the QE for their analysis and processing. A

precondition, obviously, to achieve these results is to have the G-DSE, and then the QE fully integrated in the
Grid middleware used by EGEE.

3. Key improvements needed to better exploit this technique on EGEE
The current prototype of the G-DSE is not included yet in the Grid middleware flavours the EGEE infrastructure
is based on. The test phase carried out on the G-DSE prototype so far has made use of a parallel test bed Grid
infrastructure set up thanks to the collaboration between INFN and INAF. Such parallel infrastructure is made
of a BDII and of a RB on which the modified Grid components constituting the G-DSE have been mounted. The
mandatory precondition to make use of the G-DSE, therefore is its inclusion (i.e. the modified components of
the Grid middleware) in the Grid infrastructure used by EGEE.

4. Industrial relevance
The G-DSE has been originally thought to solve a specific problem of a scientific community and the analysis
of new application fields has been focussed so far in the scientific research area.
Because G-DSE however represents a general solution to make of any database an embedded resource of the
Grid, quite apart from the nature and kind of data contained within it, it is natural for the G-DSE to extend its
applicability even in the field of industrial applications whenever the access to complex data structures is a
crucial aspect.

1b: Astrophysics/Astroparticle physics - Fusion - High-Energy physics / 52

Status of Planck simulations application

Author: Claudio Vuerli[1]

Co-authors: Andrea Zacchei [2]; Fabio Pasian [1]; Giuliano Taffoni [1]

[1] *INAF-SI*

[2] *INAF-OA Trieste*

1. Application context and scientific goals An accurate measure of the whole sky emission in the frequencies of the microwave spectrum and in particular of the Cosmic Microwave Background (CMB) anisotropies can have crucial implications for the whole Astrophysical community as it permits to determine a number of fundamental quantities that characterize our Universe, its origin and evolution. The ESA Planck mission is aimed to map the microwave sky performing at least two complete sky surveys with an unprecedented combination of sky and frequency coverage, accuracy, stability and sensitivity. The satellite will be launched in 2007 carrying a payload composed of a number of microwave and sub-millimetre detectors which are grouped into a high frequency instrument (HFI) and a low frequency instrument (LFI) covering frequency channels ranging from 30 up to 900 GHz. The instruments are built by two international Consortia which are also in charge of the related Data Processing Centres (DPCs). The LFI DPC is located in Trieste, the HFI DPC is distributed between Paris and Cambridge. In both Consortia, participation in the development of the data processing software to be included in the DPCs is geographically distributed throughout the participating Institutions. The overall Planck community is composed of over 400 scientists and engineers working in about 50 institutes spread in 15 countries, mainly in Europe but including also Canada and the United States. A fraction of this community, the one possibly involved with Grid activities, can be defined as the Planck Virtual Organisation (VO).

During the whole of the Planck mission (Design, Development, Operations and Post- operations), it is necessary to deal with aspects related to information management, which pertain to a variety of activities concerning the whole project, ranging from instrument information (technical characteristics, reports, configuration control documents, drawings, public communications, etc.), to the proper organisation of the processing tasks, to the analysis of the impact on science implied by specific technical choices. For this purpose, an Integrated Data and Information System (IDIS) is being developed to allow proper intra-Consortium and inter-Consortia information exchange. Within the Planck community the term "simulation" refers to the production of data resembling the output of the Planck instruments. There are two main purposes in developing simulation activities:

2. during ESA Phase A and instrument Phases A and B, simulations have been used to help finalising the design of the Planck satellite's P/L and Instruments hardware;

3. on a longer time-scale (up to launch), simulated data will be used mainly to help develop the software of the data processing pipeline DPCs, by allowing the testing of algorithms needed to solve the critical reduction problems, and by evaluating the impact of systematic effects on the scientific results of the mission, before real data are obtained.
The output of the simulation activity is Time-Ordered Information (TOI), i.e. a set of time series representing the measurements of the scientific detectors, or the value of specific house-keeping parameters, in one of the Planck instruments. TOI related to scientific measurements are often referred to as Time-Ordered Data (TOD). Common HFI-LFI tools have been built and integrated in order to build a pipeline system aimed at producing simulated data structures. These tools can be decomposed in several stages, including ingestion of astrophysical templates, mission simulator, S/C simulator, telescope simulator, electronics and on-board processing simulator. Other modules, such as the cooling system model, the instruments simulators and the TM packaging simulator, are instrument-dependent. It should be noted that the engine integrating all the tools has to be flexible enough in order to produce the different needed forms or formats of data.
The Planck Consortia participate to this joint simulations effort to the best of their scientific and instrumental knowledge, providing specific modules for the simulations pipeline. For each Consortium the code allowing to produce maps and time-ordered sequences out of simulated microwave skies is the one jointly produced for both Consortia: data simulated by HFI and LFI are therefore coherent and can be properly merged. To the output data of the common code (timelines) an additional LFI-specific code is applied to simulate on-board quantisation and packetisation, in order to produce streams of LFI TM packets.
The goal of this application is the porting of the whole simulation software of the Planck mission on the EGEE Grid infrastructure.

4. The grid added-value
Planck simulations are highly computing demanding and produce a huge amount of data. Such resources cannot be usually afforded by a single research institute, both in terms of computing power and data storage space. Our application therefore represents the typical case where the federation of resources coming from different providers can play a crucial role to tackle the shortage of resources within single institutions. Planck simulations take great advantage from this as a remarkable number of resources are available at institutions collaborating in the Planck VO, so they can be profitably invested to get additional resources shared on the Grid. The first simulation tests have been carried out on the INFN production Grid in the framework of the GRID.IT project. A complete simulation for the Planck/LFI instrument has been run on a single, dual-CPU, workstation and on Grid involving 22 nodes, one for each detector of the LFI instrument. The gain obtained by using the Grid was of ~15 times.
Another added value coming from the Grid is its authentication/authorization mechanism. Planck code as well as data are not public-domain; we need to protect the software copyright; data moreover are property of the Planck P.I. mission. The setup of a Planck VO makes possible to easily monitor and control accesses to both software and data without the need of arranging tools already available in Grid.
Last but not least a federation of users within a VO fosters the scientific

collaboration, an added value of key importance in Planck given that users who collaborates to the mission are spread all over Europe and United States.

5. Experiences and results achieved on EGEE
Due to some initial issues in the start up process of the Planck VO, we were not able to fully exploit the big amount of potential resources available for our application so far. The Planck VO has proved to be quite difficult to manage; the start up process, in particular, has been slowed down by some difficulties in the interactions between the local Planck VO managers and the respective ROCs. To overcome these issues and make the Planck VO fully operative in a short time on-site visits to Planck VO sites are foreseen in order to train local managers in setting up and maintaining the Planck VO node and even local potential users to foster the usage of the Grid technology for the Planck application needs.

6. Key issues for the promotion of the GRID technology
On the basis of our experience with the astrophysical community a special effort is requested to spread the Grid technology and make potential users fully aware of the advantages in using it. User tutorials can be extremely helpful to achieve this goal. Even the preparation of a suite of Grid oriented tools is of key importance like Grid portals and Grid Graphical User Interfaces to make users able to interact with the Grid in an easy and transparent way and to hide some complexities of the underlying technology.

1a: Life Sciences / **53**

Using Grid Computation to Accelerate Structure-based Design Against Influenza A Neuraminidases

Authors: Hurng-Chun Lee[1]; Ying-Ta Wu[2]

Co-authors: Hsin-Yen Chen [1]; Li-Yung Ho [3]

[1] *Academia Sinica Computing Center*

[2] *Academia Sinica Genomic Research Center*

[3] *Academia Sinica Grid Computing Center*

Corresponding Authors: ywu@gate.sinica.edu.tw, hurng-chun.lee@cern.ch

The potential for re-emergence of influenza pandemics has been a great threat since the report of that the avian influenza A virus (H5N1) having acquired the ability to be transmitted to humans. An increase of transmission incidents suggests the risk of human-to-human transmission, and the report of development of drug resistance variants is another potential concern. At present, there are two effective antiviral drugs available, oseltamivir (Tamiflu) and zanamivir (Relenza). Both drugs were discovered through structure-based drug design targeting influenza neuraminidase (NA), a viral enzyme that cleaves terminal sialic acid residue from glycoconjugates. The action of NA is essential for virus proliferation and infectivity; therefore, blocking the actives would generate antivirus effects. To minimize non-productive trial-and-error approaches and to accelerate the discovery of novel potent inhibitors, medicinal chemists can take advantage of using modeled NA variant structures and doing structure-based design.

A key work in structure-based design is to model complexes of candidate compounds to structures of receptor binding sites. The computational tools for the work are based on docking tools, such as AutoDock, to carry out quick conformation search of small compounds in the binding sites, fast calculation of binding energies of possible binding poses, prompt selection for the probable binding modes, and precise ranking and filtering for good binders. Although docking tools can be run automatically, one should control the dynamic conformation of the macromolecular binding site (rigid or flexible) and the spectrum of the screening small organics (building blocks and/or scaffolds; natural and/or synthetic compounds, diversified and/or "drug-like" filtered libraries). This process is characterized by computational and storage load which pose a great challenge to resources that a single institute can afford (For example, using AutoDock to evaluate one compound structure for 10 poses within the target enzyme would take 200 Kilobyte storage and 15 minutes on an average PC). The task to evaluate 1 million compound structures 100 poses each would cost 2 Terabyte and more than hundred years). To support such kind of computing demands, this project was initiated to develop a service prototype for distributing huge amount of computational docking requests by taking the advantages of the LCG/EGEE Grid infrastructure.

According to what we have learned from both the High-Energy Physics experiments and the Biomedical community, an effective use of large scale computing offered by the Grid is very promising but calls for a robust infrastructure and careful preparation. Important points are the distributed job handling, data collection and error tracking: in many cases this might be a limitation due to the need of grid-expert personnel effort. Our final goal is to deliver an effective service to academic researchers who for the most part are not Grid experts, therefore we adopted a light-weight and easy-to-use framework for distributing docking jobs on the Grid. We expect that this decision will benefit future deployment efforts and improve application usability.

Introducing the DIANE framework in building the service is aimed at handling the Grid applications in master-worker model, a native computing model of distributing docking jobs on the Grid. With the skeletal parallelism, applications plugged into the framework inherit the intrinsic DIANE features of distributed job handling such as automatic load balancing, and failure recovery. The python-based implementation also lowers the development effort of controlling application jobs on the Grid. With the hiding of composing JDL and of submitting jobs, users can even easily distribute their application jobs on the Grid without having Grid knowledge. In addition, this system can be used to seamlessly merge local guaranteed resources (like a dedicated cluster) with on-demand power provided by the Grid, allowing researches to concentrate on setting up of their application without facing a heavy entry barrier to move in production mode where more resources are needed.

In a preliminary study, we arranged the work into six tasks: (1) target 3D structure preparation; (2) compound 3D structure preparation and refinement, (3) compound properties and filter, (4) Autodock run (5) probable hits analysis and selection, and (6) complex optimization and affinity re-calculation. The DIANE framework has been applied to distribute about 75000 time-consuming AutoDock processes on LCG for screening possible inhibitor candidates against neuraminidases. In addition to show the distribution efficiency, advantages of adopting DIANE framework in the AutoDock application are also discussed in terms of usability, stability and scalability.

1c: Earth Observation - Archaeology - Digital Library / 55

Data Grid Services for National Digital Archives Program in Taiwan

Author: Wei-Long Ueng[1]

Co-authors: Eric Yen [1]; Hui-Min Lin [1]

[1] *Academia SINICA Grid Computing Centre, Taiwan*

Corresponding Author: wlueng@twgrid.org

Digital archives/libraries are widely recognized as a crucial component of the global information infrastructure for the new century. Research and development projects in many parts of the world are concerned about using advanced information technologies for managing and manipulating digital information, ranging from data storage, preservation, indexing, searching, presentation, and dissemination capabilities to organizing and sharing of information over networks.
Digital Archive demands for reliable storage systems for persistent digital objects, well-organized information structure for effective content management, efficient and accurate information retrieval mechanism and flexible services for varying users needs. Hundreds of Petabyte of digital information has been created and dispersed all over the internet since computers had been used for information processing, and the amount still grows in the rate of tens of Petabyte per year. Grid technology offers a possible solution for aggregating and processing diversified heterogeneous Petabyte scale digital archives. Metadata-based information representation makes specific and relative information retrieval more accurately, makes information resources interoperable, and paves the way for formal knowledge discovery. Taking advantage of advancing IT, semantic level information indexing, categorizing, analyzing, tracking, retrieving and correlating could be implemented.
Data Grid aims to set up a computational and data-intensive grid of resources for data analysis. It requires coordinated resource sharing, collaborative processing and analyzing on huge amounts of data produced and stored by many institutions.
In Taiwan, a National Digital Archive Project (NDAP) was initiated in 2002 with its pilot phase started in 2001. According to the record in 2005, more than 60 Terabytes digital objects was generated and archived by 9 major content holders in Taiwan. Not only delicate and gracious Chinese cultural assets can be preserved and made available via the Internet, but this approach could be proposed as a new paradigm of academic researches based on digital and integrated information resources. The design and implementation phase is ongoing and we would like to illustrate in the EGEE User Forum.
Academia SINICA Grid Computing Centre (ASGC) is in charge of building a new generation of Grid-based research infrastructure in Academia SINICA and in Taiwan based on EGEE and OSG as the Grid middleware. This infrastructure is a major component for the development and the deployment of the National Digital Archive Project (NDAP) providing long-term preservation of the digital contents and unified data access. These services will be built upon the e-Science infrastructure of Taiwan. The Storage Resource Broker (SRB) developed at SDSC, is a Middleware which enables scientists to create, manage and collaborate with flexible, unified "virtual data collections" that may be stored on heterogeneous data resources distributed across a network. The SRB system is the first and the largest (in terms of the data volume) data store in Academia SINICA right now. The system was deployed by ASGC in early 2004, which consists of 7 sites in different institutes, linked by a dedicated fibre campus network, and provided 60 TB capacities in total. In early 2006, it will expand to 120 TB. As of January 2006, more than 30 TB and 1.4 million files have been archived in the distributed mass storage environment. All files are also preserved in two copies on different sites.
In this presentation, idea for utilizing Data Grid infrastructure for NDAP will be depicted and discussed. We will describe the use of SRB in building a collaborative environment for Data Grid Services of NDAP. In the environment, many data intensive applications are developed. We also describe our integration experience in building applications of NDAP. For each application we characterize the essential data virtualization services provided by the SRB for distributed data management.

1c: Earth Observation - Archaeology - Digital Library / 56

Expandig GEOsciences on DEmand

Author: Gael Youinou[1]

[1] *Unknown*

Corresponding Author: gyouinou@cgg.com

Worldwide population faces difficult challenges for the coming years to produce enough energy to sustain global growth and predict main evolutions of the Earth such as earthquakes. Seismic data processing and reservoir simulation are key technologies to help researchers in geosciences to tackle these challenges.

Modern seismic data processing and geophysical simulations require greater amounts of computing power, data storage and sophisticated software. The research community hardly keeps pace with this evolution, resulting in difficulties for small or medium research centres to exploit their innovative algorithms.

Grid Computing is an opportunity to foster sharing of computer resources and give access to large computing power for a limited period of time at an affordable cost, as well as sharing data and sophisticated software.
The capability to solve new complex problems and validate innovative algorithms on real scale problems is also a way to attract and keep the brightest researchers for the benefit of both the academic and industrial R&D geosciences communities.

Under the "umbrella"of the EGEE Infrastructure project was created EGEODE, "Expanding Geosciences On Demand"Open Virtual Organization.

EGEODE is dedicated to research in geosciences for both public and private industrial research & development and academic laboratories.
The Geocluster software, which includes several tools for signal processing, simulation and inversion, enables researchers to process seismic data and to explore the composition of the Earth's layers. In addition to Geocluster, which is used only for R&D, CGG (http://www.cgg.com) develops, markets and supports a broad range of geosciences software systems covering seismic data acquisition and processing, as well as geosciences interpretation and data management.

Many typical Grid Computing projects aim pure Research domains in infrastructure, middleware and usage such as High Energy Physics, Bio informatics, Earth Observation. EGEODE moves the focus towards collaboration between Industry and Academia.

There are two main potential impacts:
1 - The transfer of know-how and services to industry.
2 - The consolidation and extension of EGEODE community, which includes both industrial and academic research centres.

The general benefits of grid computing are:
- Access to computing resources without investing in large IT infrastructure.
- Optimise IT infrastructure
o Load balancing between Processing Centres
o Smoothing peaks of production
o Service continuity; Business Continuity Plan
o Better fault tolerant system and applications
o Leverage Processing Centres capacity
- Lower the total cost of IT by sharing available resources with other members of the community.

And the specific benefits for the Research community:
- Easy access to academic software and comprehensive, industrial software.
- Free the researcher from the additional burden of managing IT hardware and software complexity and limitations.
- Create a framework to share data and project resources with other teams across Europe and worldwide.
- Share best practices, support, and expertises.
- Enable cross-organizational teamwork and partnership.

Some of these benefits have been demonstrated through other Grid Projects and need to be validated in our Geosciences community. Sharing IT resources and Data is typically the primary goal of a Grid Project. Early indicators in our V.O. show that facilitating access to software and simplifying management of hardware and software complexity are also extremely important.

Poster and Demo session + cocktail / 57

Applications integrated on the GILDA's testbed.

Author: Roberto Barbera[1]

Co-authors: Alberto Falzone [2]; Antonio Calanducci [3]; Antonio Carrieri [3]; Emidio Giorgio [3]; Gianluca Passaro [3]; Giuseppe Andronico [3]; Giuseppe La Rocca [3]; Giuseppe Platania [3]; Marco Pappalardo [3]; Rita Ricceri [3]; Rosanna Catania [3]; Salvatore Monforte [3]; Valeria Ardizzone [3]

[1] *INFN - University of Catania*

[2] *NICE Srl*

[3] *INFN Sez. Catania - Italy*

Corresponding Author: giuseppe.larocca@ct.infn.it

Created with the goal of providing an infrastructure for training and dissemination, GILDA revealed itself also as a cute entry point for those communities, often without any experience of distributed computing, desired to test whether or not their applications would receive an added value from the grid. The wide range of applications supported, shows also as a single testbed can serve applications and communities with disparate purposes and final goals. The intensive use of the GENIUS web portal eased the approach to grid for native users, hiding the complexity of middleware, providing also an immediate interface when graphical input/output is required. Hereafter a list of the most significant applications supported in these two years is reported. A list of the most relevant applications that have been integrated on the GILDA's testbed is reported. During the on-line demo session will be presented one or two of these applications focusing on the main EGEE services used.

GA4tS
The acronym GA4tS stands for "Genetic Algorithm for thresholds Searching". It represents a medical application on a grid infrastructure connection, designed in the framework of the INFN MAGIC-5 project, which aims at developing interactive tools to help radiologists with mass detection in mammography image analysis. Given a database of mammography images and extracted from each image a certain number of suspicious regions or regions of interest (ROI), GA4tS is a genetic algorithm able to discriminate among two possible ROI populations (the positive ROI population and the negative ROI population), performing a ROI-based classification. A positive ROI is a pathological ROI, containing a neoplastic lesion or a cluster of micro calcifications. Instead, a negative ROI has no kind of pathology and means healthy tissue. The huge amount of computing power exploitable by the genetic algorithm during its computation represents the grid added value. GA4tS interacts with the LFC's catalog in order to transfer on the worker node the MATLAB Math and Graphics Run-Time Library needed by the genetic algorithm.

Computational Chemistry
The GEMS (gGrid Enabled Molecular Simulator) prototype has been initially implemented on the GILDA test bed infrastructure for the specific case of the study of the properties of gas phase atom-diatom reactions. Recently the prototype has been ported on the production grid. The specific theoretical approach adopted requires massive integrations of trajectories and parallel runs on the largest number of nodes available. Here the advantages of the grid are in the large availability of nodes where the parallel software can run on.

gMOD
gMOD (grid Movie on Demand) is a new application developed to show up how the Grid can give its contribution to make businesses in the world of Entertainment. Plugged into GENIUS, the goal of gMOD is providing a Video-On-Demand service. They are presented a list of movies (movie trailers in our case due to license issues) to choose from and once they have made a choice, the video file is streamed in real time to the video client in the user's workstation. gMOD is built on top of the new EGEE

gLite middleware and makes use of many gLite services (FiReMan and AMGA Catalog, WMS and VOMS). It is worth nothing that gMOD has been realized having in mind the commercial issues and technical problems of a Video On Demand service but can also be used to retrieve any kind of digital multimedia contents from the network with many possible interesting applications such as, for example, e-Learning Systems and Digital Libraries. The grid added value in this case is represented from the large capability of storage, and the absolute safety provided from the use of digital certificates, which gives the faculty to the provider of revoking them in any moment, and setting a predefined and unchangeable time for the provided services.

hadronTherapy
hadronTherapy is a simulation program based on the CERN toolkit GEANT4, developed at INFN LNS. hadronTherapy simulates the beam line and particles revelators used in the proton-therapy facility for the cure of eye cancer at CATANA (Centro AdroTerapia e Applicazioni Nucleari avanzate), active even at INFN-LNS. The typical advantages of porting a Montecarlo code on the grid, the linear factor gained with the simulation splitting, are improved with the recombination of outputs produced by the sub jobs and analyzed. A graphical output is finally obtained exploiting the ROOT's features.

Patsearch
PATSEARCH is a flexible and fast pattern matcher able to search specific combinations of oligonucletide consensi and secondary structure elements. It is able to find, in a given sequence(s), kinds of loop structures that characterize tRNAs, rRNAs and/or any kind of pattern in DNA and protein sequences. Thanks to the grid, PatSearch's application is able to split the search of the given sequence(s) submitting up to ten independent jobs and collects, at the end, the partial results and produce a final output. PatSearch interacts with the LFC's catalog in order to transfer on the worker node's working directory the input file needed by the pattern matcher. PatSearch is one of the candidate applications of the recently approved EU BioInfoGrid Project.

NEMO and ANTARES
The NEMO collaboration has undertaken a R&D program for the construction of an underwater km3 wide telescope for high energy neutrino astronomy in the Mediterranean sea, while ANTARES is constructing a smaller (0.1 km2) underwater neutrino telescope near the Toulon coast. The CORSIKA Montecarlo simulation code is used by NEMO to simulate the interaction of primary cosmic ions with the atmosphere up to the sea level with particular reference to the atmospheric muons generated. In fact, muons represent one of the main sources of background for underwater telescopes for high energy neutrino astronomy. Mass production of muons at the sea level has been simulated first on GILDA and then on the INFN Grid production grid both for the NEMO and ANTARES set-ups. The NEMO collaboration from the grid takes the advantages of the thousands of CPU, which allows to split their simulation in n sub jobs, gaining a factor of n in execution time. Also CORSIKA simulations uses large input files, which could have been handled with much more difficulty without the grid capacity of storage.

Poster and Demo session + cocktail / 58

VOCE - Central European Production Grid Service

Author: Jan Kmunicek[1]

Co-authors: Daniel Kouril [1]; Ludek Matyska [1]

[1] *CESNET*

Corresponding Author: kmunicek@ics.muni.cz

This contribution describes a grid environment of the Virtual Organization for Central Europe (VOCE). VOCE infrastructure currently consists of computational resources and storage capacities provided by Central European resource owners. Unlike majority of other virtual organizations VOCE tends to be generic VO providing application neutral environment especially suitable for Grid newcomers allowing them to get quickly first experience with Grid computing and to test and evaluate Grid environment towards their specific application needs. VOCE facilities currently provide base for Central European t-infrastructure. The main goal of VOCE is to assist in adapting a software for use on a fully production Grid, not within a closed "teaching" environment, even for applications that do not have any Grid / cluster /remote computing experience. The VOCE application neutrality can be seen as an important feature that allows to provide an environment where different application requirements meet and expectations are to be fulfilled. All technical aspects related to the supported middleware (LCG, gLite), computing environments (MPI support), specific user interface support (Charon and P-GRADE portal) will be discussed and preliminary users experiences evaluated.

2c: Special type of jobs (MPI, SDJ, interactive jobs, …) - Information systems / 59

Grid Computing and Online Games

Author: Rafael Garcia Leiva[1]

[1] *Adago Ingenieria*

Corresponding Author: rafael.garcia@andago.com

With the fast growth of the video games and entertainment industry - thanks to the appearance of new games, new technologies and innovative hardware devices - the capacity to react becomes critical for competing in the market of services and entertainment. Therefore it is necessary to be able to count on advanced middleware solutions and technological platforms that allow a fast unfolding of custom made services.

Andago has developed the online games platform Andago Games that provides the technological base necessary for the creation of online Games services around which the main entertainment sites will be able to establish solid business models. The platform Andago Games allows to quickly create online multiplayer games channels with the following services for the final user:

- Pay per Play/ pay per subscription
- Reserving of gaming rooms or servers and advance management of games
- Advanced statistics
- Automatic game launch
- Clans
- Championships, downloads, chat, etc.

However, the platform requires important investments by operators and portals, limiting the number of possible customers. Grid computing will reduce dramatically the amount of these investments by means of sharing resources among different operators and portals. Also, Grid computing offers the possibility to create virtual organizations, where operators and portals could share games and contents, and even their user's base. Technically, the goal is to be able to share expensive resources between providers and to allow billing based on usage. From a business perspective our goal is to open new commercial opportunities in the domain of entertainment.

A common problem with online games is that operators, portals and games providers would like to share resources and aim at sharing the costs to optimize their businesses. Yet business entities are generally required to play all business roles. The European market is still too fragmented and it is hard to reach the critical mass of users needed to make online games businesses profitable and to ensure resource liquidity. Having a Grid infrastructure makes it possible to divide tasks among different actors and in consequence each actor could concentrate on the business it knows best. Application developers provide the applications, portal providers create the portals to attract users, and Telcos/ISP will provide the infrastructure required. Such Virtual Organisations allow for profitable alliances and resource integration. The outcome of a grid enabled online games platform will be to provide the middleware to make this collaboration happen. The Grid ensures not only decreasing costs for businesses, but allows for creating a global European market as applications, infrastructure and users can be shared independently of political and social borders, smoothly integrated and better exploited.

There are also big advantages for users. For example, they will have a larger offer, better quality of service and certainly cheaper services. Grid centralized portals would provide thousands of games and entertainment content from different providers. Today, if one buys a new game and wants to play it online, the user has to connect to a server (possibly) in the USA, unless a local server was set up. Having a Grid infrastructure would largely ease that process. Users will simply connect to the Grid, play and join the international community of users.

An online games scenario implies strong requirements on QoS for the provision in real-time of distributed multimedia content all over the world. Also usage monitoring is quite important due to the user profiling and its matching with the content (underage access to inadequate contents). Privacy, billing and community building are other properties relevant for online games and entertainment.

Summary:

Andago plans to extend its Online Games platform to use grid middleware to solve common problems found in online games portals. Technically, the goal is to be able to share expensive resources between providers and to allow billing based on usage. From a business perspective our goal is to open new commercial opportunities in the domain of entertainment.

1a: Life Sciences / 60

Early Diagnosis of Alzheimer's Disease Using a Grid Implementation of Statistical Parametric Mapping Analysis

Author: Livia Torterolo[1]

Co-authors: Andrea Schenone [2]; Barbara Canesi [1]; Elisa Molinari [1]; Francesco Beltrame [2]; Ivan Porro [1]

[1] *Bio-Lab, DIST, University of Genoa*

[2] *DIST, University of Genoa*

Corresponding Author: livia@bio.dist.unige.it

A voxel based statistical analysis of perfusional medical images may provide powerful support to the early diagnosis for Alzheimer's Disease (AD). A Statistical Parametric Mapping algorithm (SPM), based on the comparison of the candidate with normal cases, has been validated by the neurological research community to quantify ipometabolic patterns in brain PET/SPECT studies. Since suitable "normal patient" PET/SPECT images are rare and usually sparse and scattered across hospitals and research institutions, the Data Grid distributed analysis paradigm ("move code rather than input data") is well suited for implementing a remote statistical analysis use case, described as follow.

Summary:

The SPM software library was originally developed and is made freely available by the Functional Imaging Lab (FIL) at the Wellcome Department of Imaging Neuroscience (London University College) for activation studies in functional MR.
Since then, the use of SPM was extended and, through a specifically defined analysis protocol, SPM routines are presently the standard within the neurological research community as regards a voxel based analysis of PET/SPECT studies for the early diagnosis of AD.
In order to achieve correct results, the SPM software library provides a number of functionalities related to image processing and statistical analysis: normalization, co-registration, smoothing, parameter estimation, statistical mapping.
The statistical parametric mapping algorithm (the most important functionality for our goal) performs a statistical analysis in order to compare, on a voxel-by-voxel base, the perfusion values in the test images against the corresponding values in normal images.

As a result of a previous research project [1], remote access to SPM is being made available through the Italian Portal of Neuroinformatics providing doctors from peripheral hospitals with an invaluable tool to increase the "comparison database" and therefore improving the AD diagnosis. The portal contains a section entirely dedicated to the statistical analysis of PET/SPECT images, accessible by authorized users only. Doctors or researchers accessing the portal may thus be supported in running analysis tasks on suspect AD patient studies. Directly from the portal, a user can upload the suspect AD image and select the normal cases for statistical calculation.
The SPM application is available to authorized users without downloading any software tool. In order to use it, no particular hardware resource or specific computer knowledge is needed.

In order to evaluate the potential advantage of porting such an implementation to a Grid environment, it is worth noting that during the statistical parametric mapping a large set of images of normal patients is required to be used for comparison. This is because the accuracy of ipoperfusion maps is strictly related to the number of normal studies compared to the test image.
On the other hand, due to ethical issues and to the high costs of neuroimaging technologies, PET and SPECT studies on normal subjects are very rare. The NEST-DD project, funded by the European Commission [2], collected a database of about 100 images in order to make available the first large dataset for these studies.
Moreover the images of normal subjects are covered by privacy and security issues and for this reason they cannot be freely moved on the net or published by the centre that made the analysis. As a consequence, only doctors working at very large institutions, locally owning large databases of normal images, can usually carry out SPM-based analyses.
Starting from these considerations, the aim of our project has been to enable doctors from small peripheral hospitals to use large sets of normal PET/SPECT images provided by medical research institutes distributed on the net, by remotely extracting the information needed for the statistical analysis from the normal images and collecting it without moving the original image files.

Furthermore, the execution time of the analysis must be compatible with an interactive clinical application in a busy medical environment. The time required for the analysis can be reduced, since:
- some aspects of calculation could be parallelized and distributed on the computational resources associated to the remote databases of normal images;
- the time required for data transfers over the network would be reduced, since the code amounts to just few KB, compared to images sizing up to 100 MB.

The use of GRID technologies well matches all of the above issues and allows easy access to distributed data as well as to distributed computational resources.

Grid implementation has been carried on GILDA Infrastructure that provides a series of sites and services spread all over Italy and the rest of the world on which LCG and gLite middleware are installed and several Virtual Organisations are enabled.

A LCG node has been installed at University of Genoa for the implementation of biomedical applications and in particular for this application.

The objectives of the LCG-based implementation are:
- to distribute PET/SPECT images on different storage resources available on the GRID and register them on a catalogue.
- to insert and manage metadata in order to make the user able to select normal images for the statistical analysis using their own attributes.
- to access images from User Interface using Logical File Names (LFN) without moving them from storage resources.

