

GRID SYSTEMS AND THEIR APPLICATIONS TO BIOMEDICAL SCIENCE

SYSTEMY GRIDOWE I ICH ZASTOSOWANIA W NAUKACH BIOMEDYCZNYCH

MACIEJ MALAWSKI*, TOMASZ SZEPIENIEC**, IRENA ROTERMAN-KONIECZNA***

* *Institute of Computer Science AGH, Mickiewicza 30, 30-059 Kraków*

** *ACK Cyfronet AGH, Nawojki 11, 30-950 Kraków*

*** *Department of Bioinformatics and Telemedicine, Jagiellonian university Medical College,
Faculty of Medicine, św. Łazarza Str. 16, 31-530 Kraków, www.bit.cm-uj.krakow.pl*

Abstract: In this paper we briefly introduced the current developments in e-science and the potential the Grid technology can bring to the scientific community in terms of sharing computing power, access to scientific data and supporting collaboration. After high energy physics, which was the pioneering science exploiting the benefits of grid systems, now also the biomedical sciences are beginning to join the Grid community. The important issue is that the adaptation of applications to the grid requires some effort; therefore a close collaboration between computer scientists (grid experts) and domain scientists becomes a crucial factor to success. What makes the collaboration between biomedical and grid computing community especially promising and challenging is the fact that the complexity of the biological systems is so high, that even the largest computing infrastructure such as EGEE project is not powerful enough to solve all the problems of biomedical science. This still requires a lot of interesting scientific work in the development of new bio-algorithms and grid technologies to support them.

Keywords: grid systems, distributed computing, large-scale applications, in-silico experiments

Streszczenie: Celem artykułu jest przedstawienie podstawowych zagadnień dotyczących systemów gridowych i ich zastosowań w naukach biologicznych i medycznych. Ideą systemów gridowych jest współdzielenie zasobów pomiędzy rozproszonymi ośrodkami komputerowymi na świecie, w celu efektywniejszego wykorzystania ich mocy obliczeniowej i umożliwienia realizacji zadań wymagających współpracy pomiędzy wieloma grupami naukowców. Aplikacje mogące wykorzystać potencjalne możliwości gridu to takie obliczenia, które dają się wykonać w sposób równoległy, czyli dają się podzielić na większą ilość zadań do wykonania. Systemy gridowe umożliwiają automatyczne przydzielanie zadań do dostępnych zasobów obliczeniowych oraz przesyłanie danych w obrębie gridu. Obecnie największa na świecie infrastruktura gridowa projektu EGEE dysponuje 30000 procesorów i 14 PB danych w 180 ośrodkach w 42 krajach świata. Przykładami biomedycznych aplikacji gridowych są symulacje struktury białek, testowanie leków przeciwko wirusom ptasiej grypy, symulacje przepływu w układzie krwionośnym pacjenta a także analizy statystyczne i epidemiologiczne na dużych klinicznych zbiorach danych.

Słowa kluczowe: systemy gridowe, obliczenia rozproszone, aplikacje wielkiej skali, eksperymenty in-silico

E-science and Grid technology

Scientific computing becomes an increasingly important methodology of conducting modern research, complimenting traditional theoretical and experimental activities. Computers can be used to perform large-scale simulations and data analysis, while the Internet allows access to huge amount of scientific data and facilitates collaboration among scientists. Such a trend of increasing utilization of information technology for scientific applications is often described under the term e-science. E-science poses several requirements on computer infrastructure, and we will enumerate the most important ones:

- A huge amount of computer power is required to conduct simulations of natural phenomena, e. g. in climate modeling, geophysics, biology or pharmacy
- Current experiments produce a large amount of data, which need to be stored, made accessible to large collaborations, and processed during the analysis stage; prime examples are experiments in high energy physics producing data in the order of Peta Bytes per year.
- Infrastructure belonging to and administered by single institutions should be shared among others, allowing more optimal resource utilization and collaborative access to shared instruments, e. g. telescopes, satellites or spectrometers.

These requirements lead to the development of Grid technology [Grid], which is regarded as perhaps the most suitable solution for providing the information infrastructure for e-science. Since there is no simple or widely agreed definition of a Grid, we will try to briefly introduce the main ideas associated with this term.

Generally, the most typical Grid system consists of resources, which may include computing processors (CPUs), mass storage media (disk systems, tape libraries), and instruments. These resources are in general heterogeneous and geographically distributed, and are owned and maintained by participating institutions, often large computing centers. This means that they belong to separate administrative domains, with various access mechanisms and policies, including security considerations. It is now becoming clear, that sharing such resources between institutions in a collaborative way is not a trivial task. Grid technology aims at facilitating such collaborative resource sharing, by virtualizing the resources in order to make them appear in a uniform way to the users. Many institutions may form a so-called virtual organization and offer some of their resources for collaborative usage, benefiting from better balanced utilization of CPU power or access to shared data, without compromising local security policies.