To reach the first result, LCG File Catalog (LFC) was selected: it allows users and applications to locate files (or replicas) on the LCG Grid maintaining mappings between logical and physical file names.
As next step, the ARDA Metadata Grid Application (AMGA) has been integrated, fulfilling also the second requirement. Actually, LCG does not provide a satisfactory metadata management system and AMGA fills this hole. The collected metadata are associated to files stored on the LCG Grid through a reference on the LFC catalogue system and are used to select images directly through the portal. AMGA provides the ability to allow only certain people to access specified attributes. This is very important because all medical data should be considered as sensitive to preserve patient privacy.
To meet the third requirement, LCG Data Management and File access tools have been selected.
In particular lcg_util and Grid File Access Library (GFAL) tools was used to transparently interact with LFC catalog and to perform calls for storage management and file access using the correct protocol for file transfer.

[1] S. Scaglione, I. Castiglioni, E. Molinari, F. Cesari, F. Repetto, A. Schenone, J.Abutalebi, D. Perani, M.C. Gilardi and F. Beltrame, "Neuroinformatics portal as knowledge repository and e-service for neuroapplication and data mining", Medicon 2004 (Mediterranean Conference on Medical and Biological Engineering), Naples, July 31-August 5 2004
[2] K. Herholz, E. Salmon, D. Perani, J-C. Baron, V. Holthoff, L. Frolich, P. Schonknecht, K. Ito, R. Mielke, E. Kalbe, G. Zundorf, X. Delbeuck, O. Pelati, D. Anchisi, F. Fazio, N. Kerrouche, B. Desgranges, F. Eustache, B. Beuthien-Baumann, C. Menzel, J. Schroder, T. Kato, Y. Arahata, M. Henze, and W-D. Heiss "Discrimination between Alzheimer Dementia and Controls by Automated Analysis of Multicenter FDG PET"NeuroImage 17, 302–316 (2002)

2b: Data access on the grid / 61

Development of gLite Web Service Based Security Components for the ATLAS Metadata Interface

Author: Thomas Doherty[1]

Co-authors: Fabian Lambert [2]; Jerome Fulachier [2]; Solveig Albrand [2]

[1] *University of Glasgow*

[2] *IN2P3*

Corresponding Author: t.doherty@physics.gla.ac.uk

Introduction

AMI (ATLAS Metadata Interface) is a developing application, which stores and allows access to dataset metadata for the ATLAS experiment. It is a response to the large number of database-backed applications needed by an LHC experiment called ATLAS, all with similar interface requirements. It fulfills the need of many applications by offering a generic web service and servlet interface, through the use of self-describing databases. Schema evolution can be easily managed, as the AMI application does not make any assumptions about the underlying database structure. Within AMI data is organized in "projects". Each project can contain several namespaces (*). The schema discovery mechanism means that independently developed schemas can be managed with the same software.

This paper summarises the impact of the requirements contracted to AMI of five gLite metadata interfaces. These interfaces namely MetadataBase, MetadaCatalog, ServiceBase, FASBase and MetadaSchema [1] deal with a range of previously identified use cases on dataset (and logical files) metadata by particle physicists and project administrators working on the ATLAS experiment. The future impact on AMI architecture of the VOMs security structure and the gLite search interface are both discussed.

Fundamental Architecture of AMI

The AMI core software can be used in a client server model. There are three possibilities for a client (software installed on client side, from a browser and web services) but the relevant client with regards to grid services is the Web Services client.

Within AMI there are generic packages, which constitute the middle layer of its three-tier architecture. Command classes can be found within these packages. These classes are key to the implementation of the gLite methods in each of the interfaces. The implemented gLite interfaces are therefore situated on the server side in this middle layer and directly interface with the client tier and the command classes in this middle layer. It is possible to choose a corresponding AMI command that is equivalent to the basic requirements of each of the gLite Interface methods.

[Figure 1]

Figure 1: A Schematic View of the Software Architecture of AMI [2]. This diagram shows the AMI Compliant Databases as the top layer. This interfaces with the lowest software layer, which is JDBC. The middle layer BkkJDBC package allows for connection to both MySQL and Oracle. The generic packages contain command classes which are used in managing the databases. Application specific software in the outer layer can include the generic web search pages.

The procedure used to further understand the structure necessary to implement the gLite methods was to observe how AMI is designed to absorb commands into its middle tier mechanism. This was achieved by mapping the delegation of methods through the relevant code and is best illustrated with the use of an UML sequence diagram in figure 2.

The deployment of AMI as a web application in a web container can take place using Tomcat. To set up web services for AMI it is necessary to plug the Axis framework into Tomcat. Then with the use of WSDL and the axis tools that allow conversion from WSDL to Java client classes a Java web service client class can be deployed which communicates with the gLite interfaces.

(*) namespace is "database" in MySQL terms, "schema" in ORACLE and "file" in SQLite.

[Figure 2]

Figure 2: UML sequence diagram of basic workings of AMI. Note: A controller class delegates what command class is invoked. A router loader is instantiated to connect to a database. XML output is returned to the gLite interface implementation class.

A direct consequence of grid services is secure access. This involves authentication and authorisation of users and machines. Authorisation in AMI is handled by a local role-based mechanism. Authentication is implemented by securing the web services using grid certificates.

Currently permissions in AMI are based on a local role system. An EGEE wide role system called Virtual Organizations Membership Service (VOMS) [3] is being developed.

AMI would then have to be set up to read and understand VOMS attributes and grant permissions based on a user's role in ATLAS. Requirements analysis work is currently underway on the impact of this VOMS system on the AMI architecture.

Also directly relevant to the gLite interface was the implementation of a query language for performing cascaded searches through all projects. This implementation used a library (JFLEX) to define our own grammar rules, following the EGEE gLite Metadata Query Language (MQL) specification. It allows AMI to execute a search in a generic way on several databases of any type (MySQL, ORACLE or SQLite for example) starting only from one MQL query.

Conclusion

This paper presents a description of the implemention of the gLite Interfaces for AMI. It summarises how AMI was set up fully with these implementation classes interfacing with web service clients and how these clients are made secure with the aid of grid certificates.

AMI as mentioned provides a set of generic tools for managing database applications. AMI also supports geographical distribution with the use of web services. To implement the gLite interfaces as a wrapper to AMI using these web services provides the user with a generic and secure metadata interface. Along with the gLite search interface, any third party application should be able to plug in AMI knowing it supports a well defined API.

References

[1] Developer's Guide for the gLite EGEE Middleware - http://edms.cern.ch/document/468700
[2] ATLAS Metadata Interfaces (AMI) and ATLAS Metadata Catalogs, Solveig Albrand, Jerome Fulachier, LPSC Grenoble
[3] VOMs - http://hep-project-grid-scg.web.cern.ch/hep-project-grid-scg/voms.html

1b: Astrophysics/Astroparticle physics - Fusion - High-Energy physics / 62

The ATLAS Rome Production Experience on the LHC Computing Grid

Author: Simone Campana[1]

[1] *CERN/IT/PSS*

Corresponding Author: simone.campana@cern.ch

The Large Hadron Collider at CERN will start data acquisition in 2007. The ATLAS (A Toroidal LHC ApparatuS) experiment is preparing for the data handling and analysis via a series of Data Challenges and production exercises to validate its computing model and to provide useful samples of data for detector and physics studies. The last Data Challenge, begun in June 2004 and ended in early 2005, was the first performed completely in a Grid environment. Immediately afterwards, a new production activity was necessary in order to provide the event samples for the ATLAS physics workshop, taking place in June 2005 in Rome. This exercise offered a unique opportunity to estimate the reached improvements and to continue the validation of the computing model. In this contribution we discuss the experience of the "Rome production"on the LHC Computing Grid infrastructure, describing the achievements, the improvements with respect to the previous Data Challenge and the problems observed, together with the lessons learned and future plans.

2a: Workload management and Workflows / 63

BOSS: the CMS interface for job summission, monitoring and book-keeping

Authors: Barry MacEvoy[1]; Claudio Grandi[2]; David Colling[1]; Giuseppe Codispoti[2]; Paolo Capiluppi[2]; Stuart Wakefield[1]; William Bacchi[2]; Yong-Jun Zhang[1]

[1] *Imperial College London*

[2] *Universita di Bologna*

Corresponding Author: giuseppe.codispoti@cern.ch

BOSS (Batch Object Submission System) has been developed in the context of the CMS experiment to provide logging and bookkeeping and real-time monitoring of jobs submitted to a local farm or a grid system. The information is persistently stored in a relational database (right now MySQL or SQLite) for further processing. In this way the information that was available in the log file in a free form is structured in a fixed-form that allows easy and efficient access. The database is local to the user environment and is not requested to provide server capabilities to the external world: the only component that interacts with it is the BOSS client process.
BOSS can log not only the typical information provided by the batch systems (e.g. executable name, time of submission and execution, return status, etc⋯), but also information specific to the job that is being executed (e.g. dataset that is being produced or analyzed, number of events done so far, number of events to be done, etc⋯). This is done by means of user-supplied filters: BOSS extracts the specific user-program information to be logged from the standard streams of the job itself filling up a fixed form journal file to be retrieved and processed at the end of job running via the BOSS client process.
BOSS interfaces to a local or grid scheduler (e.g. LSF, PBS, Condor, LCG, etc⋯) through a set of scripts provided by the system administrator, using a predefined syntax. This allow hiding to the upper layers its implementation details, in particular whether the batch system is local or distributed. The interface provides the capability to register, un-register and list the schedulers. BOSS provides an interface to the local scheduler for the operations of job submission, deletion, querying and output retrieval. At output retrieval time the information in the database is updated using information sent back with the job.
BOSS provides also an optional run-time monitoring system that, working in parallel to the logging system, collects information while the computational program is still running, and presents it to the upper layers through the same interface. The real-time information sent by the running jobs are collected in a separate database server, the same real-time database server may support more than one BOSS database. The information in the real-time database server has a limited lifetime: in general it is deleted after that the user has accessed it, and in any case after successful retrieval of the journal file. It is not possible to use the information in the real-time database server to update the logging information in the BOSS database once the journal file for the related job has been processed.
The run-time monitoring is made through a pair client-updater registered as a plug-in module: they are the only components that interact with the real time database. The real-time updater is a client of the real-time database server: it sends the information of the journal file to the server at pre-defined intervals of time. The real-time client is a tool used by BOSS to update his database using the real-time information.
The interface with the user is made through:
a command line , kept as similar as possible to the one of the previous versions; it is the minimal way to access BOSS functionalities to give a straightforward test and training instrument;
C++ API, increasing functionalities and ease-to-use for programs using BOSS: currently it is under development and is meant to grown-up with the users requirements;
Python API, giving almost the same functionalities of the C++ one, plus the possibility to run BOSS from a python command line.
User programs may be chained together to be executed by a single batch unit (job). The relational structure supports not only multiple programs per job (program chains) but also multiple jobs per chain (in the event of job resubmission). Homogeneous jobs, or better "chains of programs", may be grouped together in tasks (e.g. as a consequence of the splitting of a single processing chain into many processing chains that may run in parallel). The description of a task is passed to BOSS through an

XML file, since it can model its hierarchical structure in a natural way.
The process submitted to the batch scheduler is the BOSS job wrapper. All interactions of the batch scheduler to the user process pass through the BOSS wrapper.
The BOSS job wrapper starts the chosen chaining tool, and optionally the real-time updater. An internal tool for chaining programs linearly is implemented in BOSS but in future external chaining tools may be registered to BOSS so that more complex chaining rules may be requested by the users. BOSS will not need to know how they work and will just pass any configuration information transparently down to them.
The chaining tool starts a BOSS "program wrapper"for each user program.The program wrapper starts all processes needed to get the run-time information from the user programs into the journal file. This program wrapper is unique and it has to be started passing only one parameter, the program id.
The BOSS client determines finished jobs by a query to the scheduler. It retrieves the output for those jobs and uses the information in the journal file to update the BOSS database.
The BOSS client pops the information about running jobs from the real-time database server through the client part of the registered Real Time Monitor. It also deletes from the server the information concerning jobs for which the BOSS database has already been updated using the journal file. The information extracted from the real-time database server may be used to update the local BOSS database or just to show the latest status to the user.

1b: Astrophysics/Astroparticle physics - Fusion - High-Energy physics / 64

Massive Ray Tracing in Fusion Plasmas on EGEE

Authors: Eduardo Huedo[1]; Ignacio M. Llorente[2]; Jose Luis Vazquez-Poletti[2]; Ruben S. Montero[2]

[1] *Centro de Astrobiología (Spain)*

[2] *Universidad Complutense de Madrid (Spain)*

Corresponding Author: jlvazquez@fdi.ucm.es

Plasma heating in magnetic confinement fusion devices can be performed by launching a microwave beam with frequency in the range of the cyclotron frequency of either ions or electrons, or close to one of their harmonics. The Electron Cyclotron Resonance Heating (ECRH) is characterized by the small size of the wavelength that allows one to study the wave properties using the geometrical optics approximations. This means that the microwave beam can be simulated by a large amount of rays. If there is no critical plasma layer (like cut off or resonance) close to the beam waist, it is possible to use the far field approximation and the beam can be simulated by a bunch of one or two hundred rays, which can be performed in a cluster. However, if the beam waist is closed to the critical layer and the heating method uses Electron Bernstein Waves (EBW), the number of rays needed is much larger. Being all the ray computations independent, this problem is well suited to be solved in the grid relying on the EGEE infrastructure [1].

We have developed a MRT (Massive Ray Tracing) framework using the lcg2.1.69 User Interface C++ API. It sends over the grid the single ray tracing application (called Truba [2]) which performs the tracing of a single ray. This framework works in the following way: First of all, a launcher script generates the JDL files needed. Then, the MRT framework launches all the single ray tracing jobs simultaneously, periodically querying each job's state. And finally, it retrieves the job's output.

We performed several experiments in the SWETEST VO with a development version of Truba, whose average execution time on a Pentium 4 3.20 GHz is 9 minutes. Truba's executable file size is 1.8 MB, input file size is 70 KB, and output file size is about 549 KB. In the SWETEST VO, there were resources from the following sites: LIP (16 nodes, Intel Xeon CPU 2.80 GHz), IFIC (117 nodes, AMD Athlon 1.2 Ghz), PIC (69 nodes, Intel Pentium 4 2.80 GHz), USC (100 nodes, Intel Pentium III 1133 MHz), IFAE

(11 nodes, Intel Pentium 4 2.80 GHz) and UPV (24 nodes, Pentium III). All Spanish sites are connected by RedIRIS, the Spanish Research and Academic Network. The minimum link bandwidth is 622 Mbps and the maximum, 2.5 Gbps.

The MRT framework traced 50 rays and it took an overall time of 88 minutes. In this case, we analyzed the following parameters: execution time (how much time took Truba to be executed in the remote resource not including queue time), transfer time, overhead (how much overhead is introduced by the Grid and the framework itself due to all the inner nodes and stages the job passes through) and productivity (number of jobs per time unit). The average execution time was 10.09 minutes and its standard deviation was 2.97 minutes (this is due to the resource heterogeneity). The average transfer time was 0.5 minutes and its standard deviation was 0.12 minutes (this is due to dynamic network bandwidth). The average overhead was 29.38 minutes. Finally, the productivity was 34.09 rays/hour.

Nevertheless, we found the lack of opportunistic migration (some jobs remained "Scheduled"for too long) and fault tolerance mechanisms (specially during submission using Job Collections, retrieving output and some "Ready"status that were really "Failed"and took too long to be rescheduled) as limitations of the LCG-2 infrastructure (some of the nodes marked by the GOC as "OK"were not). Even, problems handling Job Collections and submitting more than 80 jobs were found.

In order to bypass these problems, we used GridWay, a light-weight framework. It works on top of Globus services, performing job execution management and resource brokering, allowing unattended, reliable, and efficient execution of jobs, array jobs, or complex jobs on heterogeneous, dynamic and loosely-coupled Grids. GridWay performs all the job scheduling and submission steps transparently to the end user and adapts job execution to changing Grid conditions by providing fault recovery mechanisms, dynamic scheduling, migration on-request and opportunistic migration [3]. This scheduling is performed using the data gathered from the Information System (GLUE schema) that is part of the LCG-2 infrastructure.

GridWay performs the job execution in three simple steps: Prolog, which prepares the remote system by creating an experiment directory and transferring the needed files. Wrapper, which executes the actual job and obtains its exit status code. And Epilog, which finalizes the remote system by transferring the output back and cleaning up the experiment directory.

After performing different experiments in similar conditions, we obtained the following results. The overall time was 65.33 minutes. The average execution time was 10.06 minutes and its standard deviation was 4.32 minutes (this was almost the same with the pilot application). The average transfer time was 0.92 minutes and its standard deviation was 0.68 minutes (this was higher because of the submission of the Prolog and Epilog scripts). The average overhead was 22.32 minutes (this was lower as less elements were taking part in the scheduling process). And finally, the productivity was 45.92 rays/hour.

The reason for this higher productivity is that GridWay reduces the number of nodes and stages the job passes through. Also, this productivity is the result of GridWay's opportunistic migration and fault tolerance mechanisms.

As a key improvement needed to better exploit this technique on EGEE we can find that the data contained in the Information System should be updated more frequently and should represent the real situation of the remote resource when trying to submit a job to it. This is a commitment between the resource administrator and the rest of the EGEE community.

The last aspect we would like to notice is the difference between the LCG-2 API and DRMAA. While the LCG-2 API relays on a specific middleware, DRMAA (which is a GGF standard) doesn't. The scope of this user API specification is all the high level functionality which is necessary for an application to consign a job to a DRM system, including common operations on jobs like synchronization, termination or suspension. In case this abstract is accepted, we would like to perform an on line demonstration.

REFERENCES:
[1] Massive Ray Tracing in Fusion Plasmas: Memorandum of Understanding. Francisco Castejón. CIEMAT. Spain.
[2] Electron Bernstein Wave Heating Calculations for TJ-II Plasmas. Francisco Castejón, Maxim A. Tereshchenko, et al. American Nuclear Society. Volume 46, Number 2, Pages 327-334, September 2004.
[3] A Framework for Adaptive Execution on Grids. E. Huedo, R. S. Montero and I. M. Llorente. Software - Practice & Experience 34 (7): 631-651, June 2004.

2a: Workload management and Workflows / 65

G-PBox: A framework for grid policy management

Authors: Andrea Caltroni[1]; Andrea Ferraro[2]

Co-authors: Antonia Ghiselli [2]; Gian Luca Rubini [2]; Vincenzo Ciaschini [2]

[1] *INFN*

[2] *INFN-CNAF*

Sharing computing and storage resources among multiple Virtual Organizations which group people from different institutions often spanning many countries, requires a comprehensive policy management framework.
This paper introduces G-PBox, a tool for the management of policies which integrates with other VO-based tools like VOMS, an attribute authority and DGAS an accounting system, to provide a framework for writing, administering and utilizing policies in a Grid environment.

Summary:

Introduction
One of the most innovative concept of Grid computing is the Virtual Organization (VO) which represents a distributed community of users sharing a common goal.
The VO is key to administering the allocation of grid resources and data access but is also influencing all aspects of the Grid environment from authentication and authorization to accounting. It has, for example, been included in the attributes of the user certificate.
VOs and resource owners share contracts to regulate resource usage. These contracts or policies can be difficult to enforce or manage just by formal agreements and some institutions might be reluctant to participate in collaborative efforts because of the lack of an automatic enforcement of policies.

G-PBox
Current Grid middleware software allows resource owners control over their own resources only. It is necessary that VO administrators are able to issue agreements on resource usage with resources owners.
G-PBox helps manage the administrative domain of a VO (but also the domain of a resource owner), storing all policies concerning the domain itself.
A distribution mechanism allows policies to be delivered to their intended targets and repositories to be synchronized.

G-PBoxes form a network which allows the creation of policies on any administrative domain.
An authoritative mechanism permits to grant a domain control over other domains.
Policies are described in the XACML language, a high level language for encoded data exchange.
The actual usability by G-PBox of a specific category of policies may depend on the availability of external information such as the ones provided by an accounting tool

or an information system. The communication with these external tools is performed through a plug-in mechanism.

G-PBox INTERNAL interactions
G-PBox interacts with resource owners and VOs through a Policy Enforcement Point (PEP). PEPs have been implemented for the LCG Resource Broker (RB) and the LCG Computing Element (CE) using the LCAS/LCMAPS plugin technology.
A PEP for a LCG Storage Element (SE) should use the same CE LCAS plugin features.
An example of interaction with the RB is how to apply policies to the matchmaking process performed by the resource broker (RB). The first such request we got was to have a RB capable of splitting resources in a series of classes, each with its own priority, and then split job assignment to resources based on such priority and the user's VOMS credentials.
The chosen solution was to require a resource to publish a tag describing their class in the information system , and then write policies associating a specific group/role combination to a class of resources. At this point, the PEP plugin for the RB only had to contact the G-PBox and provide it with the following parameters: the action (job-submission), the credentials of the user and a list of suitable resources, each with its associated tag. It would then obtain as a result a set of allow/deny answers for each resource describing whether the user was allowed to submit a job or not.
Another PEP we implemented was a PEP for the Computing Element (CE), whose job was to take over grid user mapping to local accounts based on given policies.

G-PBox EXTERNAL interactions
Among the external components required or necessary for the implementation of a specific feature, we can underline:
- VOMS: An attribute authority, is a required component for the basic functionality of G-PBox
- DGAS or any other accounting system is necessary to implement policies which need accounting data

VOMS handles the authorization part of the security mechanism allowing a user to provide authorization data as she tries to access a resource provider. The type of data VOMS handles is information about a user's relationship with the virtual organizations she belongs to. This information is described by VOMS using the concept of groups a user belongs to, roles a user is allowed to impersonate and capabilities a user should present a resource provider for special processing needs.
DGAS implements Resource Usage Metering, Accounting and Account Balancing (through resource pricing) in a distributed Grid environment. Accounting requires accurate Usage Metering which is performed by lightweight sensors installed on the Computing Elements.

FUTURE plans
G-PBox is a relatively young tool and is still in the development process. The first testing experience of the G-PBox framework, related to different groups of a VO and enforced by a Grid Resouce Broker and checked by the CEs, demonstrated the effectiveness of such an approach. Other policies, like CPU fair sharing and storage quota management, have been required and are going to be implemented.
The current G-PBox v1.0 is included in the gLite distribution from release 1.5.

References
[1] V. Ciaschini, A. Ferraro, A. Ghiselli, G. Rubini, R. Zappi, A. Caltroni. G-PBox: a policy framework for Grid environments. In Proceedings CHEP04, September 2004.
[2] R. Alfieri, R. Cecchini, V. Ciaschini, L. dell'Agnello, A . Frohner, A. Gianoli, K. Lorentey, F. Spataro. VOMS, an Authorization System for Virtual Organizations. 1st European Across Grids Conference, Santiago de Compostela, February 13-14, 2003.
[3] A. Guarise. DataGrid Accounting System - Architecture v 1.0. DataGrid-01-TED-0126-1_0 - Feb. 14, 2003.
[4] OASIS eXtensible Access Control Markup Language (XACML)Version2.0, http://www.oasis-open.org/ committees/tc_home.php?wg_abbrev=xacml
[5] The Distributed Grid Accounting System (DGAS), http://www.to.infn.it/grid/accounting/main.html
[6] VOMS at INFN Authorization Working Group, http://grid-auth.infn.it

[7] The G-PBox Home Page at INFN, http://infnforge.cnaf.infn.it/gpbox

1a: Life Sciences / 66

BioDCV: a grid-enabled complete validation setup for functional profiling

Author: Silvano Paoli[1]

Co-authors: Annalisa Barla [2]; Cesare Furlanello [2]; Davide Albanese [2]; Giuseppe Jurman [2]; James F. Reid [3]; Roberto Flor [2]; Stefano Cozzini [4]; Stefano Merler [2]

[1] *ITC-irst*

[2] *ITC-irst, Trento, Italy*

[3] *INT/IFOM-FIRC, Milano, Italy*

[4] *INFM/Democritos, Trieste, Italy*

Corresponding Author: silpaoli@itc.it

Abstract
BioDCV is a distributed computing system for the complete validation of gene profiles. The system is composed of a suite of software modules that allows the definition, management and analysis of a complete experiment on DNA microarray data. The BioDCV system is grid-enabled on LCG/EGEE middleware in order to build predictive
classification models and to extract the most important genes on large scale molecular oncology studies. Performances are evaluated on a set of 6 cancer microarray datasets of different sizes and complexity, and then compared with results
obtained on a standard Linux cluster facility.

Introduction
The scientific objective of BioDCV is a large scale comparison of prognostic gene signatures from cancer microarray datasets realized by a complete validation system and run in Grid. The models will constitute a reference experimental landscape for new studies. Outcomes of BioDCV consist of a predictive model, the straighforward evaluation of its accuracy, the lists of genes ranked for importance, the identification of patient subtypes. Molecular oncologists from medical research centers and collaborating bioinformaticians are currently the target end-users of BioDCV. The comparisons presented in this paper demonstrate the factibility of this approach on public data as well as on original microarray data from IFOM-Firc. The complete validation schema developed in our system involves an intensive replication of a basic classification task on resampled versions of the dataset. About 5x105 base
models are developed, which may become 2x106 if the experiment is replicated with randomized output labels. The scheme must ensure that no selection bias effect is contaminating the experiment. The cost of this caution is high computational complexity.

Porting to the Grid
To guarantee fast, slim and robust code, and relational access to data and a model descriptions, BioDCV was written in C and interfaced with SQLite (http://www.sqlite.org), a database engine which supports concurrent access and transactions useful in a distributed environment where a dateset may be replicated for up to a few million models. In this paper, we present the porting of our application to grid systems, namely the Egrid (http://www.egrid.it) computational grids. The Egrid infrastructure is based on Globus/EDG/LCG2 middleware and is

integrated as an independent virtual organization within Grid.it, the INFN production
grid. The porting requires just two wrappers, one shell script to submit jobs and one
C MPI program. When the user submits a BioDCV job to the grid, the grid middleware looks for the CE (Computing Element: where user tasks are delivered) and the WNs (Worker Nodes: machines where the grid user programs are actually executed) require to run the parallel program. As soon as the resources (CPUs in WNs) are available, the shell script wrapper is executed on the assigned CE. This script distributes the microarray dataset from the SE (Storage Element stores user data in the grid) to all the involved WNs. It then starts the C MPI wrapper which spawns several instances of the BioDCV program itself. When all BioDCV instances are completed, the wrapper copies all outputs including model and diagnostic data from the WNs to the starting SE. Finally, the process outputs are returned, thus allowing the reconstruction of a complete data archive for the study.

Experiments and results
Two experiments were designed to measure the performance of the BioDVC parallel application in two different computing available environments: a standard Linux cluster and a computational grid.
In Benchmark 1, we study the scalability of our application as a function of the number of CPUs. The benchmark is executed on a Linux clusters formed by 8 Xeon 3.0 CPUs and on the EGEE grid infrastructure ranging from 1 to 64 Xeon CPUs. Two DNA microarray datasets are considered: LiverCanc (213 samples, ATAC-PCR, 1993 genes) and
PedLeuk (327 samples, Affymetrix, 12625 genes). On both dataset we obtain a speed-up curve very close to linear. The speed-up factor for n CPUs is defined as the user time for one CPU divided by the user time for n CPUs.
In Benchmark 2, we characterize the BioDCV application different d (number of features) and N (number of samples) values for a complete validation experiment, and we execute a task for each dataset on the EGEE grid infrastructure using a fixed number of CPUs. The benchmark was run on a suite of six microarray datasets: LiverCanc, PedLeuk, BRCA (62 samples, cDNA, 4000 genes), Sarcoma (35 samples, cDNA, 7143 genes), Wang (286 samples, Affymetrix, 17816 genes), Chang (295 samples, cDNA, 25000 genes). It can be observed that effective execution time (total execution time without queueing time at grid site) increases linearly with the dataset footprint, i.e. the product of number of genes and number of samples. The performance penalty payed with respect to a standard parallel run performed on local cluster is limited and it is mainly due to data transfer from user machine to grid site and between WNs.

Discussion and Conclusions
The two experiments, which sum up to 139 CPU days within the Egrid infrastructure, implicate that general behavior of the BioDCV system on LCG/EGEE computational grids can be used in practical large scale experiments. The overall effort for gridification was limited to three months. We will investigate if substituting a model of one single job asking for N CPUs (MPI approach) with a model that submits N different single CPU jobs can overcome some limitations. Next step is porting our system under EGEE's Biomed VO.

BioDCV is an open source application and it is currently distributed under GPL (SubVersion repository at http://biodcv.itc.it).

1c: Earth Observation - Archaeology - Digital Library / 67

Solid Earth Physics on EGEE

Author: Eric Clévédé[1]

Co-authors: Geneviève Moguilny [1]; Geneviève Patau [1]; Gwendoline Pajot [1]

[1] *Institut de Physique du Globe de Paris*

This abstract describes the "Solid Earth Physics" applications of the ESR(Earth Science Research) VO. These applications, developed or ported by the "Institut de Physique du Globe de Paris" (IPGP) address mainly seismology, data processing as well as simulation.
Solid Earth Physics deployed successfully two applications on EGEE.
The first one allows the rapid determination of earthquake mechanisms, and the second one, SPECFEM3D, allows numerical simulation of earthquakes in complex three-dimensional geological models.
A third application, currently being ported, will allow gravity gradiometry studies from GOCE satellite data.

1) Rapid determination of Earthquake centroid moment tensor (E. Clévédé, IPGP)

The goal of this application is to provide first order informations on seismic source for large Earthquakes occurring worldwide.
These informations are: the centroid, which corresponds to the location of the space-time barycenter of the rupture; the first moments of the rupture in the point-source approximation,
which are the scalar moment giving the seismic energy released (from which the moment magnitude is deduced), the source duration, and the moment tensor that describes the global mechanism of the source (from which is deduced the orientation of the rupture plane and the kind of displacement on this plane).
The data used are three-components long-period seismic signals (from 1 to 10 MHz) recorded worldwide. In the case of a 'rapid' determination we use data from the GEOSCOPE network that allows us to obtain records from a dozen of stations within a few hours after the occurrence of the event.
In order to deal with the trade-off between centroid and moment tensor determinations, the centroid and the source duration are estimated by an exploration over
a space-time grid (longitude, latitude, depth and source duration).
When the centroid is supposed to be known and fixed, the relation between the moment tensor and the data is linear.
Then, for each point of the centroid parameter space, we compute Green functions (one for each of the 6 elements of the moment tensor) for each receiver, and proceed to linear inversions in the spectral domain, for each different source durations.
The best solution is determined by the data fit.

This application is well adapted to the EGEE grid, as each point of the centroid parameter space can be treated independently, the main part of the time computation being the Green functions computation.
For a single point, a run is performed in a few minutes.
In a typical case, an exploration
grid (longitude, latitude, depth and source duration) of 10x10x10x10 requires about 100h of time computation, which is reduced to about 1 hour over a hundred different jobs submitted to the EGEE grid.

The new features for workflow provided by gLite should allow the simplification of the management of the different steps of a run.

2) SPECFEM3D: Numerical simulation of earthquakes in complex three-dimensional geological models (D. Komatitsch MIGP; G. Moguilny, IPGP)

The spectral-element method (SEM) for regional scale seismic wave propagation problems is used to model wave propagation at high frequencies and for complex geological structures.

Simulations based upon a detailed sedimentary basin model and this accurate numerical technique produce generally nice waveform fits between the data and 3-D synthetic seismograms. Moreover, remaining discrepancies between the data and synthetic seismograms could ultimately be utilized to improve the velocity model based upon a structural inversion, or the source parameters based upon a centroid moment-tensor (CMT) inversion.

This application, written in Fortran 90 and using MPI, is very scalable and already ran outside EGEE on 1994 processors in the Japanese Earth Simulator, and inside EGEE on 64 processors at Nikhef (NL).

The amount of disk space and memory depend on the input parameters but are never very large. However, this application has some technical constraints : the I/O have to be done in local files (on each node) and on shared files (seen by all nodes), and the script must be able to submit 2 executable files sequentially, which use the same nodes in the same order. This is because the SPECFEM3D software package consists of two different codes, a mesher and a solver, which work on the same data.

Some successful tests have been done with gLite but the problem of differentiate a node (with several CPUs) and a CPU when requiring the resources, doesn't seem to be solved.

It also will be interesting to have access to "fast clusters" (with high throughput and low latency networks, as Myrinet, SCI…), and, to access larger configurations, by having the possibility to access various sites during a given run.

3) Gravity gradiometry (G. Pajot, IPGP)

The GOCE satellite (see [1]) is to be launched by the European Space Agency by the end of this year. Onboard is an instrument, called a gradiometer, which measures the spatial derivatives of the gravity field in three independent directions of space. Although gravity gradiometry was born more than a century ago and successfully used for geophysical prospecting, GOCE satellite will provide the first set of gravity gradiometry data on the whole Earth with unprecedented spatial resolution and accuracy and specific methods have to be developed. Thanks to these data, we will be able to derive information about the Earth inner mass distribution patterns at various scales (from the sedimentary basin to the Earth's Mantle).

To this aim, we develop a pseudo Monte Carlo inversion method (see [2]) to interpret GOCE data. One step of it is the model generation, which is the limiting factor of it. A model is a possible density distribution, to which correspond calculated gravity gradients as they would be measured by the instrument. These calculated gradients are compared to those actually measured; the nearer they are from measured ones, the closer the model is from real Earth. One rough pseudo random model takes about 5 minutes to be generated on a 2.8 GHz CPU, finest ones generation reaches 20 minutes and a set of 1000 models is a good basis to start the model space exploration, each one being independent from the others. Thus, EGEE is the perfect frame to develop such an application. We test and validate our algorithm using a set of marine gradiometry measurements provided by the Bell Geospace Company. These data need a frequent restricted access. First results of the application and solutions to the confidentiality problem are exposed here.

References:
[1] http://ganymede.ipgp.jussieu.fr/frog/
[2] Geophysical Inversion with a Neighbourhood Algorithm -I. Searching a parameter space,*Sambridge, M.,*Geophys. J. Int., **138*, 479-494, 1999.

In conclusion, the main goal of these three applications is to create a Grid-based infrastructure to process, validate and exchange large sets of data within the worldwide Solid Earth physics community as well as to provide facilities for distributed computing. The stability of the infrastructure and the easiness to use the Grid are prerequisites to reach these objectives and bring the community to use the Grid facilities.

1c: Earth Observation - Archaeology - Digital Library / 68

Diligent and OpenDLib: long and short term exploitation of a gLite Grid Infrastructure

Author: Pasquale Pagano[1]

Co-authors: Andrea Manzi [1]; Davide Bernardini [1]

[1] *CNR-ISTI*

Corresponding Author: pasquale.pagano@isti.cnr.it

The demand for Digital Libraries has recently grown considerably, DLs are perceived as a necessary instrument to support communication and collaboration among the members of communities of interest; many application domains require DL services, e.g. e-Health, e-Learning, e- Government, and many of the organizations that demand a DL are small, distributed, and dynamic, because they use the DL to support temporary activities such as courses, exhibitions, projects, etc.
Nowadays the construction and management of a DL requires high investments and specialized personnel because the content production is very expensive and multimedia handling requires high computational resources. The effect are that years are spent in designing and setting up a DL and that the DL systems lack interoperability and the services provided are difficult to reuse.
This development model is not suitable to satisfy the demand of many organizations, so the purpose of DILIGENT is to create a Digital Library Infrastructure that will allow members of dynamic virtual research organizations to create on-demand transient digital libraries based on shared computing, storage, multimedia, multi-type content, and application resources. Following this vision Digital libraries are not ends in themselves; rather they are enabling technologies for digital asset management, electronic commerce, electronic publishing, teaching and learning, and other activities.
DILIGENT is a is a three-year European funded project that aims at developing a test-bed DL infrastructure able to create a multitude of DLs on-demand, manage the resources of a DL (possibly provided by multiple organizations), and operate the DL during its lifetime. These DLs created by DILIGENT will be active on the same set of shared resources: content sources (i.e. repositories of information searchable and accessible), services (i.e. software tools, that implement a specific functionality and whose descriptions, interfaces and bindings are defined and publicly available) and hosting nodes (i.e. networked entities that offer computing and storage capabilities and supply an environment for hosting content sources and services).
By exploiting appropriate mechanisms provided by the DL infrastructure, producer organizations register their resources and provide a description of them. The infrastructure manages the registered resources by supporting their discovering, reservation, monitoring and by implementing a number of functionalities that aim at supporting the required controlled sharing and quality of service.
The composition of a DL is dynamic since the services of the infrastructure continuously monitor the status of the DL resources and, if necessary, change the components of the DL in order to offer the best quality of service. By relying on the shared resources many DLs, serving different communities, can be created and modified on-the-fly, without big investments and changes in the organizations that set them up.
The DILIGENT infrastructure is being constructed by implementing a service oriented

architecture in a Grid framework. The DILIGENT design will be service oriented in order to provide as many reusable components as possible for other e-applications that could be created on top of the basic DILIGENT infrastructure. Furthermore, DILIGENT exploits the Grid middleware, gLite, and the Grid production infrastructure released by the Enabling Grid for E-Science in Europe (EGEE) project. By merging a service-oriented approach with a Grid technology we can exploit the advantages of both. In particular, the Grid provides a framework where a good control of the shared resources is possible. By taking full advantage of the scalable, secure, and reliable Grid infrastructure each DL service will provide an enhanced functionality with respect the equivalent non-Grid-aware service. Moreover, the gLite Grid enables the execution of very computational demanding applications, such as those required to process multimedia content. DILIGENT will enhance existing Grid services with the functionality needed to support the complex services interactions required to build, operate and maintain transient virtual digital libraries.
In order to support the services of the DILIGENT framework and the user community expectations some key Grid services are needed: the Grid infrastructure should support a cost-effective DL operational model based on transient, flexible, coordinated "sharing of resources", address the main DL architecture requirements (distribution, openness, interoperability, scalability, controlled sharing, availability, security, quality), provide a basic common infrastructure for serving several different application domains and offer high storage and computing capabilities that enable the provision of powerful functionality on multimedia content e.g. images and videos.
From the conceptual point of view the services that implement the DILIGENT infrastructure are organized in a layered architecture.
The top layer, i.e. the Presentation layer, is user-oriented. It supports the automatic generation of user-community specific portals, providing personalized access to the DLs.
The Workflows layer contains services that make it possible to design and verify the specification of workflows, as well as services ensuring their reliable execution and optimization. Thanks to these set of services it is possible to expand the infrastructure with new and complex services capable to satisfy unpredicted user needs.
The DL Components layer contains the services that provide the DL functionalities. Key functionalities provided by this area are: management of metadata; automatically translation for achieving metadata interoperability among disparate and heterogeneous content sources; content security through encryption and watermarking; archive distribution and virtualization; distributed search, access, and discovery; annotation; cooperative work through distributed workspace management.
The services of the lower architectural layer, the Collective Layer, jointly with those provided by the gLite Grid middleware released by the EGEE project, manage the resources and applications needed to run DLs. The set of resources and the sharing rules are complex since multiple transient DLs are created on-demand and are activated simultaneously on these resources.
Following the first tests performed on the first releases of the gLite middleware the following Grid requirements were identified: it should be possible to query for the maximum number of CPUs concurrently available in order to allow to a DILIGENT high level service to automatically prepare a DAG where each node will be entitled to process a partition of the data collection, to use parametric jobs/automatic partitioning on data, to support service certificate for a high level service, to specify a job specific priority, to specify a priority for a user or for a service, to ask for on-disk encryption of data, to dynamically manage VO creation and to dynamically support user/service affiliation to a VO.
DILIGENT will be demonstrated and validated by two complementary real-life application scenarios: one from the culture heritage domain, one from the environmental e-Science domain. The former is an interesting challenge thanks to the multidisciplinary collaborative research, the image based retrieval, the semantic analysis of images, and the support for research and teaching. The latter obliges DILIGENT to manage a wide variety of content types (maps, satellite images, etc.) with very large, dynamic data sets in order to support community events, report generation, disaster recovery.
The DILIGENT project collaborates with EGEE mainly through technical interactions (technical meetings (mainly with JRA1), gLite mailing lists subscription, tutorial)

and feedback on EGEE activities and on DILIGENT project (gLite bugs submission and grid related DL requirements).
Now DILIGENT has two independent infrastructures (gLite v1.4): a Development Infrastructure (DDI) and a Testing infrastructure (DTI). These infrastructures are geographically distributed, linking 6 sites in Athens, Budapest, Darmstadt, Pisa, Innsbruck and Rome. We are running gLite experimentation tests on these infrastructures since July 2005 and we collected some useful data about data and job management.
As first approach to exploit the gLite Grid storing and processing on demand capabilities, we developed two experimental brokers that, starting from an existing digital library management system, named OpenDLib, allow interfacing the DDI.
The gLite SE broker provides OpenDLib services with the pool of SEs available via the gLite software. Moreover, it optimizes the usage of the available SEs. In particular, this service interfaces the gLite I/O server to perform the storage (put) and withdrawal (rm) of files and the access to them (get). In designing this service one of our main goals was to provide a workaround to two main problems, i.e. inconsistence between catalog and storage resource management systems, and failure without notification in the access or remove operations. Although the gLite SE broker could not improve the reliability of the requested operations we designed it in such a way to: (i) monitor its requests, (ii) verify the status of the resources after the processing of the operations, (iii) repeat the registration in the catalog and/or storage of the file until it is considered correct or unrecoverable, (iv) return a valid message reporting the exit status of the operation.
The gLite WMS wrapper provides to the other OpenDLib services with the computing power supplied by gLite CEs. Actually, the goal of this service is to provide an higher level interface than those provided by the gLite components for managing jobs, i.e. applications that can run on CEs, and DAGs, i.e. direct acyclic graphs of dependent jobs. The gLite WMS broker has therefore been designed to: (i) deal with more than one WMS, (ii) monitor the quality of service provided by these WMSs by analyzing the number of managed jobs and the average time of their execution, and, finally, (iii) monitor the status of each submitted job querying the Logging and Bookkeeping (LB) service.

1a: Life Sciences / 69

BIOINFOGRID: Bioinformatics Grid Application for life science

Author: Luciano Milanesi[1]

Co-authors: Andreas Gisel [1]; Giorgio Maggi [2]; Giovanni Meloni [3]; Ivan Merelli [1]; Mirco Mazzuccato [4]; Pietro Lio' [5]; Sabino Liuni [1]; Sándor Suhai [6]; Vincent Breton [7]; Yannick Legre [7]

[1] *National Research Council - Institute of Biomedical Technologies*

[2] *Dipartimento di Fisica,INFN Sezione di Bari,Via Amendola 173, Bari, Italy*

[3] *CONSORZIO INTERUNIVERSITARIO LOMBARDO PER L'ELABORAZIONE AUTOMATICA, Via Raffaelo Sanzio 4, Segrate, 20090, Italy*

[4] *NAZIONALE DI FISICA NUCLEARE, Via Marzolo 8, Padova, 35131, Italy*

[5] *THE CHANCELLOR, MASTERS AND SCHOLARS OF THE UNIVERSITY OF CAMBRIDGE, the old Schools, Trinity Lane, Cambridge, CB2 1TS, United Kingdom*

[6] *DEUTSCHES KREBSFORSCHUNGSZENTRUM HEIDELBERG, Im Neuenheimer Feld 280, Heidelberg, 69120, Germany*

[7] *CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE , 3, rue Michel-Ange, Paris, 16, 75794, France.*

Corresponding Author: luciano.milanesi@itb.cnr.it

Project descriptions

The European Commission promotes the Bioinformatics Grid Application for life science (BIOINFOGRID) project. The BIOINFOGRID project web site will be available at http://www.itb.cnr.it/bioinfogrid.

The project aims to connect many European computer centres in order to carry out Bioinformatics research and to develop new applications in the sector using a network of services based on futuristic Grid networking technology that represents the natural evolution of the Web.

More specifically the BIOINFOGRID project will make research in the fields of Genomics, Proteomics, Transcriptomics and applications in Molecular Dynamics much easier, reducing data calculation times thanks to the distribution of the calculation at any one time on thousands of computers across Europe and the world.

Furthermore it will provide the possibility of accessing many different databases and hundreds of applications belonging to thousands of European users by exploiting the potential of the Grid infrastructure created with the EGEE European project and coordinated by CERN in Geneva.

The BIOINFOGRID project foresees an investment of over one million euros funded through the European Commission's "Research Infrastructures"budget.
Grid networking promises to be a very important step forward in the Information Technology field. Grid technology will make a global network made up of hundreds of thousands of interconnected computers possible, allowing the shared use of calculating power, data storage and structured compression of data. This goes beyond the simple communication between computers and aims instead to transform the global network of computers into a vast joint computational resource.

Grid technology is a very important step forward from the Web, that simply allows the sharing of information over the internet. The massive potential of Grid technology will be indispensable when dealing with both the complexity of models and the enormous quantity of data, for example, in searching the human genome or when carry out simulations of molecular dynamics for the study of new drugs.

The grid collaborative and application aspects.

The BIOINFOGRID projects proposes to combine the Bioinformatics services and applications for molecular biology users with the Grid Infrastructure created by EGEE (6th Framework Program). In the BIOINFOGRID initiative we plan to evaluate genomics, transcriptomics, proteomics and molecular dynamics applications studies based on GRID technology.

Genomics Applications in GRID
• Analysis of the W3H task system for GRID.
• GRID analysis of cDNA data.
• GRID analysis of the NCBI and Ensembl databases.
• GRID analysis of rule-based multiple alignments.
Proteomics Applications in GRID
• Pipeline analysis for domain search for protein functional domain analysis.
• Surface proteins analysis in GRID platform.
Transcriptomics and Phylogenetics Applications in GRID
• Data analysis specific for microarray and allow the GRID user to store and search this information, with direct access to the data files stored on Data Storage element on GRID servers.
• To validate an infrastructure to perform Application of Phylogenetic based on execution application of Phylogenetic methods estimates trees.
Database and Functional Genomics Applications
• To offer the possibility to manage and access biological database by using the GRID EGEE.
• To cluster gene products by their functionality as an alternative to the normally used comparison by sequence similarity.
Molecular Dynamics Applications
• To improve the scalability of Molecular Dynamics simulations.

• To perform simulation folding and aggregation of peptides and small proteins, to investigate structural properties of proteins and protein-DNA complexes and to study the effect of mutations in proteins of biomedical interest.
• To perform a challenge of the Wide In Silico Docking On Malaria.

EGEE and EGEEII future plan

BIOINFOGRID will evaluate the Grid usability in wide variety of applications, the aim to build a strong and unite BIONFOGRID Community and explore and exploit common solutions.
The BIOINFOGRID collaboration will be able to establish a very large user group in Bioinformatics in EUROPE. This cooperation will be able to promote the Bioinformatics and GRID applications in EGEE and EGEEII. The aim of the BIOINFOGRID project is to bridge the gap, letting people from the bioinformatics and life science be aware of the power of Grid computing just trying to use it. We intend to pursue this goal by using a number of key bioinformatics applications and getting them run onto the European Grid Infrastructure.
The most natural and important spin off of the BIOINFOGRID project will then be a strong dissemination action within the user's communities and across them. In fact, from one side application's experts will meet Grid experts and will learn how to re-engineer and adapt their applications to "run on the Grid"and, from the other side (and at the same time), application's experts will meet other-applications'experts with a high probability that ones'expertises can be exploited as others'solutions.
The BIOINFOGRID project will provide the EGEEII with very useful inputs and feedbacks on the goodness and efficiency of the structure deployed and on the usefulness and effectiveness of the Grid services made available at the continental scale. In fact, having several bioinformatics scientific applications using these Grid services is a key moment to stress the generality of the services themselves.

1d: Computational Chemistry - Lattice QCD - Finance / 70

Grid computation for Lattice QCD

Authors: Bruno Taglienti[1]; Giuseppe Andronico[2]; Leonardo Giusti[3]; Luigi Del Debbio[3]; Silvano Petrarca[4]

[1] *INFN Sezione di Roma1*

[2] *INFN SEZIONE DI CATANIA*

[3] *CERN*

[4] *Universita La Sapienza*

Corresponding Author: giuseppe.andronico@ct.infn.it

This is the first use of the GRID structure to an expensive QCD lattice calculation performed under the VO theophys.
It concerns the study on the lattice of the SU(3) Yang-Mills topological charge distribution, which is one of the most important non pertubative features of the theory. The first moment of the distribution is the topological susceptibility, which enters in the famous Witten Veneziano formula (See Luigi Del Debbio, Leonardo Giusti, Claudio Pica Phys.Rev.Lett.94:032003,2005 and references therein). The codes adopted in this project, are optimized to run with high efficiency on a single pc using the SSE2 feature of Intel and AMD processors to implement the performances.
(L. Giusti, C. Hoelbling, M. Luscher, H. Wittig,Comput.Phys.Commun.153:31-51,2003)
Different codes based on parallel structure are already being developed and tested. They need a band interconnection among nodes greater than 250 MBytes/s and we hope they can be sent to the GRID in

the future. The first physical results of the project are planned to be presented at Lattice2006 international symposium at the end of July in Tucson by the collaboration (L. Del Debbio (Edinburgh), L. Giusti (Cern), S. Petrarca (univ. of Roma 1), B. Taglienti (INFN, Sez. of Roma1).
The production on a "small" SU(3) lattice(12ˆ4) at beta=6.0 is finished.
The results are very encouraging.
We started a new run on a 14ˆ4 lattice whith the same physical volume. Although the statistics is yet unsufficient, the signal is confirmed.

The total CPU time used from the beginning of the work (20-10-2005) up to now (26-01-2006) under the VO theophys turns out to be 70000 hours.
Total number of job submitted is about 6500.
Failures (approximately):
500 due to non-sse2 CPU.
1000 job aborted due to unknown reasons.

A typical 12ˆ4 job requires 220 MB of ram; all the production has been divided in
small chunks requiring approximately 12 hours of CPU. (Longer jobs are prone to be aborted
by the GRID system). Every job reads and writes 5.7MB from/to a storage element.

The resouces needed by the typical 14ˆ4 job are nearly a factor of 2 for CPU, ram and storage.
We organized the production in 120 simultaneous jobs, and each job runs on a
single processor.
The job time length is chosen as a compromise between the
job time limit actually imposed by the GRID system and the bookkeeping activity needed to acquire the result and start a new job.

2c: Special type of jobs (MPI, SDJ, interactive jobs, …) - Information systems / 71

User Applications of R-GMA

Authors: Abdeslem Djaoui[1]; Alastair Duncan[1]; Antony Wilson[1]; John Walk[1]; Linda Cornwall[1]; Martin Craig[1]; Rob Byrom[1]; Robin Middleton[1]; Steve Fisher[1]; Steve Hicks[1]

[1] *RAL*

Corresponding Author: s.m.fisher@rl.ac.uk

The Relational Grid Monitoring Architecture (R-GMA) provides a uniform method to access and publish both information and monitoring data. It has been designed to be easy for individuals to publish and retrieve data. It provides information about the grid, mainly for the middleware packages, and information about grid applications for users. From a user's perspective, an R-GMA installation appears as a single virtual database. R-GMA provides a flexible infrastructure in which producers of information can be dynamically created and deleted and tables can be dynamically added and removed from a schema. All of the data that is published has a timestamp, enabling its use for monitoring. R-GMA is currently being used for job monitoring, application monitoring, network monitoring, grid FTP monitoring and the site functional tests (SFT).

R-GMA is a relational implementation of the Global Grid Forum's (GGF) Grid Monitoring Architecture (GMA). GMA defines producers and consumers of information and a registry that knows the location of all consumers and producers. R-GMA provides Consumer, Producer, Registry and Schema services.

The consumer service allows the user to issue a number of different types of query: history, latest and continuous. History queries are queries over time sequenced data and latest queries correspond to the intuitive idea of current information. For a continuous query, new data are broadcast to all subscribed consumers as soon as those data are published via a producer. Consumers are automatically matched with producers of the appropriate type that will satisfy their query.

Data published by application code is stored by a producer service. R-GMA provides a producer service that includes primary and secondary producers. Primary producers are the initial source of data within an R-GMA system. Secondary producers can be used to republish data in order to co-locate information to speed up queries (and allow multi-table queries), to reduce network traffic and to offer different producer properties. It is envisaged that there will be numerous primary producers and one or two secondary producers for each subset of data. Both primary and secondary producers may use memory or a database to store the data and may specify retention periods. Memory producers give the best performance for continuous queries, whereas database producers give the best performance where joins are required.

It is not necessary for users to know where other producers and consumers are: this is managed by the local producer and consumer services on behalf of the user. In most cases it is not even necessary to know the location of the local producer and consumer services, as worker nodes and user interface nodes are already configured to point to their local R-GMA producer and consumer services.

There are already a number of applications using R-GMA. The first example is job monitoring. There was a requirement to allow grid users to monitor the progress of their jobs and for VO administrators to get an overview of what was happening on the grid. The problems were that the location in which a grid job would end up was not known in advance, and that worker nodes were behind firewalls so they were not accessible remotely.

SA1 has adopted the job wrapper approach, as this did not require any changes to the application code. Every job is put in a wrapper that periodically publishes information about the state of the process running the job and its environment. These data are currently being published via the SA1 JobMonitoring table within R-GMA. A second application has been written to run on the resource broker nodes. This application examines the logging and bookkeeping logs and publishes data about the changes in state of grid jobs. These data are made available via the SA1 JobStatusRaw table.

Both the producer in the job wrapper and the producers on the resource broker nodes make use of R-GMA memory primary producers. A database secondary producer is used to aggregate the data.

Other uses of R-GMA include application monitoring, network monitoring and gridFTP monitoring. There are a number of different ways to implement application monitoring including the wrapper approach, as the job monitoring, and instrumentation of the application code. Instrumentation of the code can mean using a logging service, e.g. log4j, which publishes data via R-GMA, or calling R-GMA API methods directly from the application code.

The network monitoring group, NA4, have been using R-GMA to publish a number of network metrics. They used memory primary producers in the network sensors to publish the data and a database secondary producer to aggregate the data.

SA1 have made use of the consumer service for monitoring grid FTP metrics. They have written a memory primary producer that sits on the gridFTP server nodes and publishes statistics about the file transfers. A continuous consumer is used to pull in all the data to a central location, from where it is written to an Oracle database for analysis. This was used for Service Challenge 3.

Two patterns have emerged from the use made of R-GMA for monitoring. In both patterns data is initially published using memory primary producers. These may be short lived and only make the data available for a limited time, e.g. the lifetime of a grid job. In one pattern data are made persistent by using a consumer to populate an external database which applications query directly. In the other pattern, an R-GMA secondary producer is used to make the data persistent and also make it available for querying through R-GMA.

In the coming months we plan to add support for multiple Virtual Data Bases, authorization within the context of a Virtual Data Base using VOMS attributes, registry replication, load balancing over multiple R-GMA servers and support for Oracle.

R-GMA is an information and monitoring system that has been specifically designed for the grid environment. It can be used by systems, VOs and individuals and is already in use in production.

Poster and Demo session + cocktail / 72

gLite Service Discovery for users and applications

Authors: Abdeslem Djaoui[1]; Alastair Duncan[1]; Antony Wilson[1]; John Walk[1]; Linda Cornwall[1]; Martin Craig[1]; Rob Byrom[1]; Robin Middleton[1]; Steve Fisher[1]; Steve Hicks[1]

[1] *RAL*

Corresponding Author: s.m.fisher@rl.ac.uk

In order to make use of the resources of a grid, to submit a job or query information for example, a user must contact a service that provides the capability, usually via a URL. Grid services themselves must often contact other services to do their work. In order to locate services, some kind of dynamic service directory is required and there exist several grid information systems, such as R-GMA and BDII, that can provide this service. However each information system has its own unique interface, so JRA1 have developed a standard Service Discovery API to hide these differences from applications that simply want to locate services that meet their criteria.

The gLite Service Discovery API provides a standard interface to access service details published by information systems. There are four methods available for discovering services, these are: listServices, listAssociatedServices, listServicesByData and listServicesByHost. These all take a range of arguments for narrowing the search and all return a list of service structures. Once you have found a service it is then possible to use other methods to obtain more detailed information about it (using its unique id). These methods are: getService, getServiceDetails, getServiceData, getServiceDataItem, getServiceSite and getServiceWSDL.

The gLite Service Discovery API provides interfaces for the Java and C/C++ programming languages and a command line tool (glite-sd-query). It uses plugins for the R-GMA and BDII information systems, and for retrieving the information from an XML file. Other plugins (e.g. UDDI) could be developed if needed.

JRA1 also provide a service tool, rgma-servicetool, to allow any service running on a host to easily publish service data via R-GMA. All a service has to do is to provide a description file that contains static information about itself and the name of a command to call, plus any required parameters, in order to obtain the current state of the service. This information is then published via R-GMA to a number of tables that conform to the GLUE specification. The data published to these tables are used by the R-GMA gLite Service Discovery implementation. Any service, including VO services, can make use of rgma-servicetool.

The existing system assumes that the underlying information system has been correctly configured. In the case of R-GMA this means that the client needs to know the local R-GMA server (sometimes known as a "Mon box"). A user coming to an unknown environment with a laptop needs to first find the information system before interacting with it. This is the well-known bootstrapping problem that can be solved by IP multicast techniques. We will provide discovery of local services without making use of existing information systems and with near-zero configuration. Clients send a multicast query to a multicast group and services that satisfy the query respond directly to the client using unicast. This capability will initially be added to R-GMA services. Once this has been done it will be possible to introduce additional R-GMA servers at a site, for example to take increased load, without the need to reconfigure any clients. The existing SD API with the R-GMA plugin will immediately benefit from the new server. Subsequently this component, suitably packaged, will be made available to other gLite services.

The combination of the rgma-servicetool and the gLite Service Discovery makes it simple for any service to make itself known and then for user and high-level applications to find these services. In addition once the bootstrapping code is developed and added to R-GMA, the configuration of R-GMA, and thereby SD with the R-GMA plugin, will become trivial.

2d: VO tools - Portals / 73

GridICE monitoring for the EGEE infrastructure

Authors: Antonio Pierro[1]; Cristina Aiftimiei[2]; Enrico Fattibene[3]; Gennaro Tortone[4]; Giacinto Donvito[1]; Giuseppe Misurelli[3]; Guido Cuscela[1]; Natascia De Bortoli[4]; Sergio Andreozzi[3]; Sergio Fantinel[2]

[1] *INFN-Bari*

[2] *INFN-LNL*

[3] *INFN-CNAF*

[4] *INFN-Napoli*

Corresponding Author: sergio.andreozzi@egi.eu

Grid computing is concerned with the virtualization, integration and management of services and resources in a distributed, heterogeneous environment that supports collections of users and resources across traditional administrative and organizational domains.

One aspect of particular importance is Grid monitoring, that is the activity of measuring significant Grid resource-related parameters in order to analyze usage, behavior and performance of a Grid system. The monitoring activity can also help in the detection of fault situations, contract violations and user-defined events.

In the framework of the EGEE (Enabling Grid for E-sciencE) project, the Grid monitoring system called GridICE has been consolidated and extended in its functionalities in order to meet requirements from three main categories of users: Grid operators, site administrators and Virtual Organization (VO) managers. Besides the specific needs of these categories, GridICE offers a common sensing, collection and presentation framework enabling to share common features, while also offering user-specific needs.

A first common aspect to the different users is the set of measurements to be performed. Typically, there is a wide number of base measurements that are of interest for all parties, while a small number is specific to them. What makes the difference is the aggregation criteria required to present the monitoring information. This aspect is intrinsic to the multidimensional nature of monitoring data. Example of aggregation dimensions identified in GridICE are: the physical dimension referring to geographical location of resources, the Virtual Organization (VO) dimension, the time dimension and the resource identifier dimension.

As an example, considering the entity 'host' and the measure 'number of started processes in down state', the Grid operator can be interested in accessing the sum of the measurement values for all the core machines (e.g., workload manager, computing element, storage element) in the whole infrastructure, while the Virtual Organization manager can be interested in the sum of the measurement values for all the core machines that are authorized to the VO members. Finally, the site administrator can be interested in accessing the sum of the measurement values for all machines part of its site.

Another aspect that is common to all the consumers is being able to start from summary views and to drill down to details. This feature can enable to verify the composition of virtual pools or to sketch the sources of problems.

As regards the distribution of monitoring data, GridICE follows a 2-level hierarchical model: the intra-site level is within the domain of an administrative site and aims at collecting the monitoring data at a single logical repository; the inter-site level is across sites and enables the Grid-wide access to the site repository. The former is typically performed by a fabric monitoring service, while the latter is performed via the Grid Information Service. In this sense, the two levels are totally decoupled and different fabric monitoring services can be adapted to publish monitoring data to GridICE, thought the proposed default solution is the CERN Lemon tool.

Considering the sensing activity, GridICE adopts the whole set of measures defined in the GLUE Schema 1.2, further it provides extensions to cover new requirements. The extensions include a more complete host-level characterization, Grid jobs related attributes and summary info for batch systems (e.g., number of total slots, number of worker nodes that are down).

The development activity in the EGEE project has focused on the following aspects: the redesign of the presentation level took into consideration the usability principles and compliance with W3C standards; sensors for measuring parameters related to Grid job have been re-engineered to scale to the number of jobs envisioned by big sites (e.g., LCG Tier 1 centers); new sensors have been written to deal with summary information for computing farms; stability and reliability of both server and sensors.

The deployment activity covers the whole EGEE framework with several server instances supporting the work of different Grid sub-domains (e.g., whole EGEE Grid domain, ROC domain, national domain). Other Grid projects have adopted GridICE for monitoring their resources (e.g., EUMedGrid, EUChinaGRID, EELA).

As regards the user experience, GridICE has proven to be useful to different users in different ways. For instance, Grid operators have summary views for aspects such as information sources status and host status. Site administrators appreciate the job monitoring capability showing the status and computing activity of the jobs accepted in the managed resources. VO managers use GridICE to verify the available resources and their status before to start the submission of a huge number of jobs. Finally, GridICE has been positively adopted in dissemination activities.

While GridICE has reached a good maturity level in the EGEE project, many challenges are still open in the dynamic area of Grid systems. The short term plans are: (1) as regards the discovery process, there is the need to finalize the transition from the MDS-based information service to the gLite service discovery plus publisher services such as R-GMA producers and CEMon; (2) integration with information present in the Grid Operation Center (GOC) database for accessing resource planned downtime and other management information; (3) tailored sensors for the workload management service; (4) sensors for measuring data transfer activities across Grid sites.

References:

Dissemination website: http://grid.infn.it/gridice

Publications:
http://grid.infn.it/gridice/index.php/Research/Publications

Summary:

C. Aiftimiei, S. Andreozzi, G. Cuscela, N. De Bortoli, G. Donvito, S. Fantinel, E. Fattibene, G. Misurelli, A. Pierro, G.L. Rubini, G.Tortone.

Poster and Demo session + cocktail / 74

Migrating Desktop - graphical front-end to grid - On-line Demonstration

Author: Marcin Plociennik[1]

Co-authors: David Rodriguez [2]; Harald Kornmayer [3]; Jesus Marco [2]; Norbert Meyer [1]; Pawel Wolniewicz [1]

[1] *PSNC*

[2] *IFCA*

[3] *FZK*

Corresponding Author: pawelw@man.poznan.pl

Demo description:

Demo will show following features and functionality:
- graphical user environment for job submission, monitoring and other grid operations
- running applications from different disciplines and communities
- running within MD platform batch and MPI applications
- running sequential and interactive applications

Two applications had been selected to present MD framework and mentioned above features: parallel ANN training application, MAGIC Monte Carlo Simulation

Parallel ANN training application - Interactive application from CrossGrid –(description of usecase in technical background section)
This application is used to train an Artificial Neural Network (ANN) using simulated data for the DELPHI experiment. The ANN is trained to distinguish between signal (Higgs bosson) events and background event (in the demo the background used

includes WW and QCD events). The evolution of the training can be monitored using
the
MD with a graphics presenting curent error, and 4 small graphics that show the ANN
value vs. an event variable (that can be selected by the user). The application is
compiled with MPICH-P4 for intracluster use and with MPICH-G2 for intercluster use.
This application uses the interactive input channel to let the user make a clean
stop
of the training (instead of killing the job), and also the possibility of resetting
the ANN weights to random values, to avoid local minima.

MAGIC Monte Carlo Simulation
The MAGIC Monte Carlos Simulation (MMCS) is one of the generic applications within
EGEE. As the simulation of extensive air showers initiated by high
energetic cosmic rays is very compute intensive, the MAGIC collaboration –together
with Grid resource centers from the EGEE project - migrate the MMCS application
within the last years to the EGEE infrastructure to speed up the production of the
simulations. To get enough statistics for a physics analysis, many jobs with the
same
input parameters but different random numbers needs to be submitted. The submission
tools from the MAGIC Grid are integrated in the Migrating Desktop and its underlying
infrastructure. Therefore all services und features of the Migrating Desktop like
Job
Monitoring, Data management, etc. can be used by members of the MAGIC virtual
organization.

Platform and services
Testbed:
- EGEE production, GILDA and CrossGrid testbed
Applications:
- MAGIC application running on EGEE,
- ANN interactive application running on CrossGrid testbed

Services:
- usage of following EGEE services:
- WMS: RB, LB, CE
- Data Management: SE, LCG-UTILS (Replica Manager)
- Information Index
- usage of following CrossGrid testbed services
- WMS: RB, LB, CE
- Data Management: SE, LCG-UTILS (Replica Manager)
- Information Index

Technical background:

A number of Grid middleware projects are working on user interfaces for interaction
with grid applications, however due to the dynamic and complex nature of the Grid,
it's not easy to attract new users like ordinary scientists. To solve this problem
we
introduce the concept of Migrating Desktop which is a graphical, user oriented tool
that simplifies the use of the grid technology in the application area.
The Migrating Desktop (MD)is an advanced graphical user interface and a set of tools
combined with user-friendly outlook, similar to window based operating systems. It
hides the complexity of the grid middleware and allows to access grid resources in
an
easy and transparent way with special focus on interactive and parallel grid
applications. These applications are both compute- and data-intensive and are
characterised by the interaction with a person in a processing loop. MD can attract
new users by its features: easy to use, platform independed, available everywhere,
enables possibility to easily add new application that can be batch or interactive,
sequential or parallel. Thanks to its open architecture it can easily integrate
existing or incoming tools that for example supports grid operations or enables
collaborative work.

This research refers to three different grid projects: EU BalticGrid project, EU CrossGrid project, and Progress (co-founded by Sun Microsystems and the Polish State Committee for Scientific Research). As a key product of CrossGrid project, Migrating Desktop has proved its usefulness in everyday work of users community.

Technical background
Platform overview
The aim of the Migrating Desktop is to provide scientists with a framework which hides the details of most Grid services and allows working with grid application in an easy and transparent way. The graphical user interface integrates and makes use of
number of middleware and integrates the individual tools into a single product providing a complete grid front-end. It is built on base of a mechanism for discovering, integrating, and running modules called bundles based on the OSGi specification. When the MD is launched, the users can work with environment composed of the set of bundles. Usually a small tool is written as a single bundle, whereas a complex tool has its functionality split across several bundles. A bundle is the smallest unit of our platform that can be developed and delivered separately. Such approach allows increasing functionality in an easy way without the need of architecture changes.
The Migrating Desktop framework allows the user to access transparently the Grid resources, run sequential or interactive, batch or MPI applications, monitoring and visualization, and manage data files. MD provides a front-end framework for embedding
some of the application mechanisms and interfaces, and allows the user to have virtual access to Grid resources from other computational nodes.
The MD is a front end to The Roaming Access Server (RAS), which intermediates to communication with different grid middleware and applications. The Roaming Access Server offers a well-defined set of web-services that can be used as an interface for
accessing HPC systems and services (based on various technologies) in a common and standardised way. All communication bases on web services technology.
Our platform can work with different grid testbeds: based on LCG 2.3/2.4, LCG 2.6, Progress 1.0. Due to its open system nature it can be easily ported to support other testbeds.

Applications use cases

MAGIC Monte Carlo Simulation
The MAGIC Monte Carlos Simulation (MMCS) is one of the generic applications within EGEE. As the simulation of extensive air showers initiated by high energetic cosmic rays is very compute intensive, the MAGIC collaboration–together with Grid resource centers from the EGEE project - migrate the MMCS application within the
last years to the EGEE infrastructure to speed up the production of the simulations. The simulation of the air showers requires the most computing time, e.g. a request for a Monte Carlo sample of 1.0 million gamma-events would need around 1500 computing
hours on a standard CPU (2˜MHz PentiumIV). This can be speeded up by using many resources by parallelizing the application, if possible. Therefore the simulation of a Monte Carlo sample is split in subjobs of 1000 events to run in parallel on distributed Grid resources.The resulting 1000 data files are transferred and stored on a dedicated Grid storage center automatically when a subjob is finished. When all files are available, a program merges them to one single file that is processed by the next program of the Monte Carlo workflow.

To track and manage the big number of jobs, a meta database containing information about single jobs, their status and available data was set up. The metadata are stored in a separate relational database combining information from the Grid domain with data needed by MAGIC scientists. A Grid user requests a given number of Monte Carlo events by writing this into the meta database, while a daemon process regularly submits smaller bunches of subjobs to the Grid resources. The current implementation of the MMCS system does not require any additional software installation on Grid resources.

The submission tools from the MAGIC Grid are integrated in the Migrating Desktop and its underlying infrastructure. Therefore all services und features of the Migrating Desktop like Job Monitoring, Data management, etc. can be used by members of the MAGIC virtual organization.

Interactive Application (CrossGrid) –Parallel ANN training application.
The user launches the ANN job wizard from the MD Job Wizard menu or from an already existing job shortcut. After filling all the necessary parameters in Job Wizard the user submits the job. Once it is running the ANN plugin can be launched. In the plugin the user can see a panel with four graphics representing the value of the ANN for a subset of the training events (signal events in green and background events in red) vs. several variables of the events. The user can change the selected variables using the combo list at the bottom of the plugin window. At the right side the user can see the graphic representing the evolution of the ANN training error vs the training epoch. The plugin also includes three options: "reset weights" that resets the values of the ANN weights to random, "Stop application" - the program goes out of the training loop stopping the training and "Exit" for closing the plugin window. The user after the error is more or less in a plateau should press the "Reset weights" button and observe the error evolution. Afterwards, if necessary to finish the demo the user can press the "Stop Application" button.

Used technology
The Migrating Desktop bases on the Java applet technology. It can be launched using the Java Webstart technology or using a web browser with the appropriate Java Plug-in
included in the Java Runtime Environment (JRE). We are basing on Swing libraries for
designing graphical user interface, the Java CoG Kit version 1.2 is being used as an interface to Globus (for operation on proxy and GridFTP/FTP) functionality and Axis ver.1.1 web services client for communication with the Roaming Access Server. Migrating Desktop follows OSGi Service Platform specification version 4 (August 2005) and is based on the same plugin engine as Eclipse platform. Currently RAS for cooperation with EGEE infrastructure is using LCG2.6 platform but it is foreseen to move to gLite.

Summary:

Demo description:

Demo will show following features and functionality:
- graphical user environment for job submission, monitoring and other grid operations
- running applications from different disciplines and communities
- running within MD platform batch and MPI applications
- running sequential and interactive applications

Two applications had been selected to present MD framework and mentioned above features: parallel ANN training application, MAGIC Monte Carlo Simulation

Parallel ANN training application - Interactive application from CrossGrid –(description of usecase in technical background section)
This application is used to train an Artificial Neural Network (ANN) using simulated data for the DELPHI experiment. The ANN is trained to distinguish between signal (Higgs bosson) events and background event (in the demo the background used includes WW and QCD events). The evolution of the training can be monitored using the
MD with a graphics presenting curent error, and 4 small graphics that show the ANN value vs. an event variable (that can be selected by the user). The application is compiled with MPICH-P4 for intracluster use and with MPICH-G2 for intercluster use. This application uses the interactive input channel to let the user make a clean

stop
of the training (instead of killing the job), and also the possibility of resetting
the ANN weights to random values, to avoid local minima.

MAGIC Monte Carlo Simulation
The MAGIC Monte Carlos Simulation (MMCS) is one of the generic applications within EGEE. As the simulation of extensive air showers initiated by high
energetic cosmic rays is very compute intensive, the MAGIC collaboration –together with Grid resource centers from the EGEE project - migrate the MMCS application within the last years to the EGEE infrastructure to speed up the production of the simulations. To get enough statistics for a physics analysis, many jobs with the
same
input parameters but different random numbers needs to be submitted. The submission tools from the MAGIC Grid are integrated in the Migrating Desktop and its underlying infrastructure. Therefore all services und features of the Migrating Desktop like
Job
Monitoring, Data management, etc. can be used by members of the MAGIC virtual organization.

Platform and services
Testbed:
- EGEE production, GILDA and CrossGrid testbed
Applications:
- MAGIC application running on EGEE,
- ANN interactive application running on CrossGrid testbed

Services:
- usage of following EGEE services:
- WMS: RB, LB, CE
- Data Management: SE, LCG-UTILS (Replica Manager)
- Information Index
- usage of following CrossGrid testbed services
- WMS: RB, LB, CE
- Data Management: SE, LCG-UTILS (Replica Manager)
- Information Index

1d: Computational Chemistry - Lattice QCD - Finance / 75

SALUTE –GRID Application for problems in quantum transport

Author: Emanouil Atanassov[1]

Co-authors: Aneta Karaivanova [1]; Mihail Nedjalkov [1]; Todor Gurov [1]

[1] *IPP-BAS*

Corresponding Author: emanouil@parallel.bas.bg

Authors: E. Atanassov, T. Gurov, A. Karaivanova and M. Nedjalkov
Department of Parallel Algorithms
Institute for Parallel Processing - Bulgarian Academy of Sciences
E-mails:{emanouil, gurov, anet, mixi}@parallel.bas.bg

Abstract body:
SALUTE (Stochastic ALgorithms for Ultra-fast Transport in sEmiconductors) is an MPI Grid application developed for solving computationally intensive problems in quantum transport.

Monte Carlo (MC) methods for quantum transport in semiconductors and semiconductor devices have been actively developed during the last decade. If temporal or spatial scales become short, the evolution of the semiconductor carriers cannot be described in terms of the Boltzmann transport [1] and therefore a quantum description is needed. We note the importance of active investigations in this field: nowadays nanotechnology provides devices and structures where the carrier transport occurs at nanometer and femtosecond scales. As a rule quantum problems are very computationally intensive and require parallel and Grid implementations.

SALUTE is a pilot grid application developed at the Department of Parallel Algorithms, Institute for Parallel Processing - BAS where the stochastic approach relies on the numerical MC theory applied to the integral form of the generalized electron-phonon Wigner equation. The Wigner equation for the nanometer and femtosecond transport regime is derived from a three equations set model based on the generalized Wigner function [2]. The full version of the equation poses serious numerical challenges. Two major formulations (for homogeneous and inhomogeneous cases) of the equation are studied using SALUTE.

The physical model in the first formulation describes a femtosecond relaxation process of optically excited electrons which interact with phonons in one-band semiconductor [3]. The interaction with phonons is switched on after a laser pulse creates an initial electron distribution. Experimentally, such processes can be investigated by using ultra-fast spectroscopy, where the relaxation of electrons is explored during the first hundreds femtoseconds after the optical excitation. In our model we consider a low-density regime, where the interaction with phonons dominates the carrier-carrier interaction. In the second formulation we consider a highly non-equilibrium electron distribution which propagates in a quantum semiconductor wire [4]. The electrons, which can be initially injected or optically generated in the wire, begin to interact with three dimensional phonons. The evolution of such process is quantum, both, in the real space due to the confinements of the wire, and in the momentum space due to the early stage of the electron-phonon kinetics. A detailed description of the algorithms can be found in [5, 6, 7].

Monte Carlo applications are widely perceived as computationally intensive but naturally parallel. The subsequent growth of computer power, especially that of the parallel computers and distributed systems, made possible the development of distributed MC applications performing more and more ambitious calculations. Compared to the parallel computing environment, a large-scale distributed computing environment or a Computational Grid has tremendous amount of computational power. Let us mention the EGEE Grid which today consists of over 18900 CPU in 200 Grid sites.

SALUTE solves an NP-hard problem concerning the evolution time. On the other hand, SALUTE consists of Monte Carlo algorithms which are inherently parallel. Thus, SALUTE is a very good candidate for implementations on MPI-enabled Grid sites. By using the Grid environment provided by the EGEE project middleware, we were able to reduce the computing time of Monte Carlo simulations of ultra-fast carrier transport in semiconductors. The simulations are parallelized on the Grid by splitting the underlying random number sequences.

Successful tests of the application were performed at several Bulgarian and South East European EGEE GRID sites using the Resource Broker at IPP-BAS. The MPI version was MPICH 1.2.6, and the execution was performed on clusters using both pbs and lcgpbs jobmanagers, i.e. with shared or non-shared home directories. The test results show excellent parallel efficiency. Obtaining results for larger evolution times requires more computational power, which means that the application should run on larger sites or on several sites in parallel. The application can provide results for other types of semiconductors like Si or for composite materials.

Figure 1. Distribution of optically generated electrons in a quantum wire.

REFERENCES

[1] J. Rammer, Quantum transport theory of electrons in solids: A single-particle approach, Reviews of Modern Physics, series 63 no 4, 781 - 817, 1991.
[2] M. Nedjalkov, R. Kosik, H. Kosina, and S. Selberherr, A Wigner Equation for Nanometer and Femtosecond Transport Regime, In: Proceedings of the 2001 First IEEE Conference on Nanotechnology, (October, Maui, Hawaii), IEEE, 277-281, 2001.
[3] T.V. Gurov, P.A. Whitlock, "An efficient backward Monte Carlo estimator for solving of a quantum kinetic equation with memory kernel", Mathematics and Computers in Simulation, Vol. 60, 85-105, 2002.
[4] M. Nedjalkov, T. Gurov, H. Kosina, D. Vasileska. and V. Palankovski, Femtosecond Evolution of Spatially Inhomogeneous Carrier Excitations: Part I: Kinetic Approach, to appear in Lecture Notes in Computing Sciences, Springer-Verlag Berlin Heidelberg, Vol. 3743, (2006)
[5] E. Atanassov, T. Gurov, A. Karaivanova, and M. Nedjalkov, SALUTE –an MPI GRID Application, in: Proceedings of the 28th International Convetion, MIPRO 2005, May 30-June 3, Opatija, Croatia, 259 - 262, 2005.
[6] T.V. Gurov, M. Nedjalkov, P.A. Whitlock, H. Kosina and S. Selberherr, Femtosecond relaxation of hot electrons by phonon emission in presence of electric field, Physica B, vol 314, p. 301, 2002
[7] T.V. Gurov and I.T. Dimov, A Parallel Monte Carlo Method for Electron Quantum Kinetic Equation, LNCS, Vol. 2907, Springer-Verlag, 153—160, 2004

Summary:

A Grid application developed for electron transport problems called SALUTE (Stochastic ALgorithms for Ultra-fast Transport in sEmiconductors) is presented. Monte Carlo algorithms for solving quantum-kinetic equations describing a bunch of problems are developed. A physical model of a femtosecond relaxation of optically excited electrons which interact with phonons in an one-band semicondoctor is considered. The electron-phonon interaction is switched on after a laser pulse creates an initial electron distribution. Two cases of this process are investigated - with and without an applied electric field.

We describe Grid implementation of the developed algorithms which are CPU-intensive. Using this application innovative results for different materials can be obtained. Here we present the first version of SALUTE which is used to obtain innovative results for GaAs materials. The results from a number of tests on MPI-enabled Grids are shown and disscussed.

2c: Special type of jobs (MPI, SDJ, interactive jobs, ...) - Information systems / 76

Real time computing for financial applications

Author: Stefano Cozzini[1]

Co-author: Riccardo di Meo [2]

[1] *CNR-INFM Democritos and ICTP*

[2] *Egrid project ICTP*

Corresponding Author: cozzini@democritos.it

Computing grids are quite attractive for large scale financial applications: this is especially evident in the segment of dynamic financial services, where applications must complete complex tasks within strict deadlines. The traditional response has been to over-provision for making sure there is plenty of 'headroom' in resource availability, thereby maintaining large computational resources booked and unused with a great cost in terms of infrastructure. Moreover nowadays some of

these complex tasks need an amount of computing power that is unfeasible to keep in house.
Computing grids can deliver the amounts of power needed in such a scenario, but there are still large limitations to overcome. In this brief report we address the solution we developed to provide real time computing power through the EGRID facility for a test case financial application.
The test case we consider is an application that estimates the sensitivities of a set of stocks
to specific risk factors: technical details about the procedure can be found elsewhere; we will present here only the computational details of the application to better define the problem we faced, and the solutions adopted for porting it to the grid.

We implemented different technical solutions for our application in a sort of trial and error fashion. We will present briefly all of the attempts.

All implemented solutions rely on a "job reservation mechanism": we allocate grid resources in advance to eliminate latency due to the job submission mechanism. In this way, as soon as we get enough resources allocated we can interact with them in real time.
The drawback is that being an advanced booking strategy, for "best effort"services this approach could be unfeasible. It is not the case for this experimental work though, but the limitation should be taken into account when approaching production runs.
The booking mechanism has been implemented in the following way. An early submission of a bunch of jobs is run for securing the availability of WN at a given time.
Each pooled node will executes a program that regularly checks a host (usually the UI, but not necessarily). The contacted host enrolls this WN for the user's program, as soon as the user executes that program. When the execution terminates the results are available in real time without any delay introduced by WMS of the grid. The WNs remain booked, and so are ready to be enrolled again for other program executions; eventually they are freed by the user.
This approach, where the WN asks to be enrolled in a computation thereby acting as a client, is needed because the WN cannot be reached directly from the UI.

1d: Computational Chemistry - Lattice QCD - Finance / **77**

The EGRID facility

Authors: Alessio Terpin[1]; Angelo Leto[1]; Antonio Messina[1]; Ezio Corso[1]; Riccardo Murri[1]; Riccardo di Meo[1]; Stefano Cozzini[2]

[1] *Egrid project ICTP*

[2] *CNR-INFM Democritos and ICTP*

Corresponding Author: cozzini@democritos.it

The EGRID project aims at implementing a national Italian facility for processing economic and financial data using computational grid technology. As such, it acts as the underlying fabric on top of which partner projects, more strictly focused on research in itself, develop end-user applications.
The first version of the EGRID infrastructure has been in operation since October 2004. It is based on European Data-Grid (EDG) and the Large Hadron Collider Computing Grid (LCG) middleware, and it is hosted as an independent Virtual Organization (VO) within INFN's grid.IT. Several temporary workarounds were implemented mainly to tackle privacy and security issues on data management: in these last few months the infrastructure was fully re-designed
to better address them. The redesigned infrastructure makes use of several new tools: some are part of EDG/LCG/EGEE middleware, while some others were developed

independently within EGRID. Moreover the EGRID project joined recently EGEE as pilot application in the field of finance, which means that the EGRID VO will be soon recognized on the full EGEE computational grid; this may impose some compatibility constraints because of the afore mentioned additions we make, which we will handle when the time comes.

The new infrastructure will be composed of various architectural layers that will take care of different aspacts.

Security issue has been handled at the low middleware level that manages data: an implementation of the SRM (Storage Resource Manager) protocol is being completed where novel ideas have been applied, thereby breaking free from the limitations of current approaches. Indeed, the SRM standard is becoming widely used as a storage access interface and, hopefully, it will soon be available on the full EGEE infrastructure. The EGRID technical staff has an on-going long time collaboration with INFN/CNAF on the StoRM SRM server, with the intention to use this software for providing the kind of fine grained access control that the project demands.
What StoRM does is to add appropriate permissions (using POSIX ACLs) to a file being requested by a user, and to remove them when the client is done with the file. Since permissions are granted on-the-fly, grid users can be mapped into pool accounts, and no special permission sets need to be enforced prior to grid usage.
An important role is given to a secure web service (ECAR) built by EGRID to act as a bridge between the (resource-level) StoRM SRM server, and the (grid-level) central LFC logical filename catalog from EGEE that replaces the old RLS of EDG.
The LFC natively implements POSIX-like ACLs on the logical file names; the StoRM server can thus read (via ECAR) the ACLs on the logical filename corresponding to a given physical file and grant or deny access to the local files, depending on the permissions on the LFC. This provides users with a consistent view of the files in grid storage.

At a higher level, in order to make even more transparent the usage of data in the grid, we also developed ELFI that allows grid resources to be accessed through the usual POSIX I/O interface. Since ELFI is a FUSE file-system implementation, grid resources are seen through a local mount-point so all the existing tools for managing the file-system automatically apply: the classical command line, any graphical user interface such as Konqueror, etc. Programs too will only have to be interfaced with POSIX, thereby aiding in grid prototyping/porting of applications.
ELFI will be installed on all WN of the farm, so applications will no longer need to explicitly run file transfer commands but simply access them directly as though they were local. Moreover, ELFI will be able to fully communicate with StoRM, and it will be installed in the host where the portal resides thereby easing portal integration of SRM resources.

The new EGRID infrastructure can be accessed via a web portal, one of the most effective ways to provide an easy-to-use interface to a larger community of users: the portal will become the main interface for naive users.
The EGRID portal that is currently under development is based on P-grade, and inherits all the features already available there: still some parts must be enhanced to comply with our requirements. The P-grade technology was chosen because it seemed sufficiently sophisticated and mature to meet our needs.
Howevever there are still missing functionalities important to EGRID.We are currently collaborating with the P-grade team in order to develop and integrate what we need:

Improved proxy management

Currently private key of the user must go through the portal, and then into the MyProxy server; we feel that for EGRID it should instead be uploaded directly from the client machine without passing through the server: this is needed to decrease security risks. To accomplish it we implemented a Java WebStart application which carries out the direct uploading. The application is seamlessly integrated into P-grade, through the standard “upload” button of the “certificates” portlet.

Data management portlet that uses ELFI

Currently P-grade does not support the SRM protocol and does not support browsing of files present in the machine hosting the portal itself. Since ELFI is our choice for accessing grid disk resources in general, including those managed through StoRM, a specific Portlet was written to browse and manipulate the file system present in the portal server itself. In fact ELFI allows grid resources to be seen as a local mount point as already mentioned it becomes easier to modify the portal for local operations rather than for some other grid service.
The Portlet allows manual transfer of files between different directories of the portal host, but since some of these directories are ELFI mount points then automatically a grid operation takes place behind the scenes. So what happens is a file movement between the portal server, remote storage and computing elements.

File management and job submission interaction

A new file management mechanism is needed besides that currently supporting "local" and "remote" files: similarly to the previous point what is required is "local on the portal server", since the portal host will have ELFI mount points allowing different grid resources to be seen as local to the portal host. In this way the workflow manager will be able to read/write input and output data through the SRM protocol.
Moreover, EGRID also needs a special version of job submission closely related to workflow jobs: what we call swarm jobs. These jobs are such that the application remains the same while the input data changes parametrically over several possible values; then a final job collects all results and makes some aggregate computation on them. At the moment the specification of each input parameter is done manually: an automatic mechanism is required.

2d: VO tools - Portals / 78

Methodology for Virtual Organization Design and Management

Authors: Jacek Kitowski[1]; Lukasz Skital[1]; Renata Slota[2]

[1] *ACC Cyfronet AGH / University of Science and Technology*

[2] *University of Science and Technology*

Corresponding Author: l.skital@cyf-kr.edu.pl

Introduction

Contemporary grid environment achieved high level of maturity. With still increasing number of various available resources, their optimal exploitation becomes a significant problem. One of solutions to the problem are Virtual Organizations (VO), which groups users and resources to solve a particular problem or a set of problems. Each problem has its own specific requirements in name of computational power, network bandwidth, storage capacity, resource availability etc. During VO design process, appropriate resources have to be selected from all available. This task can be vary difficult or time consuming, if done manually.

Current EGEE middleware (lcg 2.6 or glite 1.4.1) with VOMS or VOMRS systems address the problem of users management in existing VOs, offering web based interfaces for user registration and membership administration. However, creation of new VO is a heavy weight task, which is not automated. Existing EGEE procedures covers very well all administrative aspects, but in current form they are not feasible for automation of the VO creation task. There is no tool, which support design of new VO in EGEE environment.

In the presentation we propose a methodology of VO design. This methodology can be used to build a knowledge based system, which would support the process of VO creation by automating tasks, which do not need user interaction and support user, when the interaction is necessary. The methodology is general and can be adapted to EGEE grid environment. The knowledge based system can be used to support design of new VO without changing existing EGEE procedures.

Methodology

We propose the way of VO design which consists of three steps: definition of the VO, creation of abstract VO, creation of solid VO.

The first step of VO design is definition of the VO purpose with all requirements and constraints. This step has to be performed by an expert who knows the problem for which the VO is created. The definition of VO should be written in a form, which can be easily processed by machine, therefore we propose to use ontology for this task. The expert from the VO domain, does not have to be familiar with any ontology language. There is a need for a tool which will allow VO definition by fulfilling forms and questions. This tool can support the expert in the task, by providing hints and possible answers to questions.

The next step is creation of abstract VO. Abstract VO consists of resource types and their amount which is needed to fulfill VO requirements. Abstract VO is derived from VO definition (and available resources). Abstract VO has exact information about required computational resources, storage resources and all other specific resources, like data sources (e.g. physical experiment), but does not aim to any specific instance of resource (site). However, the expert can state, that a specific site is required in VO, and this requirement will be fulfilled in the next step - creation of solid VO. For each resource type, there are functional and not functional requirements. The functional requirements are for example installed specific software on computational resources. Non functional requirements can be availability of resource or cost of usage.

The last step of VO design is creation of solid VO. During this step abstract resources are exchanged by real instances. This task can be performed automatically. Resources selection is based on specified requirements and knowledge about the grid environment. The knowledge consists of many kinds of facts and information about each resource, like computational power, storage capacity, bandwidth (network, storage), statistics about resource availability, etc. Because of a dynamic nature of the Grid, available resources can change in time. To support VO requirements, unavailable resources should be replaced with new ones during the VO lifetime. Therefore the last step of VO design should be repeated any time when needed.

During the first step of design, apart form getting the information on needed resources, a workflow, which defines the problem would be created. The workflow visualizes a process of VO usage, from data gathering, through each necessary step, like preprocessing, computations, postprocessing and visualisation. Using the workflow, one can easily generate a specific job description (can take advantage of DAG jobs) to solve the problem. This step can be done automatically.

Summary

Optimal resource utilization is a very important task for contemporary grid environments. With grid environments growth in size and complexity, this task becomes more and more complicated. We proposed the methodology, which can positively influence the process of optimal resource utilization by supporting design of a VO. Well designed VO hides size and complexity of the grid environments, reveling only parts, which are important for the specific problem (for which VO was created). Selection of appropriate resources for VO is time consuming task, therefore it's automation can significantly improve process of VO establishment.

References
[1] EGEE Home page http://public.eu-egee.org/
[2] EGEE NA4 Home page http://egee-na4.ct.infn.it/egaap/
[3] InteliGrid http://www.inteligrid.com/
[4] KWf-Grid http://www.kwfgrid.net/main.asp

Summary:

Optimal resource utilization is a very important task for contemporary grid environments. With grid environments growth in size and complexity, this task becomes more and more complicated. We proposed the methodology, which can positively influence the process of optimal resource utilization by supporting design of a VO. Well designed VO hides size and complexity of the grid environments, reveling only parts, which are important for the specific problem (for which VO was created). Selection of appropriate resources for VO is time consuming task, therefore it's automation can significantly improve process of VO establishment.

1a: Life Sciences / **79**

In silico docking on EGEE infrastructure: the case of WISDOM

Author: Nicolas Jacq[1]

Co-authors: Astrid Maaß [2]; Florence Jacq [1]; Horst Schwichtenberg [2]; Jean Salzemann [1]; Kasam Vinod-Kusam [2]; Mahendrakar Sridhar [2]; Marc Zimmermann [2]; Martin Hofmann [2]; Matthieu Reichstadt [1]; Vincent Breton [1]; Yannick Legré [1]

[1] *CNRS/IN2P3*

[2] *Fraunhofer SCAI*

Corresponding Author: jacq@clermont.in2p3.fr

Advance in combinatorial chemistry has paved the way for synthesizing large numbers of diverse chemical compounds. Thus there are millions of chemical compounds available in the laboratories, but it is nearly impossible and very expensive to screen such a high number of compounds in the experimental laboratories by high throughput screening (HTS). Besides the high costs, the hit rate in HTS is quite low, about 10 to 100 per 100,000 compounds when screened on targets such as enzymes. An alternative is high throughput virtual screening by molecular docking, a technique which can screen millions of compounds rapidly, reliably and cost effectively. Screening millions of chemical compounds in silico is a complex process. Screening each compound, depending on structural complexity, can take from a few minutes to hours on a standard PC, which means screening all compounds in a single database can take years. Computation time can be reduced very significantly with a large grid gathering thousands of computers.
WISDOM (World-wide In Silico Docking On Malaria) is an European initiative to enable the in silico drug discovery pipeline on a grid infrastructure. Initiated and implemented by Fraunhofer Institute for Algorithms and Scientific Computing (SCAI) in Germany and the Corpuscular Physics Laboratory (CNRS/IN2P3) of Clermont-Ferrand in France, WISDOM has deployed a large scale docking experiment on the EGEE infrastructure. Three goals motivated this first experiment. The biological goal was to propose new inhibitors for a family of proteins produced by Plasmodium falciparum. The biomedical informatics goal was the deployment of in silico virtual docking on a grid infrastructure. The grid goal is the deployment of a CPU

consuming application generating large data flows to test the grid operation and services. Relevant information can be found on http://wisdom.eu-egee.fr and http://public.eu-egee.org/files/battles-malaria-grid-wisdom.pdf.

With the help of the grid, large scale in silico experimentation is possible. Large resources are needed in order to test in a transparent way a family of targets, a large enough amount of possible drug candidates and different virtual screening tools with different parameter / scoring settings. The grid added value lies not only in the computing resources made available, but also already in the permanent storage of the data with a transparent and secure access. Reliable Workload Manager System, Information Service and Data Management Services are absolutely necessary for a large scale process. Accounting, security and license management services are also essential to impact the pharmaceutical community. In a close future, we expect improved data management middleware services to allow automatic update of compound database and the design of a grid knowledge space where biologists can analyze output data.
Finally key issues to promote the grid in the pharmaceutical community include cost and time reduction in a drug discovery development, security and data protection, fault tolerant and robust services and infrastructure, and transparent and easy use of the interfaces.

The first biomedical data challenge ran on the EGEE grid production service from 11 July 2005 until 19 August 2005. The challenge saw over 46 million docked ligands, the equivalent of 80 years on a single PC, in about 6 weeks. Usually in silico docking is carried out on classical computer clusters resulting in around 100,000 docked ligands. This type of scientific challenge would not be possible without the grid infrastructure - 1700 computers were simultaneously used in 15 countries around the world. The WISDOM data challenge demonstrated how grid computing can help drug discovery research by speeding up the whole process and reduce the cost to develop new drugs to treat diseases such as malaria. The sheer amount of data generated indicates the potential benefits of grid computing for drug discovery and indeed, other life science applications. Commercial software with a server license was successfully deployed on more than 1000 machines in the same time.
First docking results show that 10% of the compounds of the database studied may be hits. Top scoring compounds possess basic chemical groups like thiourea, guanidino, amino-acrolein core structure. Identified compounds are non peptidic and low molecular weight compounds.
Future plans for the WISDOM initiative is first to process the hits again with molecular dynamics simulations. A WISDOM demonstration will be conceived at the aim to show the submission of docking jobs on the grid at a large scale. A second data challenge planned for the fall of 2006 is also under preparation to improve the quality of service and the quality of usage of the data challenge process on gLite.

2d: VO tools - Portals / 80

EnginFrame as FrameWork for Grid Enabled Web Portals on industrial and research contexts.

Authors: Alberto Falzone[1]; Andrea Rodolico[1]

[1] *NICE srl*

Corresponding Author: alberto.falzone@nice-italy.com

EnginFrame is a Web-based innovative technology, by the Italian company Nice S.r.l., that enables access and exploitation of Grid-enabled applications and infrastructures. It allows organizations to provide application oriented computing and data services to both users (via Web browsers) and in-house or ISV applications (via SOAP/WSDL based Web services), hiding all the complexity of the underlying Grid infrastructure.

In particular, EnginFrame greatly simplifies the development of Web portals exposing computing services that can run on a broad range of different computational Grid systems (including Platform LSF, Sun Grid Engine, Altair PBS, Globus, LCG-2 and gLite grid middlewares by European project EGEE).
EnginFrame supports several open and vendor neutral standards and seamlessly integrates with JSR168 compliant enterprise portals, distributed file systems, GUI virtualization tools and different kinds of authentication systems (including Globus GSI, MyProxy and a wide range of enterprise solutions).
Because EnginFrame greatly simplifies the use of Grid-enabled applications and services, it has already been adopted by numerous important industrial companies all over the world, besides many leading research & educational institutes.

Service publishing is achieved by developing simple XML-based descriptions of the interface and business logic representing the actual services implementation.
EnginFrame receives incoming requests via standard Web protocols over HTTP, authenticates and authorizes the requests and then executes the required actions into the underlying Grid computational environment.
Then, EnginFrame gathers the results and transforms them into a suitable format before sending the response to the client. Transformation of results is performed according to the nature of the client: HTML for Web browsers and XML for Web services client applications or RSS clients.
For each submitted service, a data staging area (the “spooler”) for the service input and output files is created on the file system.

Most of the information managed by EnginFrame are described by dynamically generated XML documents.
The source of such information is typically the service execution environment: an XML abstraction layer aims to submit service actions and translate raw results coming from the computational environment into XML structures.
The XML abstraction layer is designed to decouple EnginFrame from the actual Grid working environment, hiding the specific Grid technology solution. This characteristic makes possible to easily extend EnginFrame functionalities by developing ad-hoc plugins for specific computational and data Grid middlewares.
To support the integration of data Grid middleware solutions, EnginFrame introduces the concept of Virtual Spoolers that represent distributed data areas that reside outside the EnginFrame spoolers file system, but that can be remotely accessed by EnginFrame itself through the targeted data Grid technology. The structure and the content of a Virtual Spooler is described by a dynamically generated XML document.
Thus, the access to data catalogs and storage technologies is provided in a very easy way and their contents can be inspected like a “browse a file”.

Concerning technical aspects, there are some key issues that must be addressed properly in Grid Portal development in industrial contexts:
grid security and authentication aspects are critical both at Grid middleware-level and at access-level;
the authorization system should be built into the Grid system, enabling a fine-grained access control to resources (datasets, licenses, computing resources);
the accounting system, suitable to collect the resource usage and supporting reporting and billing services, should be able to collect the records from the various Grid nodes and merge them according to the business needs;
application integration and deployment to the Grid context, as well as administration should be standardized and simplified;
the access and the exploitation of Grid enabled applications by the end users should be simplified to the level of a web browsing experience;
the users shouldn’t need to be aware of the Grid infrastructure running the application, to perform their tasks.

For the industrial/engineering companies, the long and complex process that goes from the design of an industrial product to manufacturing, involves the cooperation of dozens or hundreds of people, departments or companies, often SMEs, ranging from engineering service providers to component suppliers. This can be regarded as a “virtual organization”, made of individual members or groups of people from the various companies that share, with a well defined role and profile, the overall project goal, often composed of geographically distant members, which would benefit from increased, real-time sharing of information and IT infrastructures, while preserving the intellectual properties of each of the project members. There are a number of factors, ranging from human, to organizational, to technical and to business aspects that are only partially addressed by current GRID technologies, that pratically limit the adoption of this approach.

The Web-centric approach lets users access any service virtually from anywhere, at any time, over any network and platform, including Personal Digital Assistant and Cellular Phones, thus supporting the ubiquitous access to the Grid.
Built on the experience of Industrial and Engineering requirements, the EnginFrame system has been designed to enable addressing effectively the above mentioned values, while minimizing the efforts to build and maintain a successful Grid Portal solution.

GENIUS Portal [1], based and powered by EnginFrame, jointly developed by INFN and NICE srl within the INFNGrid Project, allows in a very easy way the integration of applications ported to be executed on LCG-2 and gLite Middlewares, and many applications have been implemented on GILDA dissemination testbed [2] from the beginning and shown within dozens of tutorials, giving to the user an easy way to run jobs on the grid and to manage own data using the virtualizations offered by exposed services at different levels, locally, remotely, on catalogs. On the other hand, using the EnginFrame Framework, GENIUS Portal has inherited all the features, deriving from years of development and experience into industrial contexts, like scalability, flexibility, easy maintenance, security, fault tolerance, connectivity, data management, authorization, usability.

Conclusions.
The adoption of this innovative technology has given industries and engineering companies very important benefits in improvements in productivity running on Grid-enabled infrastructures. GENIUS, by staying aligned with the middleware development, can be an instrument to facilitate a dialog between research and industrial contexts based on a high-level services approach. This dialog can give also a very high added-value for both worlds, to spread the use of Grid infrastructures and generate a critical mass of awareness and trust.

References.
[1] "GENIUS: a simple and easy way to access computational and data grids" G. Andronico, R. Barbera, A. Falzone, P. Kunszt, G. Lo Rè, A. Pulvirenti, A. Rodolico - Future Generation of Computer Systems, vol. 19, no. 6 (2003), 805-813.
[2] "GILDA: The Grid INFN Virtual Laboratory for Dissemination Activities" G. Andronico, V. Ardizzone, R. Barbera, R. Catania, A. Carrieri, A. Falzone, E. Giorgio, G. La Rocca, S. Monforte, M. Pappalardo, G. Passaro, G. Platania - TRIDENTCOM 2005: 304-305.

Poster and Demo session + cocktail / 81

Parametric study workflow support by P-GRADE portal and MOTEUR workflow enactor

Author: Gergely Sipos[1]

Co-authors: Johan Montagnat [2]; Peter Kacsuk [1]; Tristan Glatard [2]; Zoltan Farkas [1]

[1] *MTA SZTAKI*

[2] *CNRS, I3S laboratory*

Corresponding Author: sipos@sztaki.hu

1. Composing and executing data-intensive workflows on the EGEE infrastructure

Grid computing is naturally very well suited for handling data-intensive applications involving the analysis of huge amounts of data. In many scientific areas the need for composing complex applications on grids from basic processing components has emerged. The classical task-based job description approach is providing a mean of depicting such applications but it becomes very tedious when trying to express complex application logics and large input data sets. Indeed, a different task needs to be described for each component and each input to consider. Higher level interfaces for easing the migration of applications to grid infrastructures are drastically needed. To ease the migration to grids of such complex and data intensive applications we are proposing a powerful tool which:

• Simplifies the application logic description through a graphical and intuitive editor.
• Enables the seamless integration of data intensive application running on different grid infrastructures.
• Permit try-and-retry experiments design and tuning through a flexible description and execution environment.
• Eases legacy code migration.
• Provides high level monitoring and trace analysis capabilities.

This tool is based on the integration of the PGRADE grid portal [1] and the MOTEUR workflow execution engine [2].

1. MOTEUR workflow execution engine

The service-based paradigm, plebiscited in the grid community, is elegantly enabling the composition of different application components through a common invocation interface. In addition, the service-based approach nicely decouples the description of processing logic (represented by services) and data to be processed (given as input parameters to these services). This is particularly important for describing the application logic independently from the experimental setting (the data to process).
MOTEUR is a service-based workflow enactor developed to efficiently process application workflows by exploiting the parallelism inherent to grid infrastructures. It is taking as input the application workflow description (expressed in Scufl language from the MyGrid project [3]) and the data sets to process. MOTEUR is orchestrating the execution of the application workflow by invoking asynchronously applications services. It takes care of processing dependencies and preserves the causality of computation on a highly distributed and heterogeneous environment.
Very complex data processing patterns may be described in a very compact way. In particular, the dot product (pairwise data composition) and cross product (all-to-all data composition) patterns from the Scufl language are very efficiently reducing complex data-intensive application graphs into much simpler ones. They significantly enlarge the expressiveness of the workflow language.
In addition, MOTEUR enables all level of parallelism that can be exploited in a data-intensive workflow: workflow parallelism (inherent to the workflow topology), data parallelism (different input data can be processed independently in parallel), and services parallelism (different services processing different data are independent and can be executed in parallel). To our knowledge, MOTEUR is the first service-based workflow enactor implementing all these optimizations.

1. The PGRADE portal GUI

During the last few years the P-GRADE portal has been chosen as the official portal by several Globus and LCG-2 middleware based Grid projects around Europe. In its original concept the P-GRADE Portal supported the development and execution of job-oriented workflows by the Condor DAGMan workflow manager. While DAGMan is a robust scheduler to submit jobs and to transfer input-output files among grid resources, it uses a quite simple scheduling algorithm, it is not able to invoke Web/Grid services and it cannot exploit every possible level of application parallelism (e.g.

pipelining).
To overcome these difficulties the P-GRADE portal has been integrated with the MOTEUR workflow manager. On top of that the P-GRADE Portal has been equipped with a universal interface by which it can be easily connected to other types of workflow engines. As a result every EGEE user community with its own application-specific scheduler can use the P-GRADE Portal to manage the execution of domain-specific programs on the connected Grids or VOs.
Based on the DAGMan and MOTEUR workflow managers the P-GRADE Portal supports the development and execution of stand-alone applications, parameter study applications and workflows composed from normal and/or parameter study components. These applications can be executed in LCG-2, Web services or Globus-based grids. During the execution the portal automatically selects the most appropriate plugged-in workflow manager to perform the scheduled submission of jobs, service invocation requests or data transfer processes.
The presentation introduces the capabilities of the MOTEUR-enabled P-GRADE Portal and the way in which the EGEE bioscience community is using it to solve a medical image processing problem. The community is going to develop a workflow of parameter study components that is capable to perform large number of operations on a huge set of medical images. The different components of the workflow represent Web services and are described by graphical notations. The MOTEUR workflow manager is responsible for the pipelined invocation of these Web services driven by the medical images and the different control input parameters.

[1] PGRADE portal, http://www.lpds.sztaki.hu/pgportal
[2] MOTEUR, http://www.i3s.unice.fr/_glatard/software.html
[3] UK eScience MyGrid project, http://www.mygrid.org

Summary:

This presentation addresses two special new features of the P-GRADE portal:
1. Enabling the efficient parallel Grid execution of parametric study type of applications both at job level and at workflow level. More than that it enables the creation and execution of workflows where certain components of the workflow are parametric study applications themselves.
2. Enabling the plug-in of any user community specific workflow enactors.
The talk will present how the MOTEUR enactor plug-in is handled and used in the portal in order to support the above mentioned parametric study workflow execution.

1d: Computational Chemistry - Lattice QCD - Finance / 82

An Attempt at Applying EGEE Grid to Quantum Chemistry

Authors: Mariusz Sterzel[1]; Tomasz Szepieniec[1]

Co-authors: Leszek Fiedor [2]; Mariusz Pilch [3]

[1] *Academic Computer Centre "Cyfronet"*

[2] *Faculty of Biotechnology, Jagiellonian University*

[3] *Faculty of Chemistry, Jagiellonian University*

Corresponding Author: m.sterzel@cyfronet.pl

The EGEE Grid Project enables access to huge computing and storage resources. Taking this oportunity we have tried to identyfie chemical problems that could be computed in this environment. Some of the results considered within this work are presented with description focused on requirements for the computational enviroment as well as techniques of Grid-enabling computations based on packages like GAMESS and GAUSIAN. Recently lots of work has been done in the area of parallelizing the existing codes

and discovering new ones used in quantum chemistry. That allows calculations to run much faster now than even ten years ago. However, there still exist tasks where without a large number of processors it is not possible to obtain satisfactory results. The two main challenges are harmonic frequency calculations and ab-initio (AI) molecular dynamics (MD) simulations. The former ones are mainly used to analyze molecular vibrations. Despite the fact that the algorithm for analytic harmonic frequency calculations has been known for over 20 years, only few quantum chemical codes have it implemented. The other still use numerical scheme where for a given number of atoms (N) in a molecule, , and for more accurate calculations independent steps (energy + gradients) have to be done to get harmonic frequencies. To achieve this as many processors as possible is needed to fit that huge number of calculations. This makes grids technology an ideal solution for that kind of application. The second challenge, MD simulations are mainly used in a case where 'static'calculation like for example determination of Nuclear Magnetic Resonance (NMR) chemical shifts gives wrong results. MD consists usually of two steps. In the first one the nuclear gradients are calculated, in the second one, based on obtained gradients, the actual classical forces acting on an atom are calculated. Knowing these forces one can estimate accelerations, velocities and guess new position of the atom after a given short period of time (so called time step). Finally the whole process is repeated for every new position of each atom. In case of mentioned NMR experiment we are interested in the average value of chemical shift over simulation. Of course NMR calculations are also very time consuming themselves and have to be done for many different geometries which again makes grid technology an ideal solution to final NMR chemical shift calculations.
We present here two kinds of calculations. First we show results for geometry optimization and frequency calculations for a few carotenoids. These molecules are of almost constant interest since they cooperate with chlorophyll in photosynthesis process. All the calculations have been done within EGEE Grid (VOCE VO). We also present an example of MD calculations and share our knowledge about what kind of problems can be found during such studies.

2a: Workload management and Workflows / 83

Logging and Bookkeeping and Job Provenance services

Authors: Ales Krenek[1]; Ludek Matyska[1]

Co-authors: Daniel Kouril [1]; Jan Pospisil [1]; Jiri Sitera [1]; Michal Vocu [2]; Milos Mulac [1]; Miroslav Ruda [1]; Zdenek Salvet [1]

[1] *CESNET, z.s.p.o.*

[2] *CESNET.z.s.p.o.*

Corresponding Author: ludek@ics.muni.cz

Logging and Bookkeeping (LB) service is responsible for keeping track of jobs within a complex Grid environment. Without such a service, users are unable to find out what happened with their lost jobs and Grid administrators are not able to improve the infrastructure. The LB service developed within the EGEE project provides a distributed scalable solution able to deal with hundreds thousands of jobs on large Grids. However, to provide the necessary scalability and not to slow down the processing of jobs within a middleware, it is based on a non-blocking asynchronous model. This means that the order of events sent to LB by individual parts of the middleware (user interface, scheduler, computing element, ...) is not guaranteed. While dealing with such out of order events, the LB may provide information that looks inconsistent with the knowledge user has from some other source (e.g. he got independent notification about the job state). The lecture will reveal LB internal design and we will discuss how the LB results (i.e. the job state) should be interpreted.
While LB is dealing with active jobs only, Job Provenance (JP) is

designed to store indefinitely information about all jobs that run on a Grid. All the relevant information needed to re-submit the job in the same environment is stored, including computing environment specification. Users can annotate stored records, providing yet another metadata layer useful e.g. for job grouping and data mining over the JP. We will provide basic information about the JP and its use, looking for a feedback for its improvement.

Poster and Demo session + cocktail / 84

VirtualGILDA: a virtual t-infrastructure for system administrator tutorials

Authors: Antonio Carrieri[1]; Emidio Giorgio[1]; Gianluca Passaro[1]; Gianni M. Ricciardi[1]; Giuseppe Andronico[1]; Giuseppe La Rocca[1]; Giuseppe Platania[1]; Lorenzo Neri[1]; Marco Pappalardo[1]; Rita Ricceri[1]; Roberto Barbera[1]; Rosanna Catania[1]; Salvatore Monforte[1]; Tony Calanducci[1]; Valeria Ardizzone[1]

[1] *INFN Catania*

Corresponding Author: rosanna.catania@ct.infn.it

In the Grid dissemination activity, teaching of Grid elements installation covers a very important role. While in tutorials for users availability of accounts and certificates is enough, in those ones for administrators a certain number of free machines is needed, and the requirements for a Grid-middleware compliant operating system also occurs.

The VirtualGILDA infrastructure for training aims at offering a set of Virtual Machine (VM), hosted in Catania and based on VMWare technology, with a pre-installed OS and net connectivity: in this way tutors have all the needed machines ready to use. They only need a reliable access to the Internet.

The presence of pre-installed Grid element is also possible, in order to provide tutors with a set of preconfigured machines ready to interact with elements that will be installed during the tutorial.

The use of VMWare technology is also suitable for on site tutorials, to avoid problems deriving from the wide range of machine and OS type available on each training site. Using VMs the only requirement is the presence of machines that can run VMPlayer , i.e. Linux or Windows hosts.

2d: VO tools - Portals / 85

Universal Acessibility to the Grid via Metagrid Infrastructure

Author: Soha Maad[1]

Co-authors: Brian Coghlan [1]; David O'Callaghan [1]; Eamonn Kenny [1]; Gabriel Pierantoni [1]; Oliver Lyttleton [1]

[1] *Trinity College Dublin*

Corresponding Author: soha.maad@cs.tcd.ie

This paper discusses the concept of universal accessibility [1, 2] to the grid within the context of selected application domains involving social interaction such as e-hospital, collaborative engineering, enterprise, e-government, and the media. Based on this discussion the paper proposes a metagrid infrastructure [3] as an approach to provide universal accessibility to the grid.

Universal accessibility is rooted in the concept of Design for All in Human Computer Interaction[1, 2]. It aims at efficiently and effectively addressing the numerous and diverse accessibility problems in human interaction with software applications and telematic services. So far, the key concept of universal accessibility has been supported by various development methodologies and platforms [4, 5]. Various application domains benefited from research and development in this area, including among others interactive television and media [6, 7]. Porting the concept of universal accessibility to the grid is faced by major obstacles attributed to the following: (a) the lack of an underlying functionality similar to that of a desktop operating system allowing the plug and play of resources and the direct user interaction with these resources; (b) the dilemma between hiding the grid versus making it more transparent; and (c) the software engineering practice adopted in grid middleware development, where the bottom up approach that is predominant [8] conflicts with the ethos of universal accessibility that considers accessibility at design time.

These obstacles and their impacts on universal accessibility to the grid are discussed with reference to four application domains including collaborative applications such as e-hospital, collaborative engineering, enterprise applications, the media, and e-government. In collaborative applications the key obstacle for universal accessibility to the grid is provision of interactivity while respecting various Service Level Agreements (SLAs). Several efforts are underway to resolve this issue [9, 21], but no versatile solutions have emerged so far. In the enterprise the major concern is the management of an integrated data centre [10]; the key obstacle confronted is that while already offering data-intensive computational power the grid is quite immature in its provision of permanent storage of data. This is very much a live issue in grid middleware development. In the media the major challenge is the direct access to remote external devices at the grid boundaries. For e-government accommodating various forms of interaction [11], such as government-to-government (G2G), government-to-citizen (G2C), and government-to-business (G2B), is paramount, whilst devoting a major focus on data semantics, not just structure.

So far universal accessibility to the grid was addressed from various perspectives. Efforts undertaken involved: (a) the development of grid middleware supporting interaction with heterogeneous mobile devices [12, 13]; (b) the use of operating system mobility for configuring grid application on a PC and then migrating the entire application together with the operating system instance onto the grid [14]; (c) the development of a shopping cart system based on the Web Service Resource Framework WSRF [15]; (d) the design of an approach for middleware development, based on wrapping the computational and resource intensive tasks, to allow the accessibility to the grid via hand held devices [16, 22]; (e) the development of common web-based grid application portals allowing the applications' users to customize their interfaces to the grid [17, 23, 24]; (f) the development of application models for the grid [18]; and (g) addressing security issues raised by granting grid accessibility via various media delivery channels (such as wireless devices) [19].

While each of these efforts towards universal accessibility to the grid does address the problem to some extent, none of them enables a complete solution. This paper proposes an approach, based on a metagrid infrastructure, that can potentially host solutions to all issues related to universal accessibility to the grid. This metagrid infrastructure was used thus far in the context of grid interoperability [3]. Our proposed approach extends the notion of interoperability to embrace grid application interoperability (interactivity and universal accessibility). While heavily based on existing grid middleware services and architecture such as EGEE, Globus, CrossGrid, GridPP and GGF [25, 26, 23, 27, 28], the metagrid infrastructure hosts one or more target grid techologies (e.g. it has been demonstrated simultaneously hosting WebCom, LCG2 and GT4) while also supporting its own services that provide things like universal accessibility that the target grid technologies do not. By doing so it firmly places the user within the metagrid environment rather than in any one target grid environment. The user obtains universal accessibility via the metagrid services, and the target grid technologies are relieved of the need to support direct user and device interactions.

By way of example, services currently offered by the metagrid infrastructure include a transparent grid filesystem [26] that supplies a vital missing component beneath existing middleware. The grid filesystem can support universal accessibility by supporting all forms of data access (r/w/x) in the course of collaborative interaction (collaborative engineering and e-hospital), by providing a logical user view of grid data (to support integration of the data centre in the enterprise), and by helping locate (discover) data in the course of interaction in media applications. In so doing it can improve the utility of, for example, the EGEE middleware. As further examples, proposed future services include special purpose discovery services to support various forms of interaction especially in media applications; and intelligent interpreters to support e-Government data semantics.

The paper is divided in five sections. The first section introduces the concept of universal accessibility and its relevance to the grid. The second section discusses existing obstacles facing universal accessibility to the grid in application domains involving social interaction. The third section overviews existing efforts towards universal accessibility to the grid. The fourth section propose an approach for universal accessibility to the grid based on a metagrid infrastructure and prototype services offered by this infrastructure. The paper concludes with a summary and a future research agenda.

= REFERENCES =

[1]:: Stephanidis, D. Akoumianakis, M. Sfyrakis, and A. Paramythis, Universal accessibility in HCI: Process-oriented design guidelines and tool requirements, Proceedings of the 4th ERCIM Workshop on User Interfaces for All, Edited by Constantine Stephanidis, ICS-FORTH, and Annika Waern, SICS, Stockholm, Sweden, 19-21 October 1998

[2]:: Stephandis, C., From User interfaces for all to an information society for all: Recent achievements and future challenges, Proceedings of the 6th ERCIM Workshop User Interfaces for All, October 2000, Italy

[3]:: Pierantoni, G. and Lyttleton, O. and O'Callaghan, D. and Quigley, G. and Kenny, E. and Coghlan, B., Multi-Grid and Multi-VO Job Submission based on a Unified Computational Model, Cracow Grid Workshop (CGW'05)Cracow, Poland, November 2005

[4]:: Stephanidis, C., Savidis, A., and Akoumianakis, D., Tutorial on Unified Interface Development: Tools for Constructing Accessible and Usable User Interfaces. Tutorial no. 13 in the 17th International Conference on Human Computer Interaction (HCI International'97), San Fransico, USA, 24-29 August. [Online] Available: http://www.ics.forth.gr/proj/at_hci/html/tutorials.htm

[5]:: Akoumianakis, D., Stephanidis, C., USE-IT : A Tool for Lexical Design Assistance. In C. Stephanidis (ed.) User Interfaces for All Concepts, Methods and Tools. Mahwah, NJ: 9. Beynon,

[6]:: Soha Maad, Universal Access For Multimodal ITV Content: Challenges and Prospects, Universal Access. Theoretical Perspectives, Practice, and Experience: 7th ERCIM International Workshop on User Interfaces for All, Paris, France, October 24-25, 2002. Revised Papers, N. Carbonell, C. Stephanidis (Eds.), Lecture Notes in Computer Science, Springer-Verlag Heidelberg, ISSN: 0302-9743, Volume 2615 / 2003, January 2003, pp.195-208.

[7]:: Soha Maad, Samir Garbaya, Saida Bouakaz , From Virtual to Augmented Reality in Finance: A CYBERII Application, to appear in the Journal of Enterprise Information Management

[8]:: S. Maad, B. Coghlan, G. Pierantoni, E. Kenny, J. Ryan, R. Watson, Adapting the Development Model of the Grid Anatomy to meet the needs of various Application

Domains, Cracow Grid Workshop (CGW'05), Cracow, Poland, November, 2005.

[9]:: Herbert Rosmanith, Dieter Kranzlmuller, glogin - A Multifunctional, Interactive Tunnel into the Grid, pp.266-272, Fifth IEEE/ACM International Workshop on Grid Computing (GRID'04), 2004.

[10]:: Soha Maad, Brian Coghlan, Eamonn Kenny, Gabriel Pierantoni, The Grid For the Enterprise: Bridging Theory and Practice, paper in progress, Computer Architecture Group, Trinity College Dublin.

[11]:: Maad S., Coghlan B., John R., Eamonn K., Watson R., and Pierantoni G. 2005, The Horizon of the Grid For E-Government, Proceeding eGovernment'05 Workshop, Brunel, United Kingdom, September 2005.

[12]:: Hassan Jameel, Umar Kalim, Ali Sajjad, Sungyoung Lee, Taewoong Jeon, Mobile-to-Grid Middleware: Bridging the Gap Between Mobile and Grid Environments, Advances in Grid Computing - EGC 2005, European Grid Conference, Amsterdam, The Netherlands, February 14-16, 2005, Editors: Peter M. A. Sloot, Alfons G. Hoekstra, Thierry Priol, Alexander Reinefeld, Marian Bubak, ISBN: 3-540-26918-5, Lecture Notes in Computer Science, Springer-Verlag GmbH, Volume 3470 / 2005, page 932.

[13]:: Ali Sajjad, Hassan Jameel, Umar Kalim, Young-Koo Lee, Sungyoung Lee, A Component-based Architecture for an Autonomic Middleware Enabling Mobile Access to Grid Infrastructure, Lecture Notes in Computer Science, Springer-Verlag GmbH, Volume 3823/2005, pages 1225 - 1234.

[14]:: Jacob Gorm Hansen, Eric Jul, Optimizing Grid Application Setup Using Operating System Mobility, Advances in Grid Computing - EGC 2005, European Grid Conference, Amsterdam, The Netherlands, February 14-16, 2005, Editors: Peter M. A. Sloot, Alfons G. Hoekstra, Thierry Priol, Alexander Reinefeld, Marian Bubak, ISBN: 3-540-26918-5, Lecture Notes in Computer Science, Springer-Verlag GmbH, Volume 3470 / 2005, page 952.

[15]:: Maozhen Li,Man Qi, Masoud Rozati, and Bin Yu, A WSRF Based Shopping Cart System, Advances in Grid Computing - EGC 2005, European Grid Conference, Amsterdam, The Netherlands, February 14-16, 2005, Editors: Peter M. A. Sloot, Alfons G. Hoekstra, Thierry Priol, Alexander Reinefeld, Marian Bubak, ISBN: 3-540-26918-5, Lecture Notes in Computer Science, Springer-Verlag GmbH, Volume 3470 / 2005, page 993.

[16]:: Saad Liaquat Kiani, Maria Riaz, Sungyoung Lee, Taewoong Jeon, Hagbae Kim, Grid Access Middleware for Handheld Devices, Advances in Grid Computing - EGC 2005, European Grid Conference, Amsterdam, The Netherlands, February 14-16, 2005, Editors: Peter M. A. Sloot, Alfons G. Hoekstra, Thierry Priol, Alexander Reinefeld, Marian Bubak, ISBN: 3-540-26918-5, Lecture Notes in Computer Science, Springer-Verlag GmbH, Volume 3470 / 2005, page 1002.

[17]:: Jonas Lindemann, Goran Sandberg, An Extendable GRID Application Portal, Advances in Grid Computing - EGC 2005, European Grid Conference, Amsterdam, The Netherlands, February 14-16, 2005, Editors: Peter M. A. Sloot, Alfons G. Hoekstra, Thierry Priol, Alexander Reinefeld, Marian Bubak, ISBN: 3-540-26918-5, Lecture Notes in Computer Science, Springer-Verlag GmbH, Volume 3470 / 2005, page 1012.

[18}:: Fei Wu, K.W. Ng, A Loosely Coupled Application Model for Grids, Advances in Grid Computing - EGC 2005, European Grid Conference, Amsterdam, The Netherlands, February 14-16, 2005, Editors: Peter M. A. Sloot, Alfons G. Hoekstra, Thierry Priol, Alexander Reinefeld, Marian Bubak , ISBN: 3-540-26918-5, Lecture Notes in Computer Science, Springer-Verlag GmbH, Volume 3470 / 2005, page 1056

[19]:: Syed Naqvi, Michel Riguidel, Threat Model for Grid Security Services, Advances in Grid Computing - EGC 2005, European Grid Conference, Amsterdam, The Netherlands, February 14-16, 2005, Editors: Peter M. A. Sloot, Alfons G. Hoekstra, Thierry Priol, Alexander Reinefeld, Marian Bubak , ISBN: 3-540-26918-5, Lecture Notes in Computer Science, Springer-Verlag GmbH, Volume 3470 / 2005, page 1048

[20]:: Soha Maad, Brian Coghlan, Geoff Quigley, John Ryan, Eamonn Kenny, David

O'Callaghan, Towards a Complete Grid Filesystem Functionality, submitted to special issue on Data Analysis, Access and Management on Grids, CALL FOR PAPERS , Future Generation Computer Systems, The International Journal of Grid Computing: Theory, Methods and Applications, Elsevier.

[21]:: EU FP6 Project 031857: int.eu.grid, to start May, 2006.

[22]:: Genius Portal, https://genius.ct.infn.it/

[23]:: Marian Bubak, Michal Turala, CrossGrid and Its Relatives in Europe, Proc.9th European PVM/MPI Users Group Meeting, LNCS, pp.14-15, Vol.2474, ISBN: 3-540-44296-0, Springer-Verlag, 2002.

[24]:: M.Kupczyk, R.Lichwala, N.Meyer, B.Palak, M.Plociennik, P.Wolniewicz, Applications on Demand as the exploitation of the Migrating Desktop, Future Generation Computer Systems, pp.37-44, Vol.21, Issue 1, ISSN: 0167-739X, January 2005.

[25]:: EU FP6 Project: Enabling Grids For E-sciencE, http://www.eu-egee.org/

[26]:: Globus Project, http://globus.org

[27]:: GridPP Project, http://www.gridpp.ac.uk/

[28]:: Global Grid Forum (GGF), http://www.ggf.org/

2b: Data access on the grid / 86

gLibrary: a Multimedia Contents Management System on the grid

Authors: Tony Calanducci[1]; Valeria Ardizzone[1]

Co-authors: Emidio Giorgio [1]; Gianluca Passaro [1]; Giuseppe Andronico [1]; Giuseppe La Rocca [1]; Giuseppe Platania [1]; Marco Pappalardo [1]; Roberto Barbera [1]; Rosanna Catania [1]; Salvatore Monforte [1]

[1] *INFN Catania*

Corresponding Author: tony.calanducci@ct.infn.it

Nowadays huge amounts of information are searched and used by people from all over the world, but it is not always easy to find out what one is looking for. Search engines helps a lot, but they do not provide a standard and uniform way to make queries.
The challenge of gLibrary is to design and develop a robust system to handle Multimedia Contents in a easy, fast and secure way exploiting the Grid.
Examples of Multimedia Contents are images, videos, music, all kind of electronic documents (PDF, Excel, PowerPoint, Word, HTML), E-Mails and so on. New types of content can be added easily into the system.
Thanks to the fixed structure of the attributes per each content type, queries are easier to perform allowing the users to choose their search criteria among a predefined set of attributes.
The following are possible use examples:
- A user wants to look for all the comedies in which Jennifer Aniston performed together with Ben Stiller, produced in 2004 ; or find all the songs of Led Zeppelin that last for more than 6 minutes;
- An user needs to find all the PowerPoint Presentation about Data Management System in 2005 run by Uncle Sam (fantasy name);
- A doctor wants to retrieve all the articles and presentations about lung cancer and download some lung X-ray images to be printed in his article for a scientific magazine;
- (Google for storage) a job behaves as a "storage crawler": it scans all the files stored in Storage Elements and publishes their related specific information into gLibrary for later searches through their attributes.

Not all the users of the system have the same authority into the system. Three kind of users are enabled: gLibrary Generic Users, members of a Virtual Organization recognized by the system, can browse the library and make queries. They can also retrieve the wanted files if the submitter user authorized them; gLibrary Submitter Users can upload new entries attaching them the proper values for the defined attributes; finally gLibrary Administrator are allowed to define new content type and elect Generic User granting them submission rights.
A first level of security on single file is implemented: files uploaded to Storage Elements can be encrypted using a symmetric key. This will be placed in a special directory into the system and the submitter will define which users are the rights to read it.
All the application is built on top of the grid services offered by the EGEE middleware: actual data is stored in Storage Elements spread around the world, while the File Catalog keeps track of where they are located. A Metadata Catalog service is intensively used to contains the values of attributes and satisfy user's queries. Finally, A Virtual Organization Membership Service comes in help to deal with authorization.

2a: Workload management and Workflows / 87

The gLite Workload Management System

Author: Francesco Giacomini[1]

[1] *Istituto Nazionale di Fisica Nucleare (INFN)*

Corresponding Author: francesco.giacomini@cern.ch

The Workload Management System (WMS) is a collection of components providing a service responsible for the distribution and management of tasks across resources available on a Grid, in such a way that applications are conveniently, efficiently and effectively executed.

The main purpose of the WMS as a whole is then to accept a request of execution of a job from a client, find appropriate resources to satisfy it and follow it until completion, possibly rescheduling it, totally or in part, if an infrastructure failure occurs. A job is always associated to the credentials of the user who submitted it. All the operations performed by the WMS in order to complete the job are done on behalf of the owning user. A mechanism exists to renew credentials automatically and safely for long-running jobs.

The different aspects of job management are accomplished by different WMS components, usually implemented as different processes communicating via data structures persistently stored on disk to avoid as much as possible data losses in case of failure.

Recent releases of the WMS come with a Web Service interface that has replaced the custom interface previously adopted. Moving to formal or de-facto standards will continue in the future.

In order to track a job during its lifetime, relevant events (such as submission, resource matching, running, completion) are gathered from various WMS components as well as from Grid resources (typically Computing Elements), which are properly instrumented. Events are kept persistently by the Logging and Bookkeeping Service (LB) and indexed by a unique, URL-like job identifier. The LB offers also a query interface both for the logged raw events and for higher-level task state. Multiple LBs may exist, but a job is statically assigned to one of them. Being the LB designed, implemented and deployed so that the service is highly reliable and available, the WMS heavily relies on it as the authoritative source for job information.

The types of job currently supported by the WMS are diverse: batch-like, simple workflow in the form of Directed Acyclic Graphs (DAGs), collection, parametric, interactive, MPI, partitionable, checkpointable. The characteristics of a job are expressed using a flexible language called Job Description Language (JDL). The JDL also allows the specification of constraints and preferences on the resources that can be used to execute the job. Moreover some attributes exist that are useful for the management of the job itself, for example how much to insist with a job in case of repeated failures or lack of resources.

Of the above job types, the parametric jobs, the collections, and the workflows have recently received special attention.

A parametric job allows the submission of a large number of almost identical jobs simply specifying a parameterized description and the list of values for the parameter.

A collection allows the submission of a number of jobs as a single entity. An interesting feature in this case is the possibility to specify a shared input sandbox. The input sandbox is a group of files that the user wishes to be available on the computer where the job runs. Sharing a sandbox allows some significant optimization in network traffic and, for example, can greatly reduce the submission time.

Support for workflows in the gLite WMS is currently limited to Directed Acyclic Graphs (DAGs), consisting of a set of jobs and a set of dependencies between them. Dependencies represent time constraints: a child cannot start before all parents have successfully completed. In general jobs are independently scheduled and the choice of the computing resource where to execute a job is done as late as possible. A recently added feature allows to collocate the jobs on the same resource. Future improvements will mainly concern error handling and integration with data management.

Parametric jobs, collections and workflows have their own job identifier, so that all the jobs belonging to them can be controlled either independently or as a single entity.

Future developments of the WMS will follow three main lines: stronger integration with other services, software cleanup, and scalability.

The WMS already interacts with many external services, such as Logging and Bookkeeping, Computing Elements, Storage Elements, Service Discovery, Information System, Replica Catalog, Virtual Organization Membership Service (VOMS). Integration with a policy engine (G-PBox) and an accounting system (DGAS) is progressing; this will ease the enforcement of local and global policies regulating the execution of tasks over the Grid, giving fine control on how the available resources can be used. Designing and implementing a WMS that relies on external services for the above functionality is certainly more difficult than providing a monolithic system, but in fact doing so favors a generic solution that is not application specific and can be deployed in a variety of environments.

The cleanup will affect not only the existing code base, but will also aim at improving the software usability and at simplifying service deployment and management. This effort will require the evaluation and possibly the re-organization of the current components, yet keeping the interface.

Last but not least, considerable effort needs to be spent on the scalability of the service. The functionality currently offered already allows many kinds of applications to port their computing model onto the Grid. But additionally some of those applications have demanding requirements on the amount of resources, such as computing, storage, network, and data, they need to access in order to accomplish their goal. The WMS is already designed and implemented to operate in an environment with multiple running instances not communicating with each other and seeing the same resources. This certainly helps in case the available WMSs get overloaded: it is almost as simple as starting another instance. Unfortunately this approach cannot be extended much further because it would cause too much contention on the available resources. Hence the short term objective is to make a single WMS instance able to manage 100000 jobs per day. In the longer term it will be possible to deploy a cluster of instances sharing the same state.

Poster and Demo session + cocktail / 88

Application Identification and Support in BalticGRID

Author: Algimantas Juozapavicius[1]

[1] *associate professor*

Corresponding Author: algimantas.juozapavicius@maf.vu.lt

Introduction

The Baltic Grid project, a FP6 program, involving 10 leading institutions in six countries, started in November 2005. Its aims to i) develop and integrate the research and education computing and communication infrastructure in the Baltic States into the emerging European Grid infrastructure, ii) bring the knowledge in Grid technologies and use of Grids in the Baltic States to a level comparable to that in EU members states, and iii) further engage the Baltic States in policy and standards setting activities. The integration of Baltic States into the European Grid infrastructure is primarily focusing on extending the EGEE (with which four partners are already engaged) to the Baltic States. The Baltic Grid takes advantage of the local existing e-infrastructures in the region.
The Baltic Grid project is of high strategic importance for the Baltic States and it is designed to give a rapid build-up of a Grid infrastructure, contributing to the enabling of the new member states participation in the European Research Area. One of the most important steps in Baltic Grid development is application identification and support. This activity will be carried out through three tasks.

Pilot Applications

Baltic Grid intends to initiate three pilot applications for validation and for demonstration of successful scientific use.

High-energy physics application includes statistical data analysis, production of Monte Carlo samples and distributed data analysis, nuclear and sub-nuclear physics, condensed matter physics and many-body problems. It will be implemented because of the critical importance of Grids to this community and its relative maturity.

Material sciences application presents research areas, having substantial number of potential Grid users among scientists in Baltic states. It includes tools for establishing the geometrical structure of various organic, metal-organic and inorganic materials; understanding optical and magnetic properties of molecular derivatives; predicting new technology and creation of new materials with specified characteristics. Modelling and simulation of heterogeneous processes in chemistry, biochemistry, geochemistry, electrochemistry, biology, engineering will be implemented because of MS strategic importance to the Baltic States and substantial computing needs.

A bioinformatics application will be implemented to give tools and computing procedures for sequence pattern discovery and the gene regulatory network reconstruction, inference of haplotype structure and pharmacogenetics related association, studies, modelling and exploration of mechanism of enzymatic catalysis, de novo design of proteins, quantum-mechanical investigations of organic molecules and their applications, for the refinement of 3D biological macromolecule models against X-ray diffraction or NMR data, for modeling of biosensors and other reaction-diffusion processes. This application intends also to support the collaborative efforts of scientists in the Baltic States in this highly distributed community with needs to share data from many sources and a diverse set of tools.

Special Interest Groups

The task of special interest groups (SIG) aims to improve communication among many separate research groups, having similar or related R&D interests. The development and implementation of SIGs is a relatively new idea in grid computing infrastructure based on semantics representation methods and tools and leading to enhancement of services and applications with knowledge and semantics. Research areas under consideration for SIG development and implementation are: modelling of the Baltic Sea eco-system (together with BOOS –a future operational oceanographic service to the marine industry in the Baltic region), hydrodynamic environmental models for sustainable development of the Baltic Sea coastal zone, environmental impact assessment and environmental processes modeling, life sciences and medicine.

Application Adaptation Support

This is a specific activity aiming to organize and initiate communication between application experts and Grid experts facilitating rapid Grid adaptation and deployment of applications through formation of an Application Expert Group. This group will analyze applications and identify required Grid technologies and provide consulting services to application developers. The services will include assistance with integration with the Migrating Desktop to enable GUI-based access to the BG infrastructure and services, ensuring interoperability with the BG middleware. Performance studies to find bottle necks of the deployed applications may be carried out if needed using tools for performance evaluation, like G-PM and OCM-G, developed in CrossGrid Project.

2c: Special type of jobs (MPI, SDJ, interactive jobs, …) - Information systems / 89

Scheduling Interactive Jobs

Author: Cecile Germain-Renaud[1]

Co-authors: Charles Loomis [2]; Romain Texier [3]

[1] *LRI and LAL*

[2] *LAL*

[3] *LRI*

Corresponding Author: germain@lal.in2p3.fr

1.Introduction

In the 70s, the transition from batch systems to interactive computing has been the enabling tool for the widespread diffusion of advances in IC technology. Grids are facing the same challenge. The exponential coefficients in network performance enable the virtualization and pooling of processors and storage; large-scale user involvement might require seamless integration of the grid power into everyday use.

In this paper,interaction is a short name for all situations of display-action loop, ranging from a code-test-debug process in plain ascii, to computational steering through virtual/augmented reality interfaces, as well as portal access to grid resources, or complex and partially local workflows. At various levels, EGEE HEP and biomedical communities provide examples of the requirements of a turnaround time at the human scale. Section 2 will provide experimental evidence on this fact.

Virtual machines provide a powerful new layer of abstraction in distributed computing environments. The freedom of scheduling and even migrating an entire OS and associated computations considerably eases the coexistence of deadline bound short jobs and long running batch jobs. The EGEE execution model is not based on such virtual machines, thus the scheduling issues must be addressed through the standard middleware components, broker and local schedulers. Section 3 and 4 will demonstrate that QoS and fast turnaround time are indeed feasible within these constraints.

1. EGEE usage The current use of EGEE makes a strong case for a specific support for short jobs. Through the analysis of the LB log of a broker, we can propose quantitative data to support this affirmation. The broker logged is grid09.lal.in2p3.fr, running successive versions of LCG; the trace covers one year (October 2004 to October 2005), with 66 distinct users and more than 90000 successful jobs, all production. This trace provides both the job intrinsic execution time t (evaluated as the timestamp of event 10/LRMS minus the timestamp of event 8/LRMS), and the makespan m, that is the time from submission to completion (evaluated as the timestamp of event 10/LogMonitor minus the timestamp of event 17/UI). The intrinsic execution time might be overestimated if the sites where the job is run accept concurrent execution.

The striking fact is the very large number of extremely short jobs. We call Short Deadline Jobs (SDJ) those where t < 10 minutes, and Medium Jobs (MJ) those with t between ten minutes and one hour. SDJ account for more than 90% of the total number of jobs, and consume nearly 20 of the total execution time, in the same range as jobs with t less than one hour (17%).
Next, we considering the overhead o =(m-t)/t. As usual, the overhead decreases with execution time, but for SDJ, the overhead is often many orders of magnitude superior to t. For MJ, the overhead is of the same order of magnitude as t.
Thus, the EGEE service for SDJ is seriously insufficient.
One could argue that bundling many SDJ into one MJ could lower the overhead. However, interactivity will not be reached, because results will also come in a bundle: for graphical interactivity, the result must obviously be pipelined with visualization; in the test-debug-correct cycle, there might be not very many jobs to run.

With respect to grid management, an interactivity situation translates into a QoS requirement: just as video rendering or music playing requires special scheduling on a personal computer, or video streaming requires network differentiated services, servicing SDJ requires a specific grid guarantee, namely a small bound on the makespan, which is usually known as a deadline in the framework of QoS. The overhead has two components: first the queuing time, and second the cost of traversal of the middleware protocol stack. The first issue is related to the grid scheduling policy, while the second is related to grid scheduling implementation.

1. A Scheduling Policy for SDJ

Deadline scheduling usually relies on the concept of breaking the allocation of resources into quanta, of time for a processor, or through packet slots for network routing. For job scheduling, the problem is a priori much more difficult, because jobs are not partitionable: except for checkpointable jobs, a job that has started running cannot be suspended and restarted later. Condor has pioneered migration-based environments, which provide such a feature transparently, but deploying constrained suspension in EGEE would be much too invasive, with respect to existing middleware. Thus, SDJ should not be queued at all, which seems to be incompatible with the most basic mechanism of grid scheduling policies.

The EGEE scheduling policy is largely decentralized: all queues are located on the sites, and the actual time scheduling is enacted by the local schedulers. Most often, these schedulers do not allow time-sharing (except for monitoring). The key for servicing SDJ is to allow controlled time-sharing, which transparently leverages the kernel multiplexing to jobs, through a combination of processor virtualization and slot permanent reservation. The SDJ scheduling system has two components.
- A local component, composed of dedicated single-entry queues and a configuration of the local scheduler. Technical details for can be found at http://egee-na4.ct.infn.it/wiki/index.php/ShortJobs. It ensures the followig properties: the delay incurred by batch jobs is at most doubled; the resource usage is not degraded, eg by idling processors; and finally the policies governing resource sharing (VOs, EGEE and non EGEE users,…) are not impacted.
- A global component, composed of job typing and mapping policy at the broker level. While it is easy to ensure that SDJ are directed to resources accepting SDJ, LCG and gLite do not provide the means to prevent non-SDJ jobs from using the SDJ queues, and this requires a minor modification of the broker code.

It must be noticed that no explicit user reservation is required: seamless integration also means that explicit advance reservation is no more applicable than it would be for accessing a personal computer or a video-on-demand service.

In the most frequent case, SDJ will run with under the best effort Linux scheduling policy (SCHED_OTHER); however, if hard real-time constraints must be met, this scheme is fully compatible with preemption (SCHED_FIFO or SCHED_RR policies). In any case, the limits on resource usage(e.g. as enforced by Maui) implement access control; thus the job might be rejected. The WMS notifies rejection to the application, which could decide on the most adequate reaction, for instance submission as a normal job or switching to local computation.

1. User-level scheduling Recent reports (gLite WMS Test) show impressively low middleware penalty, in the order of a few seconds, which should be available in gLite3.0. It also hints that the broker is not too heavily impacted by many simultaneous access. However, for ultra-small jobs, with execution time of the same order (XXSDJ), even this penalty is too high. Moreover, the notification time remains in the order of minutes. In the gPTM3D project, we have shown that an additional layer of user-level scheduling provides a solution which is fully compatible with EGEE organization of sharing. The scheduling and execution agents are quite different from those in Dirac: they do not constitute a permanent overlay, but are launched just as any LCG/gLite job, namely an SDJ job; moreover, they work in connected mode, more like glogin-based applications. Besides this particular case, an open issue is the internal SDJ scheduling. Consider for instance a portal, where many users ask for a continuous stream of execution of SDJ (whether XXSDJ or regular SDJ). The portal could dynamically launch such scheduling/worker agents and delegate to them the implementation of the so-called (period, slice) model used in soft real-time scheduling.

1a: Life Sciences / 90

Encrypted File System on the EGEE grid applied to Protein Sequence Analysis

Author: Christophe Blanchet[1]

Co-authors: Gilbert Deleage [1]; Rémi Mollon [1]

[1] *CNRS IBCP*

Corresponding Author: christophe.blanchet@ibcp.fr

Introduction
Biomedical applications are pilot ones in the EGEE project [1][2] and have their own virtual organization: the
"biomed"VO. Indeed, they have common security requirements such as electronic certificate system, authentication, secured transfer; but they have also specific ones such as fine grain access to data, encrypted
storage of data and anonymity. Certificate system provides biomedical entities (like users, services or Web
portals) with a secure and individual electronic certificate for authentication and authorization management.
One key quality of such a system is the capacity to renew and revoke these certificates across the whole grid.
Biomedical applications also need fine grain access (with Access Control Lists, ACLs) to the data stored on the
grid: biologists and biochemists can then, for example, share data with colleagues working on the same
project in other places. Thus, biomedical data need to be gridified with a high level of confidentiality because
they can concern patients or sensitive scientific/industrial experiments. The solution is then to encrypt the
data on the Grid storage resources, but to provide authorized users and applications with transparent and
unencrypted access.

Biological data and protein sequence analysis applications
Biological data and bioinformatics programs have both special formats and behaviors, especially highlighted
when they are used into a distributed computing platform such as grid [2].
Biological data represent very large datasets of different nature, from different sources, with heterogeneous
models: protein three-dimensional structures, functional signatures, expression arrays, etc. Bioinformatics
experiences use numerous methods and algorithms to analyze whole biological data which are available to the
community [3]. For each domain of Bioinformatics, they are several different high-quality
programs that are available for computing the same dataset in as many ways. But most bioinformatics
programs are not adapted to distributed platform. One important disadvantage is that they are only accessing
data with local file system interface to get the input data and to store their results, an other one being that
these data must be unencrypted.

The European EGEE grid
The Enabling Grids for E-sciencE project (EGEE) [4], funded by the European Commission, aimed to build on
recent advances in grid technology and to develop a service grid infrastructure such as described by Foster et
al. at the end of 1990s [5].
The EGEE middleware provides grid users with a "user interface"(UI) to launch a job. Among the components
of the EGEE grid: the "workload management system"(WMS) is responsible of job scheduling. The central
piece is the scheduler (or "resource broker") that determines where and when to send a job on the "computing
elements"(CE) and get data from the "storage elements"(SE). The "data management system"(DMS) is a key
service for our bioinformatics applications. Having efficient usage of DMS will be synonymous of good

distribution of our protein sequence analysis applications. Inside the DMS, the "replica manager system"(RMS)
provides users with data replica functionalities. But there is no available encryption service onto the
production grid of EGEE, built upon the LCG2 middleware.

"EncFile"encrypted file manager
We have developed the EncFile, encrypted file management system, to provide our bioinformatics applications
with facilities for computing sensitive data on the EGEE grid. The cipher algorithm AES (Advanced Encryption
Standard) is used with 256 bits keys. And to bring fault tolerance properties to the platform, we have also
applied a M-of-N technique described by Shamir for secret sharing [6]. We split a key into N shares, each
stored in a different server. To rebuild a key, exactly M of the N shares are needed. With less than M shares, it
is impossible to deduce several bits or even one of them.
The "EncFile"system is composed of these N key servers and one client. The client is doing the decryption of
the file for the legacy application, and is the only component able to rebuild the keys, securing their confidentiality. The transfer of the keys between the M servers and the client is secured with encryption and
mutual authentication. In order to determine user authorization, the EncFile client send the user proxy to
authenticate itself. Nonetheless, to avoid that a malicious person creates a fake EncFile client (e.g. to retrieve
key shares), a second authentication is required with a specific certificate of the EncFile system.
As seen before, most bioinformatics programs are only able to access their data through local file system
interface, and also not encrypted. To answer to these 2 strong issues, we have combined the EncFile client
and the Parrot software [7]. The resultant client (called Perroquet in Figure 1) acts as a launcher for applications, catching all their standard IO calls and replacing them with equivalent remote calls to remote
files. Perroquet understands the logical file name (LFN) locators of our biological resources onto the EGEE grid,
and do on-the-fly decryption. This has mainly two consequences: (i) higher security level because decrypted
file copies could endanger data, (ii) better performances because files aren't read twice to locally copy and to
decrypt.
Thus, the EncFile client permits any applications to transparently read and write remote files, encrypted or
not, as if they were local and plain-text files. We are using EncFile system to secure sensitive biological data
on the EGEE production platform and to analyze them with world-famous legacy bioinformatics applications
such as BLAST, SSearch or ClustalW.

Conclusion
We have developed the EncFile system for encrypted files management, and deployed it on the production
platform of the EGEE project. Thus, we provided grid users with a user-friendly component that doesn't
require any user privileges, and is fault-tolerant because of the M-of-N technique, used to deploy key shares
on several key servers. The EncFile client provides legacy bioinformatics applications with remote data access,
such as the ones used daily for genomes analyses.

Acknowledgement
This works was supported by the European Union (EGEE project, ref. INFSO-508833). Authors express thanks
to Douglas Thain for the interesting discussions about the Parrot tool.

References
[1] Jacq, N., Blanchet, C., Combet, C., Cornillot, E., Duret, L., Kurata, K., Nakamura, H., Silvestre, T., Breton,
V. : Grid as a bioinformatics tool. , Parallel Computing, special issue: High-performance parallel bio-
computing, Vol. 30, (2004).
[2] Breton, V., Blanchet, C., Legré, Y., Maigne, L. and Montagnat, J.: Grid Technology for Biomedical Applications. M. Daydé et al. (Eds.): VECPAR 2004, Lecture Notes in Computer Science 3402, pp. 204–218,
2005.
[3] Combet, C., Blanchet, C., Geourjon, C. et Deléage, G. : NPS@: Network Protein Sequence Analysis. Tibs,
25 (2000) 147-150.
[4] Enabling Grid for E-sciencE (EGEE). Online: www.eu-egee.org
[5] Foster, I. And Kesselman, C. (eds.) : The Grid 2 : Blueprint for a New Computing Infrastructure, (2004).
[6] Shamir., A. "How to share a secret". Communications of the ACM , 22(11):612–613, Nov. 1979.
[7] Thain, D. and Livny, M.: Parrot: an application environment for data-intensive computing. Scalable
Computing: Practice and Experience 6 (2005) 9-18

Summary:

Biomedical applications are pilot ones in the EGEE project and have their own virtual organization: the "biomed"VO. Indeed, they have common security requirements such as electronic certificate system, authentication, secured transfer; but they have also specific ones such as fine grain access to data, encrypted
storage of data and anonymity. Thus, biomedical data need to be gridified with a high level of confidentiality
because they can concern patients or sensitive scientific/industrial experiments. The solution is then to encrypt the data on the Grid storage resources, but to provide authorized users and applications with transparent and unencrypted access.
We have developed the EncFile system for encrypted files management, and deployed it on the production
platform of the EGEE project. Thus, we provided grid users with a user-friendly component that doesn't
require any user privileges, and is fault-tolerant because of the M-of-N technique, used to deploy key shares
on several key servers. The EncFile client provides legacy bioinformatics applications with remote data access,
such as the ones used daily for genomes analyses.

1a: Life Sciences / **91**

GPS@: Bioinformatics grid portal for protein sequence analysis on EGEE grid

Author: Christophe Blanchet[1]

Co-authors: Christophe Combet [1]; Gilbert Deleage [1]; Vincent Lefort [1]

[1] *CNRS IBCP*

Corresponding Author: christophe.blanchet@ibcp.fr

One of current major challenges in the bioinformatics field is to derive valuable information from the complete
genome sequencing projects, which provide the bioinformatics community with a large number of unknown
sequences. The first prerequisite step in this process is to access up-to-date sequence and 3D-structure
databanks (EMBL, GenBank, SWISS-PROT, Protein Data Bank…) maintained by several bio-computing centres
(NCBI, EBI, EMBL, SIB, INFOBIOGEN, PBIL, ⋯). For efficiency reasons, sequences should be analyzed using the
maximal number of methods on a minimal number of different Web sites. To achieve this, we developed a
Web server called NPS@ [1] (Network Protein Sequence Analysis) that provides biologists with many of the
most common tools for protein sequence analysis through a classic Web browser like Netscape, or through a
networked protein client software like MPSA [2]. Today, the genomic and post-genomic web portals available
have to deal with their local cpu and storage resources. That's why, most of the time, the portal
administrators put some restrictions on the methods and databanks available. Grid computing [3], as in the
European EGEE project [4], will be a viable solution to foresee these limitations and to bring computing
resources suitable to the genomic research field.
Nevertheless, the current job submission process on the EGEE platform is relatively complex and unsuitable
for automation. The user has to install an EGEE user interface machine on a Linux computer (or to ask for a
account on a public one), to remotely log on it, to init manually a certificate proxy for authentication reasons,
to specify the job arguments to the grid middleware using the Job Description Language (JDL) and then to
submit the job through a command line interface. Next, the grid-user has to check periodically the resource
broker for the status of his job: "Submitted", "Ready", "Scheduled", "Running", etc. until the "Done" status. As a
final command, he has to get his results with a raw file transfer from the remote storage area to his local file
system.
This mechanism is most of times off-putting scientist that are not aware of advanced computing techniques.
Thus, we decide to provide biologists with a user-friendly interface for the EGEE computing and storage
resources, by adapting our NPS@ web site. We have called this new portal GPS@ for "Grid Protein Sequence
Analysis", and it can be reached online at http://gpsa.ibcp.fr, yet for experimental tests only. In GPS@, we
simplify the grid analysis query: GPS@ Web portal runs its own EGEE low-level interface and provides
biologists with the same interface that they are using daily in NPS@. They only have to paste their protein
sequences or patterns into the corresponding field of the submission web page. Then simply pressing the
"submit"button launches the execution of these jobs on the EGEE platform. All the EGEE job submission is
encapsulated into the GPS@ back office: scheduling and status of the submitted jobs. And finally the result of
the bioinformatics jobs are displayed into a new Web page, ready for other analyses or for results download in
the appropriate data format.

[1] NPS@: Network Protein Sequence Analysis. Combet C., Blanchet C., Geourjon C. et Deléage G. Tibs, 2000,
25, 147-150.
[2] MPSA: Integrated System for Multiple Protein Sequence Analysis with client/server capabilities. Blanchet
C., Combet C., Geourjon C. et Deléage G. Bioinformatics, 2000, 16, 286-287.
[3] Foster, I. And Kesselman, C. (eds.) : The Grid 2 : Blueprint for a New Computing Infrastructure, (2004).
[4] Enabling Grid for E-sciencE (EGEE), online at www.eu-egee.org

92

VEGA : Virtual Environments for Grid Applications

Author: Toan Nguyen[1]

Co-author: Lizhe Wang [1]

[1] *INRIA*

Corresponding Author: toan.nguyen@inrialpes.fr

VEGA
Virtual Environments for Grid Applications

Toan Nguyen, Lizhe Wang
INRIA Rhône-Alpes
Toan.Nguyen@inrialpes.fr
+33 476 61 52 40

The widespread dissemination of grid technology during the last decade has opened new expectations from both the technological and the applications perspectives. The well-known "technology push"and "application pull"paradigm reflects precisely the dynamic equilibrium that state-of-the-art in RTD is seeking. This equilibrium is still moving fast, and probably too fast for a majority of potential users.
The equilibrium varies along paths that experts try to predict to their best knowledge. But users often adopt technologies without foreseeable prediction. The outstanding example of this state of affairs lies in the worldwide expansion of the Internet. This now classical example of dedicated technology which has become the technological revolution of the 20th century's end exemplifies the trend that the Grid might expect.
However, the maturing of technology does not suffice to provide acceptable tools for the users. The seemingly reserved adoption of Grid technology by the industry today cannot be totally explained by the lack of tools and environments suitable for application design and deployment.
Among the ongoing RTD aspects are the security and Quality of Service items. But more simply is also the lack of seamless accessibility tools. Grid technology carries a technical complexity outlook that refrains the vast majority of its potential users.
Based on these assumptions, this project aims at providing simple tools to allow the application designers and users of the Grid to ease its use. It will thus sontribute to the uptake of grid technology amongst its huge potential user community.
Building on the claimed virtualization of resources and Virtual Organizations concepts already worked out in the Grid community, the VEGA project proposes to develop user-oriented functionalities for application design, deployment and monitoring on the Grid.
It builds on existing Grid technology to enable application practitioners to seamlessly access and use it. This is far from being the case today, although struggling efforts are currently being undertaken on a large variety of items to mask the technicalities and intricacies of the Grid, in order to make

it “transparent”. But there is still a long way to go.
Loosely based on the notion of Virtual Organization (VO), the VEGA project offers the users of the Grid a specific concept that will facilitate the use of VO. More specifically, the project will implement, deploy and test the concept of Virtual Environments (VE), specifically aimed at providing the application designers and users with a dedicated set of tools, protocols and services to support seamlessly their applications in the dynamic grid environments.
Virtual Environments are not specific to any single middleware, although their deployment and testing on realistic applications will be tuned to operational grid infrastructures. They are not specific to any application domain either, and testcases are provided in scientific and business areas. In contrast with VOs, they don’t mimic any working organizations: there are no roles, membership and groups in VEs. They are application-centric, not user-centric services. The generic nature of Virtual Environments allows for easy adaptation to specific hardware infrastructures, operating systems software, middleware and applications.
Of particular interest are multidiscipline applications in engineering, environmental monitoring applications and peta data management application in citizens’life. These will be used as testcases for the validation of the VEGA project and will involve both European and Chinese partners.
Because it draws on the existing technology and fast moving grid expertise, the VEGA project will keep a close attention to other tools such as the Virtual Workspaces proposed for Globus. It will also closely follow the ongoing technology and it will use the WSRF and GT4 as a basis for its implementation. However, the impact of cooperation with China leads also to guarantee its compatibility with Chinese projects like ChinaGrid and the European Unicore and gLite middleware.
International forums and standardization bodies like GGF and EGEE users groups will be invested for close discussions on this fundamental topic.
Because interesting research has been carried out on the virtualization concepts, an innovative and fertilizing approach is followed concerning the Virtual Environments proposed.
Indeed, Grid Portals, Grid Application Toolkits, Virtual Workspaces, Dynamic Virtual Environments, Virtual Organizations, Runtime Environments, virtual users accounts, on demand computing and virtual data centers and clusters have been proposed recently in various projects to support the access to the Grid. Much of these important studies and tools have been aimed at single sign-on, authentication, authorization, end-to-end QoS support, and, last but not least, at facilitating the uptake of grid technology by the users. These fundamental bricks are currently being developed, deployed, tested and sometimes marketed on existing infrastructures and middleware.
There still remains however niches that are yet unexplored. The VEGA project proposes to invest a new functional enabling layer that will be deployed on top of existing middleware and their extensions in order to increase the ease of access and manipulation of applications on the Grid. In order to achieve this, the applications designers and the users require high-level services to define the applications easily, based on existing runtime components, on already deployed applications in order to build, deploy, run and monitor new applications possibly based on existing software, on already tested components and services, and to incrementally construct and upgrade them.
This will be based on new services compliant with WSRF that will be supported by a specific software layer on top of existing middleware. This new software layer, called the “upperware”, will be implemented on gLite.

References

Denemark, J, et al. User Management for Virtual Organizations. CoreGRID Technical Report TR-0012. Novmber 2005.
Keahey K., Doering K., Foster I. From sandbox to playground: dynamic virtual environments in the grid. Proc. 5th IEEE/ACM Int’l Workshop on Grid Computing. GRID’04. Pittsburgh (USA). November 2004.
Keahey K., Ripeanu M. , Doeirng K. Dynamic creation and management of Runtime Environments on he Grid. Workshop on Designing and Building Web Servces GGF9.

Chicago (USA). October 2003.
Kehaey K., et al. Virtual Workspaces in the Grid. Europar 2005. Pisa (I). August 2005.
Foster I. , et al. Modeling stateful resources with Web Services. Versin 1.1. March 2004.
Jankowski M., Wolniewicz P., Meyer N. Virtual user System for Globus based grids. Proc. Cracow Grid Workshop December 2004.
Chase J., et al. Dynamic virtual clusters in a grid site manager. Proc. 12th Int'l Symposium on High-performance Distributed computing. 2003.
Chuang Lin, Zhiguang Shan. Grid research in China: origins and directions. Proc. 10th IEEE Int'l Workshop on Future Trends in Distributed Computing Systems. 2004.
Laszewski L., et al. Commodity Grid Kits: Middleware for building grid computing environments. Chapter in "Making the Global infrastructure a Reality". Wiley Eds. 2003.
Lorch M., et al. Authorization and account management in the Open Science Grid. Proc. IEEE Grid Computng Workshop. 2005.
NESSI. Networked European Software and Services Initiative. Vision Document. Version 1.2b. May 2005.
Davis K. Grid Application Toolkit Canonical Implementation and user's guide. GridLab project. CGUG-0013-1.0
Nguyen, G.T., Wang, L., Antikidis, J.P. "Virtual environments for spatial data infrastructures on computing grids". ISPRS Workshop on "Service and Application of Spatial Data Infrastructures". Hangzhou (China). October 2005.
Nguyen G.T., L. Wang, V. Selmin. Virtual environments for multiphysics code validation on computing grids. International East West High Speed Flow Field Conference 2005. Beijing (China). October 2005.
Nguyen G.T., L. Wang. Virtual environments for multiphysics applications on computing grids. Invited lecture. AFFRST Seminar « Numerical Analysis and Scientific Computing with PDES and their Challenging Applications ». Center for Scientific Computing. Helsinki (Finland). October 2005.
Nguyen G.T., L. Wang. Virtual application environments on computing grids. Third International Conference on Computing, Communications and Control Technologies (CCCT'05), Austin, Texas (USA). July 2005.
Nguyen G.T., J.P. Antikidis. Virtual computing environments for problem-solving on grids. International Parallel CFD Conference. Universita de Las Palmas. Gran Canaria (Spain). May 2004.
Nguyen G.T. Grid-computing in Multidisciplinary CFD optimization problems. International Parallel CFD Conference. Invited lecture. Russian Academy of Sciences. Institute of Mathematical Modeling. Moscow (Russia). May 2003.

2b: Data access on the grid / 93

Encrypted Data Storage in EGEE

Author: Ákos Frohner[1]

Co-authors: Christophe Pera [2]; Daniel Jouvenot [3]; Johan Montagnat [4]; Patrick Guio [5]; Péter Kusnzt [1]; Ricardo Brito Da Rocha [1]; Zoltán Farkas [6]

[1] *CERN*

[2] *CNRS, CREATIS laboratory*

[3] *LAL laboratory*

[4] *CNRS, I3S laboratory*

[5] *University of Bergen*

[6] *MTA-Sztaki LPDS*

Corresponding Author: akos.frohner@cern.ch

The medical community is routinely using clinical images and associated medical data for diagnosis, intervention planning and therapy follow-up. Medical imaging is producing an increasing number of digital images for which computerized archiving, processing and analysis are needed.

Grids are promising infrastructures for managing and analyzing the huge medical databases. Given the sensitive nature of medical images, practiotionners are often reluctant to use distributed systems though. Security if often implemented by isolating the imaging network from the outside world inside hospitals. Given the wide scale distribution of grid infrastructures and their multiple administrative entities, the level of security for manipulating medical data should be particularly high.

In this presentation we describe the architecture of a solution, the gLite Encrypted Data Storage (EDS), which was developed in the framework of Enabling Grids for E-sciencE (EGEE), a project of the European Commission (contract number INFSO−508833). The EDS does enforce strict access control to any medical file stored on the grid. It also provides file encryption facilities, that ensure the protection of data sent to remote storage, even from their administrator. Thus, data are not only transferred but also stored encrypted and can only be decrypted in host memory by authorized users.

Introduction

The basic building blocks of the grid data management architecture are the Storage Elements (SE), which provide transport (e.g. gridftp), direct data access (e.g. direct file access, rfio, dcap) and administrative (Storage Resource Management, SRM) interfaces for a storage system. However the most widely adopted standard today for managing medical data in clinics is DICOM (Digital Image and COmmunication in Medicine).

The simplified goal is to secure the data movement among these blocks, and the client hosts, which actually process the data.

Challenges

Here we describe the most important challenges and requirements of the medical community and how they are addressed by EDS on the current grid infrastructure.

Access Control

The most basic requirement is to restrict the access to any data, which is on the grid, to permitted users. Although it looks like a simple requirement, the distributed nature of the architecture and the limitations of the building blocks required some work to satisfy the requirements.

The first problem faced is the complex access patterns of the medical community. It is usually not enough to define a single user or group which is allowed to access the file, but instead access is needed by a list of users. The solution is to use Access Control Lists (ACLs), instead of basic POSIX permission bits, however most of the currently deployed Storage Elements do not provide ACLs.

To solve the semantical mismatch, we “wrapped” the existing Storage Elements into a service, which enforced the access control settings, according to the medical community's requirements. This service is called the gLite I/O server, which is installed beside every used storage element.

The gLite I/O server provides a POSIX like file access interface to remote clients, and uses the direct data access methods of the Storage Element to access the data. It authenticates the clients and enforces authorization decisions (i.e. if the client is allowed to read a file or not), so it acts like a Policy Enforcement Point in the middle of the data access.

The authorization decision is not made inside the gLite I/O server. A separate service holds the ACLs (and other file metadata) of every file stored in the Storage Elements. In our deployment it was the gLite File and Replica Management (FiReMan) service, which acts like a Policy Decision Point in the architecture.

The gLite FiReMan service is a central component, which also acts like a file catalog (directory functionality), replica manager (which file has a copy on a given SE) and file authorization server (if a given client is allowed to access a file). The gLite FiReMan service supports rich ACL semantics, which satisfy the access pattern requirements of the medical community.

Encryption

The other important requirement is privacy: the sensitive medical data shall not be stored on any permanent storage or transferred over the network unencrypted, outside the originating hospital.

The solution is to encrypt every file, when it leaves the originating hospital's DICOM server, and decrypt it only inside the authorized client applications.

For the first step we developed a specialized Storage Element, the Medical Data Manager (MDM) service, which "wraps" the hospital's DICOM server and offers interfaces, which are compatible with other grid Storage Elements. In this way the hospital's data storage will look like just another Storage Element, for which we already have grid data managements solutions.

Despite the apparent similarity between the MDM service and an ordinary Storage Element there is an important difference: the MDM service serves only encrypted files. When a file is accessed through the grid interfaces, the service generates a new encryption key, encrypts the file and registers the key in a key store. Therefore every file which crosses the external network and is stored on stored on an external element stays encrypted during its whole lifetime.

On the client side we provided a transparent solution to decrypt the file: on top of the gLite I/O client libraries, we developed a client library, which can retrieve keys from t he key storage and decrypt files on the fly. The client side library provides a POSIX like interface, which hides the details of the remote data access, key retrieval and decryption.

The key storage had to satisfy several requirements: it has to be reliable, secure and provide fine grained access control for the keys.

To satisfy these requirements we developed the gLite Hydra KeyStore. To satisfy reliability the keys are not only stored at one place, but at least at two locations. To satisfy security, one service cannot store a full key, but only a part of it, thus even when the service is compromised the keys cannot be fully recovered. We implemented Shamir's Secret Sharing Scheme inside the client library to split and distribute the keys among at least three Hydra services, according to the above mentioned requirements.

The key storage also has to provide fine grained access control, similar to the files, on the keys. Our current solution actually applies the same ACLs as the FiReMan service, thus one can be sure that only those who can access the encryption key of a file are allowed to access the file itself.

Conclusion

The solution for encrypted storage described above has been already released in the gLite software stack and been deployed and demonstrated to work at a number of sites.

As the underlying software stack of the grid evolves we will also adapt our solution to exploit new functionality and to simplify our additional security layer.

Summary:

The medical community is routinely using clinical images and associated medical data for diagnosis, intervention planning and therapy follow-up. Medical imaging is producing an increasing number of digital images for which computerized archiving, processing and analysis are needed.

Grids are promising infrastructures for managing and analyzing the huge medical databases. Given the sensitive nature of medical images, practiotionners are often reluctant to use distributed systems though. Security if often implemented by isolating the imaging network from the outside world inside hospitals. Given the wide scale distribution of grid infrastructures and their multiple administrative entities, the level of security for manipulating medical data should be particularly high.

In this presentation we describe the architecture of a solution, the gLite Encrypted Data Storage (EDS), which was developed in the framework of Enabling Grids for E-sciencE (EGEE), a project of the European Commission (contract number INFSO–508833). The EDS does enforce strict access control to any medical file stored on the grid. It also provides file encryption facilities, that ensure the protection of data sent to remote storage, even from their administrator. Thus, data are not only transferred but also stored encrypted and can only be decrypted in host memory by authorized users.

2b: Data access on the grid / 94

The gLite File Transfer Service

Author: Gavin Mccance[1]

[1] *CERN*

In this paper we describe the architecture and implementation of the gLite File Transfer Service (FTS) and list the most basic deployment scenarios. The FTS is addressing the need to manage massive wide-area data transfers on dedicated network channels while allowing the involved sites and users to manage their policies. The FTS manages the transfers in a robust way, allowing for an optimized high throughput between storage systems.

The FTS can be used to perform the LHC Tier-0 to Tier-1 data transfer as well as the Tier-1 to Tier-2 data distribution and collection. The storage system peculiarities can be taken into account by fine-tuning the parameters of the FTS managing a particular channel. All the manageability related features as well as the interaction with other components that form part of the overall service are described as well. The FTS is also extensible so that particular user groups or experiment frameworks can customize its behavior both for pre- and post-transfer tasks.

The FTS has been designed based on the experience gathered from the Radiant service used in Service Challenge 2, as well as the CMS Phedex transfer service. The first implementation of the FTS was put to use in the beginning of the Summer 2005. We report in detail on the features that have been requested following this initial usage and the needs that the new features address. Most of these have already been implemented or are in the process of being finalized. There has been a need to improve the manageability aspect of the service in terms of supporting site and VO policies.

Due to different implementations of specific Storage systems, the choice between 3rd party gsiftp transfers and SRM-copy transfers is nontrivial and was requested as a configurable option for selected transfer channels. The way the proxy certificates are being delegated to the service and are used to perform the transfer, as well as how proxy renewal is done has been completely reworked based on experience. A new interface has been added to enable administrators to perform management directly by contacting the FTS, without the need to restart the service. Another new interface has been added in order to deliver statistics and reports to the sites and VOs interested in useful monitoring information. This is also presented through a web interface using javascript. Stage pool handling for the FTS is being added in order to allow pre-staging of sources without blocking transfer slots on the source and also to allow the implementation of back-off strategies in case the remote

staging
areas start to fill up.

The reliable transport of data is one of the cornerstones for distributed systems. The transport mechanisms have to be scalable and efficient, making optimal usage of the available network and storage bandwidth. In production grids the most important requirement is robustness, meaning that the service
needs to be run over extended periods of time with little supervision. Moreover, the transfer middleware has to be able to apply policies for failure, adapting parameters dynamically or raising alerts where necessary. In
large Grids, we have the additional complication of having to support multiple
administrative domains while enforcing local site policies. At the same time,
the Grid application needs to be given uniform interface semantics independent
of site-local policies.

There are several file transfer mechanisms in use today in Data Grids, like http(s), (s)ftp , scp or bbftp, but probably the most commonly used one is GridFTP, providing a highly performant secure transfer service. The Storage Resource Manager SRM interface, which is being standardized through the Global
Grid Forum, provides a common way to interact with a Storage Element, as well
as a data movement facility, called SRM copy, which in most implementations will again make use of GridFTP to perform the transfer on the user's behalf between two sites.

The File Transfer Service is the low level point to point file movement service provided by the EU-funded Enabling Grids for E-SciencE (EGEE) project's
gLite middleware. It has been designed in order to address the challenging requirements of a reliable file transfer service in production Grid environments. What distinguishes the FTS from other reliable transfer services
is its design for policy management. The FTS can also act as the resource manager's policy enforcement tool for a dedicated network link between two sites as it is capable of managing the policies of the resource owner as well
as of the users (the VOs). The FTS has dedicated interfaces to manage these policies. The FTS is also extensible; upon certain events user-definable functions can be executed. The VOs may make use of this extensibility point to
call upon other services when transfers complete (e.g. register replicas in catalogs) or to change the policies for certain error handling operations (e.g. the retry strategy).

The LHC Computing Project (LCG) is the project that has built and maintains a
data storage and analysis infrastructure for the entire high energy physics community of the Large Hadron Collider (LHC), the largest scientific instrument on the planet located at CERN. The data from the LHC experiments will be distributed around the globe, according to a multi-tiered model, where
CERN is the "Tier-0", the centre of LCG.
The goal of LCG Service Challenges is to provide a production quality environment where services are run for long periods with 24/7 operational support. These services include the Network and Reliable File Transfer services. In Summer 2005 Service Challenge 3 started with gLite File Transfer

Service and CMS Phedex. The gLite FTS benefited from this collaboration and
from the experience of prototype LCG Radiant Service, used in Service
Challenge 2. This meant that from the beginning its design took into
account
all the requirements imposed by a production Grid infrastructure. The
continuous interaction with the experiments was useful in order to react
quickly to reported problems, as well as to keep the development focused on
real use cases.

Summary:

The gLite File Transfer Service

1b: Astrophysics/Astroparticle physics - Fusion - High-Energy physics / 95

CRAB: a tool for CMS distributed analysis in grid environment.

Authors: Alessandra Fanfani[1]; Daniele Spiga[2]; Federica Fanzago[3]; Marco Corvo[4]; Stefano Lacaprara[5]

Co-authors: Giovanni Ciraolo [6]; Nicola De Filippis [7]; Nikolai Smirnov [3]

[1] *Bologna University*

[2] *Perugia University*

[3] *INFN-PADOVA*

[4] *CERN/CNAF*

[5] *INFN Legnaro*

[6] *Firenze University*

[7] *Bari University*

Corresponding Author: fanzago@pd.infn.it

The CMS experiment will produce a large amount of data (few PBytes each year) that will be distributed and stored in many computing centres spread in the countries participating to the CMS collaboration and made available for analysis to world-wide distributed physicists.
CMS will use a distributed architecture based on grid infrastructure to analyze data stored at remote sites, to assure data access only to authorized users and to ensure remote resources availability.
Data analisys in a distributed environment is a complex computing task, that assume to know which data are available, where data are stored and how to access them.
The CMS collaboration is developing a user friendly tool, CRAB (Cms Remote Analysis Builder), whose aim is to simplify the work of final users to create and to submit analysis jobs into the grid environment. Its purpose is to allow generic users, without specific knowledge of grid infrastructure, to access and analyze remote data as easily as in a local environment, hiding the complexity of distributed computational services.
Users have to develop their analisys code in an interactive environment and decide which data to analyze, providing to CRAB data parameters (keywords to select data and total number of events) and how to manage produced output (return file to UI or store into remote storage).
CRAB creates a wrapper of the analisys executable which will be run on remote resources, including CMS environment setup and output management. CRAB splits the analisys into a number of jobs according to user provided information about number of events. The job submission is done using grid workload management command.
The user executable is sent to remote resource via inputsandbox, together with the job. Data discovery, resources availability, status monitoring and output retrieval of submitted jobs are fully handled by CRAB.
The tool is written in python and have to be installed to the User Interface, the

user access point to the grid.
Up to now CRAB is installed in ~45 UI and about ~210 different kind of data are available in ~40 remote sites.
The weekly rate of submitted jobs is ~10000 with a success rate about 75%, that means jobs arrive to remote sites and produce outputs, while the remnant 25% aborts due to site setup problem or grid services failure.
In this report we will explain how CRAB is interfaced with other CMS/grid services and will report the daily user's experience with this tool analyzing simulated data needed to prepare the Physics Technical Design Report.

Summary:

Report about CRAB, a tool for CMS analysis in grid environment: how it is interfaced with CMS/grid services and user's experience.

1a: Life Sciences / 96

A service to update and replicate biological databases

Author: Jean Salzemann[1]

[1] *IN2P3/CNRS*

Corresponding Author: jean.salzemann@cern.ch

One of the main challenges in molecular biology is the management of data and databases. A large fraction of the biological data produced is publicly available on web sites or by ftp protocols. These public databases are internationally known and play a key role in the majority of public and private research. But their exponential
growth raises an usage problem. Indeed, scientists need easy access to the last update of the databases in order to apply bioinformatics or data mining algorithms. The frequent and regular update of the databases is a recurrent issue for all host or
mirror centres, and also for scientists using the databases locally for confidentiality reasons.

We proposed a solution for the updates of these distributed databases. This solution come as a service embedded into the grid which uses its mechanisms and automatically performs updates. So we developed a set of web services that will rely on the grid to
manage this task, with the aim of deploying the services under any grid middleware with a minimum of adaptation. This includes a client/server application with a set of
rules and a protocol to update a database from a given repository and distribute the update through the grid storage elements while trying to optimize network bandwidth, file transfers size and fault tolerance, and finally offer a transparent automated service which does not require user intervention. This represents the challenges of the database update in a grid environment and the solution we proposed is basically to define two types of storage on the grid storage elements: some storage of reference where the update is first performed and working storage spaces where the jobs will pick up the information. The idea is to replicate the update on the grid from these reference points to the storage elements. From the service point of view, it is necessary that the grid information system can locate sites who host a given database in order to have the benefits of a dynamical database replication and location. From the user point of view, we need to dispose of the location information
for each database in order to achieve scalability and find replica on the grid, this

means having a metadata for each database that can refer to several physical
locations on the grid and contain certain information as well, because the replica
do
not concern single files but a whole database with several files and/or
directories.

This service is being deployed on two French Grid infrastructures: RUGBI (based on Globus Toolkit 4) and Auvergrid (based on EGEE), so we plan a future deployment of this service on EGEE, especially in the Biomed VO, but the real issues are that the service need to be deployed as a grid service, and managed as a grid service, so some
people from the VO should be able to deploy and administrate the service beside the site administrators, a role which is finding its limits in current VO management. The
service is supposed to be embedded into the grid and is not just a pure application laid on it. Eventually it will be possible to offer this service as an application, but it would mean that its use is not mandatory and not automated, which is synonymous with losing its benefits and transparency since the user will need to specify the use of the service in his workflow. There are also future plans to add some optimisation on the deployment of the databases: for example, being able to split databases to store each part on a different storage element, or add the ability
to offer several reference storages per database which would require to synchronize these storages with each other. The service will mature through its deployment on grid middlewares and will surely improve as it is used in production environments.

2b: Data access on the grid / 97

Use of the Storage Resource Manager Interface

Author: Maarten Litmaath[1]

[1] *CERN*

SRM v2.1 features and status

Version 2.1 of the Storage Resource Manager interface offers various features that are desired by EGEE VOs, particularly HEP experiments: pinning and unpinning of files, relative paths, (VOMS) ACL support, directory operations, global space reservation. The features are described in the context of actual use cases and availability in the following widely used SRM implementations: CASTOR, dCache, DPM. The interoperability of the different implementations and SRM versions is discussed, along with the absence of desirable features like quotas.

Version 1.1 of the SRM standard is in widespread use, but has various deficiencies that are addressed to a certain extent by version 2.1. The two versions are incompatible, necessitating clients and servers to maintain both interfaces, at least for a while. Certain problems will only be dealt with in version 3, whose definition may not be completed for many months. There are various implementations of versions 1 and 2, developed by different collaborations for different user communities and service providers, with different requirements and priorities. In general a VO will have inhomogeneous storage resources, but a common SRM standard should make them compatible, such that data management tools and procedures need not bother with the actual types of the storage facilities.

Poster and Demo session + cocktail / 98

Replication on the AMGA Metadata Catalogue

Author: Nuno Filipe De Sousa Santos[1]

Co-author: Birger Koblitz [2]

[1] *Universidade de Coimbra*

[2] *CERN*

Corresponding Author: nuno.santos@cern.ch

1. Introduction

Metadata Services play a vital role on Data Grids, primarily as a means of describing and discovering data stored on files but also as a simplified database service. They must, therefore, be accessible to the entire Grid, comprising several thousands of users spread across hundreds of Grid sites geographically distributed. This means they must scale with the number of users, with the amount of data stored and also with geographical distribution, since users in remote locations should have low-latency access to the service. Metadata Services must also be fault-tolerant to ensure high-availability.

To satisfy such requirements, Metadata Services must offer flexible replication and distribution mechanisms especially designed for the Grid environment. They must cope with the heterogeneity and dynamism of a Grid, as well as the typical workloads.

To address these requirements, we are building replication and federation mechanisms into AMGA, the gLite Metadata catalogue. These mechanisms work at the middleware level, providing database independent replication, especially suited for heterogeneous Grids. We use asynchronous replication for scalability on wide-area networks and improved fault-tolerance. Updates are supported on the primary copy, with replicas being read-only. For flexibility, AMGA supports partial replication and federation of independent catalogues, allowing applications to tailor the replication mechanisms to their specific needs.

1. Use Cases

Replication on AMGA is designed to cover a broad range of usage scenarios that are typical of the main user communities of EGEE.

High Energy Physics (HEP) applications are characterised by large amounts of read-only metadata, produced on a single location and accessed by hundreds of physicists spread across many remote sites. By using AMGA replication mechanisms, remote Grid sites can create local replicas of the metadata they require, either of the whole metadata tree or of parts of it. Users at remote sites will experience a much improved performance by accessing a local replica.

For Biomed applications the main concern with metadata is ensuring its security, as it often contains sensitive information about patients that must be protected from unauthorised users. This task is made more difficult by the existence of many grid sites producing metadata, that is, the different hospitals and laboratories where it is generated. Creating copies on remote sites increases the security risk and, therefore, should be avoided. AMGA replication allows the federation of these Grids sites into a single virtual distributed metadata catalogue. Data is kept securely on the site it was generated, but users can access it transparently from any AMGA instance, which discovers where the data is located and redirects the request to that AMGA instance, where it will be executed after the user credentials have been validated.

We believe that partial replication and federation as they are being implemented in AMGA provides the necessary building blocks for the distribution needs of many other applications, while at the same time offering scalability and fault-tolerance.

1. Current Status and Future Work

We have implemented a prototype of the replication mechanisms of AMGA, which is currently undergoing internal testing. Soon we will be ready to start working with the interested communities, with the goal of better evaluating our ideas and of obtaining user feedback to guide us through further development of the replication mechanisms.

A clear user requirement that we will study is the dependability of the system, including mechanisms for detecting failures of replicas and for recovering from those failures. If the failure is on a replica, clients should be redirected transparently to a different replica. If the failure is on the primary copy, then the remaining replicas should elect a new primary copy among themselves. All these mechanisms need an underlying discovery system to allow replicas to locate and query each other, as well as mechanisms for running distributed algorithms among the nodes of the system.

1c: Earth Observation - Archaeology - Digital Library / 99

Worldwide ozone distribution by using Grid infrastructure

Author: Monique Petitdidier[1]

Co-authors: Horst Swichtenberg [2]; Luigi Fusco [3]; Michele Iapaolo [4]; Wim Som de Cerff [5]

[1] *IPSL*

[2] *FHG/SCAI*

[3] *ESRIN/ESA*

[4] *Tor Vergata University*

[5] *KNMI*

Corresponding Author: monique.petitdidier@cetp.ipsl.fr

ESRIN : L. Fusco, J. Linford, C. Retscher
IPSL : C. Boonne, S. Godin-Beekmann, M. Petitdidier, D. Weissenbach
KNMI: W. Som de Cerff
SCAI-FHG: J. Kraus, H. Schwichtenberg
UTV : F. Del Frate, M. Iapaolo

Satellite data processing presents a challenge for any computer resources due to the large volume of data and number of files. The vast amount of data sets and databases are all distributed among different countries and organizations. The investigation of such data is limited to some sub-sets. As a matter of fact, all those data cannot be explored completely due on one hand to the limitation in local computer and storage power, and on the other hand to the lack of tools adapted to handle, control and analyse efficiently so large sets of data.
In order to check the capability of a Grid infrastructure to fill those requirements, an application based on ozone measurements was designed to be ported first on DataGrid, then on EGEE and local Grid in ESRIN.
The satellite data are provided by the experiment, GOME aboard the satellite ERS. From the ozone vertical total content, ozone profiles have been retrieved by using two different algorithm schemas, one is based on an inversion protocol (KNMI), the other on a neural network approach (UTV). The porting on DataGrid was successful however some functionalities are missing to make the application operational. In EGEE, the reliability of the infrastructure has been as reliable as a local Grid.
The second part of the application has been the validation of those satellite ozone

profiles by profiles measured by ground-based lidars. The goal was to find out collocated observations meta databases were built to solve this problem. The result has been the production of the 7 years of data on EGEE and on local Grid at ESRIN with two versions of the Neural Network algorithm and several months by the inversion algorithm. It is an amount of around 100 000 files registered on EGEE.
Then, the validation of this set of data was carried out by using all the lidar profiles available in the NDSC databases (Network Detection of Stratospheric Changes). To find collocation data an OGSA-DAI metadata server has been implemented and geospatial queries permit to search the orbit passing over the lidar site.
The second work, started during DataGrid, has been the development of a portal, specific to the Ozone application, described above, and extended latter to other satellite data like Meris…The role of this portal is to provide an operational way to a friendly end-use of Grid infrastructure. It provides the missing functionalities of the Grid infrastructure.
EGEE offers the possibility to store all the ozone data obtained by satellite experiment (GOME, GOMOS, MIPAS…) as well as ground-based network of lidars and radiosoundings…The next goal on the way is to be able to find out at a given location and/or at a given time the distribution of ozone by combining all the existing databases.
In this presentation, the scientific and operational interest will be pointed out.

2c: Special type of jobs (MPI, SDJ, interactive jobs, …) - Information systems / 100

Efficient job handling in the GRID: short deadline, interactivity, fault tolerance and parallelism

Author: Jakub MOSCICKI[1]

[1] *CERN*

Corresponding Author: jakub.moscicki@cern.ch

The major GRID infastructures are designed mainly for batch-oriented computing with coarse-grained jobs and relatively high job turnaround time. However many practical applications in natural and physical sciences may be easily parallelized and run as a set of smaller tasks which require little or no synchronization and which may be scheduled in a more efficient way. The Distributed Analysis Environment Framework (DIANE), is a Master-Worker execution skeleton for applications, which complements the GRID middleware stack. Automatic failure recovery and task dispatching policies enable an easy customization of the behaviour of the framework in a dynamic and non-reliable computing environment. We demonstrate the experience of using the framework with several diverse real-life applications, including Monte Carlo Simulation, Physics Data Analysis and Biotechnology.

The interfacing of existing sequential applications from the point of view of non-expert user is made easy, also for legacy applications. We analyze the runtime efficiency and load balancing of the parallel tasks in various configurations and diverse computing environments: GRIDs (LCG, Crossgrid), batch farms and dedicated clusters. In practice, the usage of ther Master/Worker layer allows to dramatically reduce the job turnaround time, a scenario suitable for short deadline jobs and interactive data analysis.

Finally it is also possible to easily introduce more complex synchronization patterns, beyond trivial parallelism, such as arbitrary dependency graphs (including cycles, in contrast to DAGs) which may be suitable for bio-informatics applications.

2d: VO tools - Portals / 101

User and virtual organisation support in EGEE

Authors: Alistair Mills[1]; Flavia Donno[1]; Ian Bird[1]; Torsten Antoni[2]

[1] *CERN*

[2] *FZK*

Corresponding Authors: flavia.donno@cern.ch, alistair.mills@cern.ch

User and virtual organisation support in EGEE
Providing adequate user support in a grid environment is a very challenging task due to the distributed nature of the grid. The variety of users and the variety of Virtual Organizations (VO) with a wide range of applications in use add further to the challenge.
The people asking for support are of various kinds. They can be generic grid beginners, users belonging to a given Virtual Organization and dealing with a specific set of applications, site administrators operating grid services and local computing infrastructures, grid monitoring operators who check the status of the grid and need to contact the specific site to report problems; to this list can be added network specialists and others.
Wherever a user is located and whatever the problem experienced is, a user expects from a support infrastructure a given set of services. A non-exhaustive list is the following:
a) a single access point for support;
b) a portal with a well structured sources of information and updated documentation concerning the VO or the set of services involved;
c) experts knowledgeable of the particular application in use and who can even discuss with the user to better understand what he/she is trying to achieve (hot-line); help integrating user applications with the grid middleware;
d) correct, complete and responsive support;
e) tools to help resolve problems (search engines, monitoring applications, resources status, etc.);
f) examples, templates, specific distributions for software of interest;
g) integrated interface with other Grid infrastructure support systems;
h) connection with the grid developers and the deployment and operation teams;
i) assistance during production use of the grid infrastructure.
With the Global Grid User Support (GGUS) infrastructure, EGEE attempts to meet all of these expectations. The current use of the system and the user satisfaction ratings have shown that the goal has been achieved with a certain success for the moment.
As of today GGUS has shown to be able to process up to 200 requests per day and provides all above listed services. In what follows we discuss the organization of the GGUS system, how it meets the users'needs, and the current open issues.
The model of the existing EGEE Global Grid User Support (GGUS) is as follows. The support model in EGEE can be captioned "regional support with central coordination". Users can submit a support request to the central GGUS service, or to their Regional Operations' Center (ROC) or to their Virtual Organisation (VO) helpdesks.
Within GGUS there is an internal support structure for all support requests. The ROCs and VOs and the other project wide groups such as middleware groups (JRA), network groups (NA), service groups (SA) and other grid infrastructures (OSG, NorduGrid, etc.) are connected via a central integration platform provided by GGUS. GGUS central helpdesk also acts as a portal for all users who do not know where to send their requests. They can enter them directly into the GGUS system via a web form or e-mail.
This central helpdesk keeps track of all service requests and assigns them to the appropriate support groups. In this way, formal communication between all support groups is possible. To enable this, each group has built an interface (e-mail and web front-end, or interface between ticketing systems) between its internal support structure and the central GGUS application.
In the central GGUS system, first line support experts from the ROCs and the Virtual Organizations will do the initial problem analysis. Support is widely

distributed. These experts are called Ticket Processing Managers (TPM) for generic first line support (generic TPM) and for VO specific first line support (VO TPM). These experts can either provide the solution to the problem reported or escalate it to more specialized support unit that provide network, middleware and grid service support. They may also refer it to specific ROCs or VO experts.
Behind the specialized VO TPM support units, people belonging to EGEE/NA4 groups such as the Experiment Integration Support group (EIS) help VO users with on-line support and the integration of the VO specific applications with the grid middleware. Such people can also recognize if a problem is application specific and forward the problem to more VO specific support units connected to GGUS.
TPM and VO TPMs have also the duty of following tickets, making sure that users receive an adequate answer, coordinating the effort of understanding the real nature of the problem and involving more than one second level support unit if needed. The following figure depicts the ticket flow.
To provide appropriate user support, the distributed structure of EGEE and the VOs has to be taken into account. The community of supporters is therefore distributed. Their effort is coordinated centrally by GGUS and locally by the local ROC support infrastructures.
The ROC provides adequate support to classify the problems and to resolve them if possible. Each ROC has named user support contacts who manage the support inside the ROC and who coordinate with the other ROCs'support contacts. The classification at this level distinguishes between operational problems, configuration problems, violations of service agreements, problems that originate from the resource centres and problems that originate from global services or from internal problems in the software. Problems that are positively linked to a resource centre are then transferred to the responsibility of the ROC with which the RC is associated.
MEETING USER NEEDS
As explained above, GGUS provides therefore a single entry point for reporting problems and dealing with the grid. In collaboration with the EGEE EIS team, the EGEE User Information Group, NA3, and the entire EGEE infrastructure, GGUS offers a portal where users can find up-to-date documentation, and powerful search engines to find answers to resolved problems and examples. Common solutions are stored in the GGUS knowledge database and Wiki pages are compiled for frequent or undocumented problems/features.
GGUS offers hot lines for users and supporters and a VRVS chat room to make the entire support infrastructure available on-line to users.
Special tools and grid middleware distributions are made available by the NA4/EIS team for GGUS users.
GGUS is interfaced with other grids'support infrastructures such as in the case of OSG and NorduGrid. Also, GGUS is used for daily operations to monitor the grid and keep it healthy. Therefore, specific user problems can be directly communicated to the Grid Operation Centers and broadcasted to the entire grid community.
GGUS is used also to follow and track down problems during stress testing activities such as the HEP experiments production data challenges and the service challenges.
OPEN ISSUES
Even-though GGUS has proven to provide useful services, there are still many things that need improvement. Concerning users and VOs, in particular, we have identified the following:
Small VOs do not have the resources to implement their part of the model
The large VOs such as the LHC experiments have people who provide support for the applications which the VO has to run as part of its work. These people are contacted by GGUS when tickets are assigned to the VO or then the problem needs immediate or on-line attention. It has proven difficult for some of the small VOs to provide such a service. In this case, GGUS still provides support for the VO, but if the problem is application related and cannot be resolved, then it has to be put into the state 'unsolvable'.
Supporters have other jobs to do
In EGEE, almost everyone providing support does so as part of their job. It is not usually a major part of their job. Some times it is difficult to ensure responsiveness. There is a small team which maintains and develops the GGUS system.
Supporters are concentrated in a few locations
The resources of the grid are widely distributed over 180 locations, and there are people in all of these locations looking after the basic operation of the

computers. However this is not the case for higher level support such as support for a VO application. This tends to exist in only a small number of locations, with a small number of supporters.
Scalability is constrained by the availability of supporters
The number of people who can provide support for basic operations is large, but the number of people who can provide support for higher level services is small. As the VOs become larger this will become a constraint to growth unless more supporters are found.
Limited experience in handling a large number of tickets
As part of the development of the GGUS system, it has been exercised by generating tickets. As the system is built from industry standard software parts using Remedy and Oracle, it has been found to be reliable. We believe however that if large numbers of tickets are submitted that it will show the limitations in the system.
Limited engagement of existing VOs in the implementation of GGUS
There is an organisation within EGEE called Executive Support Committee (ESC). The ESC has representatives from all of the ROCs of EGEE. This organisation meets once per month by telephone to discuss the operations and development of the support system and to decide on actions and priorities for the work. The present VOs have found it difficult to provide people for involvement with this work.
CONCLUSION
The GGUS system is now ready for duty. During 2006, it is expected that there will be a large number of tickets passing through the system as the LHC VOs move from preparing for service to being in production. It is also expected that the number of Virtual Organisations will grow as the work of EGEE-II proceeds. There will also be an increase in the number of support units involved with GGUS, and an increase in the number of ROCs and RCs.
Acronyms
EGEE Enabling Grids for E-sciencE
EIS Experiment Integration Support
GGUS Global Grid User Support
HEP High Energy Physics
JRA Joint Research Activity of EGEE
LHC Large Hadron Collider
NA Network Activity
OSG Open Science Grid
RC Resource Centre
ROC Regional Operations' Centre
SA Service Activity
TPM Ticket Process Management
VO Virtual Organisation
VRVS Virtual Rooms Videoconferencing System
Wiki Web technology for collaborative working

Poster and Demo session + cocktail / 102

CMS Dashboard of Grid Activity

Authors: Juha HERRALA[1]; Julia ANDREEVA[1]

[1] *CERN*

Corresponding Author: juha.herrala@cern.ch

The CMS Dashboard project aims to provide a single entry point to the monitoring data collected from the CMS distributed computing system. The monitoring information collected in the CMS dashboard allows to follow the processing of the CMS jobs on the LCG, EGEE and OSG grid infrastructures. The Dashboard supports tracing of the job execution failures on the Grid and erros due to problems with the experiment-specific applications. In addition the Dashboard is able to present an estimation of the I/O rates between the worker nodes and data storage and helps keeping record of the sharing of the resources between production and analysis groups and different users.

One of the final goals is to discover inefficiencies in the data distribution and problems in the data publishing.

The Dashboard data base combines the Grid-specific data from the Logging and Book-keeping system via RGMA and the CMS-specific data via Monalisa monitoring system. Web interface to the dashboard data base provides access to the monitoring data in the interactive mode and through the set of the predefined views. The interactive mode enables the possibility to get information in a detailed level, which is very important for tracking of various problems.

2d: VO tools - Portals / 103

ETICS: eInfrastructure for Testing, Integration and Configuration of Software

Author: Marc-Elian Begin[1]

Co-author: Alberto Di Meglio [1]

[1] *CERN*

A broad range of projects from a spectrum of disciplines involve the development of software born from the collaborative efforts of partners from geographically spread locations. Such software is often the product of large-scale initiatives as new technological models like the Grid are developed and new e-Infrastructures are deployed to help solve complex, computational-intensive problems.

Recent experience in such projects has shown that the software products often risk suffering from lack of coherence and quality. Among the causes of this problem we find the large variety of tools, languages, platforms, processes and working habits employed by the partners of the projects. In addition, the issue of available funding for maintenance and support of software after the initial development phase in typical research projects often prevents the developed software tools from reaching production-level quality. Establishing a dedicated build and test infrastructure for each new project is inefficient, costly and time-consuming and requires specialized resources, both human and material, that are not easily found.

The ETICS effort aims to support such research and development initiatives by integrating existing procedures, tools and resources in a coherent infrastructure, additionally providing an intuitive access point through a web portal and a professionally managed, multiplatform capability based on Grid technologies. The outcome of the project will be a facility operated by experts that will enabled distributed research projects to integrate their code, libraries and application, validate the code against standard guidelines, run extensive automated tests and benchmarks, produce reports and improve the overall quality and interoperability of the software.

ETICS objectives are not to develop new software but to adapt and integrate already existing capabilities, mainly open source, providing other research project with the possibility to focus their effort in their specific research field and to avoid wasting time and resources in such, required, but expensive, activity.

Throughout the duration of the project the ETICS partners will investigate the advantages of making use of the ETICS services, the technical challenges relates to running such a facility and its sustainability for the future.

The vision and mission of ETICS will be accomplished through the following objectives:

• Establish an international and well managed capability for software configuration, integration, testing and benchmarking for the scientific community.

Software development projects will use the capabilities provided by ETICS to build and integrate their software and perform complex distributed test and validation tasks
• Deploy and if necessary adapt best-of-breed software engineering tools and support infrastructures developed by other projects (EGEE, LCG, NMI) and other open-source or industrial entities and organize them in a coherent, easy-to-use set of on-line tools
• Create a repository of libraries that project can readily link against to validate their software in different configurations conditions
• Leverage a distributed infrastructure of compute and storage resource to support the software integration and testing activities of a broad range of software development efforts.
• Collect, organize and publish middleware and applications configuration information to facilitate interoperability analysis at the early stages of development and implementation
• Collect from the scientific community sets of test suites that users can apply to validate deployed middleware and applications and conversely software providers can use to validate their products for specific uses
• Raise awareness of the need for high-quality standards in the production of software and promote the identification of common quality guidelines and principles and their application to software production in open-source academic and research organization. Study the feasibility of a "Quality Certification"for software produced by research projects
• Promote the international collaboration between research projects and establish a virtual community in the field of software engineering contributing to the development of standards and advancement in the art

From the perspective of Grid application developers, the ETICS service should provide them with the means to automate their build and test procedures. In the longer term, via the ETICS service, users will be able to explore meaningful metrics pertaining to the quality of their software. Further, as Grid application level services (most concerned by providers of Grid turn key solutions), the ETICS service will also offer a repository or already built components, services and plug-ins, with a published quality level. Furthermore, the quality metrics provided by the ETICS services and available for each artifact in the repository will help guiding the user in selecting reliable software dependencies. Finally, the repository will also contain pre-build artifacts for specific hardware platforms and operating systems, which will help the developers to assess the platform independence of their entire service, including each and every dependency the service is relying on.

In conclusion, most Grid and distributed software project invest in a build and test system in order to automatically build and test their software and monitor key quality indicators. ETICS takes requirements from many Grid and distributed projects and with the help of Grid middleware, offers a generic yet powerful solution for building and testing software. Finally, building software via such a systematic can provide a rich pool of published quality components, services and plug-ins, on which the next generation of Grid and distributed applications could be based on and composed of.

1d: Computational Chemistry - Lattice QCD - Finance / 104

On the development of a grid enabled a priori molecular simulator

Authors: Antonio Lagana'[1]; Osvaldo Gervasi[2]

[1] *1Department of Chemistry, University of Perugia*

[2] 2Department of Mathematics and Computer Science, University of Perugia

We have implemented on the production grid of EGEE GEMS.0, a demo version of our Molecular processes simulator that deals with gas phase atom diatom bimolecular
reactions. GEMS.0 takes the parameters of the potential from a data bank and carries out the dynamical calculations by running quasiclassical trajectories [1].
A generalization of GEMS.0 to include the calculation of ab initio potentials and the use of quantum dynamics is under way with the collaboration of the members of COMPCHEM [2]. In this communication we report on the implementation of quantum dynamics procedures.
Quantum approaches require the integration of the Schroedinger equation to calculate the scattering matrix SJ (E). The integration of the Schroedinger equation can be carried out using either time dependent or time independent techniques.
The structure of the computer code performing the propagation in time of the wavepacket (TIDEP)[3] for the Ncond sets of initial conditions is sketched in Fig. 1.

```
Read input data: tfin, tstep, system data ...
     Do icond = 1,Ncond
        Read initial conditions: v, j, Etr, J ...
         Perform preliminary and first step calculations
         Do t = to, tfin, tstep
             Perform the time step propagation
             Perform the asymptotic analysis to update S
             Check for convergence of the results
         EndDo t
     EndDo icond
```

Fig. 1. Pseudocode of the TIDEP wavepacket program kernel.

The TIDEP kernel shows strict similarities with that of the trajectory one (ABCtraj)
already implemented in GEMS.0. In fact, for a given set of initial conditions, the inner loop of TIDEP propagates recursively over time the wavepacket. The most noticeable difference between this and the trajectory integration is the fact that at
each time step TIDEP performs a large number of matrix operations which increase memory and computing time requests of some orders of magnitude.
The structure of the time independent suite of codes [4] is, instead, articulated in a different way. It is in fact made of a first block (ABM) [4] that generates the local
basis set and builds the coupling matrix (the integration bed) using also the basis set of the previous sector. This calculation has been decoupled by repeating for each
sector the calculation of the basis set of the previous one (see Fig. 2). This allows
to distribute the calculations on the grid. The second block is concerned with the propagation of the solution R matrix from small to large values of the hyperradius performed by the program LOGDER [4]. For this block, again, the same scheme of ABCtraj can be adopted to distribute the propagation of the R matrix at given values of E and J as shown in Fig. 3.

```
Read input data: in, fin, step, J, Emax, ...
     Perform preliminary calculations
     Do  (rho) = (rho)in + (rho)step, (rho)fin, (rho)step
         Calculate eigenvalues and surface functions for present and previous
```

(rho)
Build intersector mapping and intrasector coupling matrices

```
EndDo (rho)
```

Fig. 2. Pseudocode of the ABM program kernel.

```
Read input data: in, fin, step, ...
Transfer the coupling matrices generated by ABM from disk
Do icond = 1,Ncond
   Read input data: J, E ...
   Perform preliminary calculations
      Do (rho) = (rho)in, (rho)fin, (rho)step
          Perform the single sector propagation of the R matrix
     EndDo (rho)
EndDo icond
```

Fig. 3. Pseudocode of the LOGDER program kernel.

References

1. Gervasi, O., Dittamo, C., Lagana', A.: Lecture Notes in Computer Science 3470, 16-22 (2005).
2. EGEE-COMPCHEM Memorandum of understanding, March 2005
3. Gregori, S., Tasso, S., Lagana', A: Lecture Notes in Computer Science 3044, 437- 444 (2004).
4. Bolloni, A., Crocchianti, S., Lagana', A.: Lecture Notes in Computer Science 1908, 338-345 (2000).

1d: Computational Chemistry - Lattice QCD - Finance / 105

The Molecular Science challenges in EGEE

Authors: Antonio Lagana'[1]; Cristian Dittamo[2]; Leonardo Arteconi[1]; Osvaldo Gervasi[2]

[1] *Department of Chemistry, University of Perugia*

[2] *Department of Mathematics and Computer Science, University of Perugia*

The understanding of the behavior of molecular systems is important for the progress of life sciences and industrial applications. In both cases is increasingly necessary to perform a study of the relevant molecular systems by using simulations and computational procedures which heavily demand computational resources. In some of these studies it is mandatory to put together the resource and complementary competencies of various laboratories. The Grid is indeed the infrastructure that allows such a cooperative modality of work. In particular for scientific purposes
the EGEE Grid is the proper environment. For this reason a Virtual Organization (VO) called CompChem has been created within EGEE. Its goal is to support the computational needs of the Chemistry and Molecular Science community and pivot the user access to the EGEE Grid facilities.
Using the simulator being implemented in CompChem the study of molecular systems is carried out by adopting various computational approaches bearing approximations of different levels.
These computational approaches can be grouped into three categories:
1. Classical and Quasiclassical: these are the less rigorous approaches.
They are, however, the most popular. The main characteristic of these computational procedures is that the related computer codes are naturally parallel. They consist in fact of a set of independent tasks, with few communications at the beginning and at the end of each task.

Related computational codes are suitable to exploit the power of the Grid
in terms of the high number of computing elements (CEs) available.
2. Semi-classical: these approaches introduce appropriate corrections the
deviations of quasiclassical estimates from quantum ones. The Grid
infrastructure is exploited for massive calculations by varying the initial
conditions of the simulation and performing the statistical analysis of the
results.
3. Quantum: this is the most accurate computational approach heavily demanding
in terms of computational and storage resources. Grid facilities and
services will be only seldomly able to support them in a proper way using
present
hardware and middleware utilities. Therefore they will represent a real
challenge for Grid service development.

The computational codes presently used are mainly produced by the laboratories member of the VO. However some popular commercial programs (DL POLY, Venus, MolPro, GAMESS, Columbus, etc) are also being implemented. These packages are at present executed only on the computing element (CE) owning the license. We are planning to implement in the Resource Broker (RB) the mapping of the licensed sites via the Job Description Language (JDL). In this way the RB will be able to schedule properly the jobs requiring licensed software. The VO is implementing[1] an algorithm to reward each participating laboratory for contributions given to the VO providing hardware resources, licensed software and specific competences.
One of the most advanced activities we are carrying out in EGEE is the simulation on the Grid of the ionic permeability of some cellular micropores. To this end we use molecular dynamics simulations to mimic the behavior of a solvated ion when driven by an electronic field through a simple model of the channel. As a model channel a carbon nanotube (CNT) was used as done in a recent molecular dynamics simulation of water filling and emptying of the interior of an open-end carbon nanotube[3-6]. In this way we have been able to calculate the ionic permeability
of several solvated ions (Na+, Mg++, K+, Ca++, Cs+) by counting the
ions forced to flow into the nanotube by the applied potential diffence along
z-axis.

References

1. Lagana', A., Riganelli, A., and Gervasi, O.: Towards Structuring Research Laboratories
   as Grid Services; submitted (2006).
2. Kalra, A., Garde, S., Hummer, G.: Osmotic water transport through carbon nanotube membranes. Proc Natl Acad Sci USA 100 (2003) 10175-10180.
3. Berezhkovskii, A., Hummer, G.: Single-file transport of water molecules through a carbon nanotube. Phys Rev Lett 89 (2002) 064503.
4. Mann, D.J., Halls, M.D.: Water alignment and proton conduction inside carbon nanotubes.
   Phys Rev Lett 90 (2003) 195503.
5. Zhu, F., Schulten, K.: Water and proton conduction through carbon nanotubes as a models for biological channels. Biophys J 85 (2003) 236-244.

Poster and Demo session + cocktail / 106

An efficient method for fine-grained access authorization in distributed (Grid) storage systems

Author: Andreas Peters[1]

[1] *CERN*

The ARDA group has developed an efficient method for fine-grained access authorization in distributed (Grid) storage systems. Client applications obtain "access tokens" from an organization's file catalogue upon execution of a file name resolution request. Whenever a client application tries to access the requested files, the token is transparently passed to the target storage system. Thus the storage service can decide on the authorization of a request without itself having to contact the authorization service.
The token is protected from access and modification by external parties using public key infrastructure. We use GSI authentication for identification to the catalogue service and to storage I/O daemons. The authorization system is as secure as GSI authentication and public key infrastructure can be. To improve the performance for the catalogue interaction, we use GSI authenticated sessions between client and server: after an initial full GSI authentication we encrypt every interaction between client and server with a dynamic symmetric key and achieve a 20 times faster performance.

The main information inside an authorization envelope are the TURL to be used by I/O daemons, the permissions on that TURL, which are 'read','write','write-once' and 'delete', the lifetime of that token, the certificate subject and the storage system name for which this token was issued. One token can contain the authorization for a group of files.

Traditional approaches use proxy->uid mapping services to apply local filesystem permissions. In a direct comparison an access token is equivalent to a VOMS proxy certificate who's proxy extensions authorize access to only one file or a group of files. However VOMS is not the appropriate system to perform authorization on file level since the issue time for such an envelope is very critical (in our implementation only few ms per access) and the VOMS integration, a VOMS server would need to be directly connected to the used file catalogues.

Our method is well applicable in situations, where every GRID user needs to have the possibility to declare a file as private to him.
The same would require in the traditional approach already one worldwide configured UID per VO member, which is very difficult to maintain if not impossible. In our implementation user roles and groups are completely virtualized through definitions in a file catalogue and do not need the one to one correspondence of roles and groups in storage systems.
In the future virtual machines might be the solution for a virtual user concept, but they are still far from deployment in the present Grid infrastructure.
Permissions in the catalogue must be attached to file GUIDs and the catalogue must make sure, that every GUID can be registered only once!

A well performing prototype using the AliEn Grid file catalogue and xrootd as a data server has been implemented. The integration of other catalogue or I/O daemons would be simple. The catalogue service itself can run different file catalogue plug-ins. The token is moved as part of a file URL, i.e. no I/O protocol changes are needed. I/O daemons need one modification in the 'open' command to decrypt the authorization envelope, reject access or replace the initial TURL passed to the open command with the TURL quoted in the envelope. This functionality is encapsulated in a C++ shared library, which allows to define additional authorization rules for certain VOs, certificates or TURL paths.

2c: Special type of jobs (MPI, SDJ, interactive jobs, …) - Information systems / 107

Grid-Enabled Remote Instrumentation with Distributed Control and Computation

Authors: Constantinos Kotsokalis[1]; Luke Dickens[2]; Tiziana Ferrari[3]

[1] *GRNET*

[2] *Imperial College*

[3] *INFN*

1 GRIDCC Applications and Requirements

The GRIDCC project [1], sponsored by the European Union under contract number 511381, and launched in September 2004, endeavors to integrate scientific and general-purpose instruments within the Grid. The motivation is to exploit the Grid opportunities for secure, collaborative work of distributed teams and to utilize the Grid's massive memory and computing resources for the storage and processing of data generated by scientific equipment. The GRIDCC project focuses its attention on eight applications, four of which will be fully integrated, tested and deployed on the Grid.
The PowerGrid will support the remote monitoring and control of thousands of small power generators; while the Control and Monitoring of HEP experimentsaims to enable remote control and monitoring of the CMS detector at CERN. The (Far) Remote Operation of Accelerator Facility is an application for the full operation of a remote accelerator in Trieste, Italy; and the Grid-based Intrusion Detection System aims to provide detection and trace-back of flow-based DoS attacks using aggregated data collected from multiple routers. The other set of relevant applications includes: meteorology, neurophysiology, handling of device farms for measurements in telecommunications laboratories, and geophysiology [2][5].
The project, by nature, requires the availability of software components that allow for time-bounded and secure interactions, while operating instrumentation in a collaborative environment. In addition to the classical request/response Grid service interaction model, a considerable amount of information needs to be streamed from the instrument back to the user. The time-bounded interactions, dictated either by the instrument sensitivity and the accompanying requirement for careful handling and fast response to extreme conditions, or by the applications themselves, lead to the need for the establishment of SLAs for QoS or other guarantees, with support for compensation and rollback. The idea of collaboration and resource sharing, inherent in the Grid, is also extended and adapted to allow the share of unique instruments among users who are geographically dispersed, and who normally would not have access to such –usually rare and/or expensive – equipment.

2 GRIDCC and gLite

To cater for the diversity of instruments and the critical nature of the equipment being handled, the GRIDCC middleware platform relies on Web Service (WS) technologies, and sustains a Service Level Agreement (SLA) infrastructure, alongside enforcement of Quality of Service (QoS) guarantees. The GRIDCC middleware architecture is fully described in [2].
A number of gLite software components are extremely relevant to the GRIDCC middleware architecture, which is designed to comprise various novel middleware components to complement them. Firstly, we plan to perform job scheduling and bookkeping via the WMS and specifically the WMProxy, and the LBProxy [2]. We also plan to rely on the Agreement Service for SLA signalling and for triggering resource-level reservations [2]–this is essential to enforce SLA guarantees. In addition, we plan to test and possibly extend CREAM, as explained in the following Section.
The WSDL interface, exposed by the gLite WMS, streamlines job submission in a number of different scenarios: direct invocation by the Virtual Control Room (VCR) - the GRIDCC portal; direct submission onto preselected CEs via the GRIDCC Workflow Management System (WfMS); and indirectly, utilising the WMS's builtin scheduling capabilities, either as a single submission or part of a workflow [2]. The WfMS and VCR are described in more detail in Section 3.
Data gathered from IEs need to be stored, in MSS sevices. Consequently, data

storage will be delegated to gLite SEs exposing SRM-compliant interfaces.
VOMS and proxy-renewal services will be used. For authentication and authorization, it is foreseen to support both X.509 certificates and the Kerberos framework. The latter will be used when low response times are required.
Finally, for QoS performance monitoring, as it is experienced by GRIDCC users and services, we require the integration of service monitoring tools and services providing information about network performance, such as the gLite Network performance Monitoring framework.

3 GRIDCC Middleware

The gap between GRIDCC's requirements and gLite's existing service support, will be filled by a number of GRIDCC solutions, which leverage the existing gLite functionality.
The need for instrument support, necessitated the development of a new grid component, the Instrument Element (IE). The IE's naming and design reflect its similarity to gLite's SE and CE. The IE provides a Grid interface to a physical instrument or set of instruments, and should allow the user to control and access instrument data [2]. To cater for the varied needs of instrumentation, the IE also has local automated management and storage capacity [2].
The desire for QoS and SLA support is provided for by the following Execution Service components. The gLite AS will be extended to establish SLAs with the IE, and the IE will need to enforce such SLAs. To achieve this, the IE conceptual model and schema need to be defined in order to publish information about the instrument-specific properties.
The GRIDCC Workflow Management System (WfMS) provides an interface for users to submit workflows, which can orchestrate WS calls to underlying services [3]. The WfMS may also need to choreograph further steps into workflows, such as the SLA negotiation and logging steps, to facilitate the satisfaction of, possibly complex, QoS demands from the user [3]. It is also responsible for monitoring running workflows and responding to workflow events - such as contacting a user if QoS demands can no longer be satisfied [2].
The Virtual Control Room (VCR), supports a user Grid portal for the underlying services, in particular to: request SLAs from the AS; steer and monitor an IE; and submit workflows to the WfMS [2][3][4]. Additionally, the VCR provides a multi-user collaborative online environment, wherein remote users and support staff, share control of and troubleshoot IEs [2][4].

4 Extending gLite

To fulfill the GRIDCC application requirements, a number of gLite functionality extensions would be useful for successful middleware integration. Firstly, information about IEs needs to be made available by the information services. Secondly, in order to enforce upper-bounded execution times, the reservation of CEs and IEs needs to be supported. To this end, we will extend the AS, by adding CE and IE-specific SLA templates. Reservation needs to be triggered and enforced by elements at the fabric-layer. For this reason, we envisage the addition of a new operation to the WSDL interface exposed by CREAM, allowing the invocation of reservation operations. As mentioned above, GRIDCC, needs for QoS to be enforced at both the single-task and workflow level. The WMS already supports some workflow functionality; however, the WMS can only process workflows involving job execution tasks. We foresee the need to merge the functionality of the GRIDCC WfMS with the gLite WMS, to benefit from the existing WMS capabilities and avoid duplication of work.

References
[1] The GRIDCC Project home page: http://www.gridcc.org.
[2] The GRIDCC Architecture –Architecture of Services for a Grid Enabled Remote Instrument Infrastructure (http://www.gridcc.org/getfile.php?id=1382).
[3] D4.1 Basic Release R1, GRIDCC Project Deliverable GRIDCC-D4.1, May 2005 (https://ulisse.elettra.trieste.it/tutos_gridcc/php/file/file_show.php?id=1418)
[4] Multipurpose Collaborative Environment, GRIDCC Project Deliverable GRIDCC-D5_2, Sept 2005

(https://ulisse.elettra.trieste.it/tutos_gridcc/php/file/file_show.php?id=1408)
[5] SPECIFIC TARGETED RESEARCH OR INNOVATION PROJECT –Annex I - "Description of Work", May 2004 (http://www.gridcc.org)
[6] EGEE Middleware Architecture and planning, EGEE Project, Deliverable EGEE-DJRA1.1-594698-v1.0, Jul 2005 (https://edms.cern.ch/document/594698/).

2b: Data access on the grid / 108

Discussion

Discussion on metadata catalogues

2b: Data access on the grid / 109

Discussion

Discussion on grid data management

2b: Data access on the grid / 110

Discussion

Discussion on application data management

111

Experience Supporting the Integration of LHC Experiments Software Framework with the LCG Middleware

2a: Workload management and Workflows / 112

Title: "IBM strategic directions in workload virtualization"

Author: Jean-Pierre Prost[1]

[1] *IBM Montpellier*

"Workload virtualization is made of several disciplines: job/workflow scheduling, workload management, and provisioning. Much work has been spent so far on these various components in isolation. A better synergistic integration of these components allowing their interoperability towards an optimized resource allocation in order to satisfy user specified service level objectives is necessary. Other challenges in the grid space deal with being able to allow meta-scheduling and

adaptive/dynamic workflow scheduling. In this talk, we present IBM strategic directions in the workload virtualization area. We also
briefly introduce our current product portfolio in that space and describe how it may evolve over time, based on customer requirements and additional business value their satisfaction could provide them."

EGAAP open session / 115

Introduction

Author: Vincent Jacques Breton[1]

[1] *Laboratoire de Physique Corpusculaire (LPC)*

EGAAP open session / 116

Fusion Status Report

Author: Francisco Casatejón[1]

[1] *CIEMAT*

EGAAP open session / 117

ARCHEOGRID Status Report

Author: P. G. Pelfer[1]

[1] *Univ. Florence and INFN*

EGAAP open session / 118

EUMEDGrid Status Report

Author: Roberto Barbera[1]

[1] *University of Catania and INFN*

EGAAP open session / 119

EELA Status Report

EGAAP open session / 120

EUchinagrid

Author: Giuseppe Andronico[1]

[1] *INFN SEZIONE DI CATANIA*

EGAAP open session / 121

Bioinfogrid

Author: Giorgio Maggi[1]

[1] *Univ. + INFN*

EGAAP open session / 122

Discussion on EGAAP future in EGEE-II

123

Discussion of the Status Reports and Proposals

Closed Session

124

Preparation of the Final EGAAP Report

Closed Session

125

EGAAP and EGEE-II

Closed Session

126

AOB

127

Coffee break

128

Preparation of the Final EGAAP Report

129

EGAAP and EGEE-II

130

AOB

131

Oracle

2b: Data access on the grid / 132

Use of Oracle software in the CERN Grid

Author: Bjorn Engsig[1]

[1] *ORACLE*

Oracle is known as a database vendor, but has much more to offer than data storage solutions.
Some key Oracle products that are in use or are being currently full-scale tested at CERN will be discussed in this talk.
It will primarily be an open discussion and interactive feedback from the audience is more than welcome

The following topics will be discussed:

Oracle Client Software distribution

How can a large to huge number of systems be given easy possibility to connect to Oracle database servers; what are the distribution rights and how is it actually distributed and configured.

Oracle Support for Linux

Oracle officially supports those Linux distributions that are in widespread use and strongly recommends that servers are being run on supported distributions. This does however not imply, that other Linux distributions cannot at all be used. This talk will elaborate on this.

Oracle Streams Replication

The various possibilities for using Oracle Streams to replication large amounts of data will be discussed.

2d: VO tools - Portals / 133

Discussion