There are many additional topics closely related to grid systems, which are however sometimes excluded from the definition. One is the utilization of idle CPU power, e. g. from workstations in student labs, unused during nightly hours, or even when a person is not interacting with a keyboard. Such idle CPU cycles can be used for performing some useful computations, and theoretical computing power of such systems can be very large. Similarly, computations may be performed using peer-to-peer networks, which offer even larger potential computing power, however with

smaller reliability and limited accessibility. Also, it is possible to build Grid-like infrastructures hidden inside a company or institution intranet.

Grid environments implementation

Having defined a grid system, now we would like to describe an example of an implementation of this idea. Probably the most popular and the most useful from the perspective of users is grid maintained by the EGEE project [EGEE]. The initial motivation of building this grid was to build infrastructure for processing and storing data produced in high energy physics experiments using Large Hadron Collider at CERN, but later the infrastructure was opened to other scientific disciplines and evolved into the main scientific computational platform in Europe. Currently the EGEE grid integrates the power of more than 30 000 CPUs and enables a possibility to store about 14 PB (1PB = 1000 000 GB) of data. The infrastructure is supported by about 180 installations distributed in 42 countries worldwide.

The EGEE grid infrastructure is based on a set of software packages that enables all necessary grid services. This software is usually named as *Grid middleware*, because this is software layered between *fabric*, which is hardware and the operating system, and the user's applications. These packages, integrated in the frame of the EGEE Project, are named gLite. The services of this middleware, could be put in two groups (see Fig. 1) *basic services* ensuring all functionality on resources connected to the Grid system like security, user access to the resource, ensuring data transfers and providing information about the resource, and *collective services* that manage work flow and data flow through many resources.

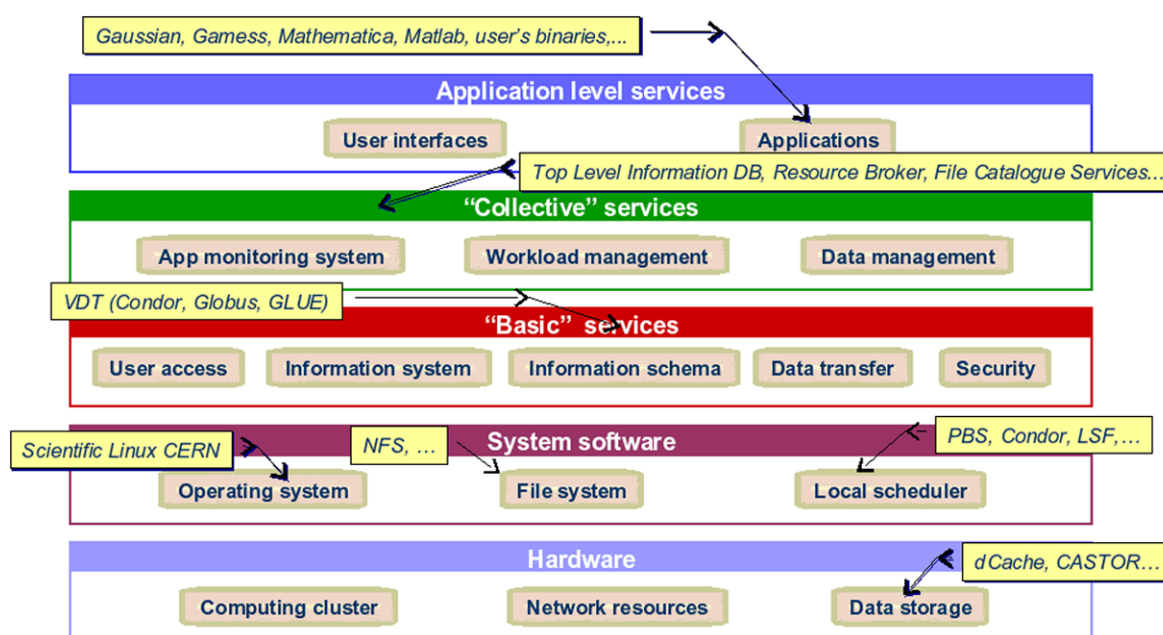


Fig. 1. gLite software stack (source: EGEE Project materials)

Users in Grid

The Grid infrastructure can be treated as a platform for sharing resources, but, of course, rights of using resources are not equal. While the Grid is subject to non-centralized administration, each resource provider could maintain his own policy for using the resources he manages. On the other hand, there are thousands of users that want to ensure resources for their needs. Providing negotiation between single resources providers and every single user would be unfeasible, so the idea of groups of users, named *Virtual Organizations* (VO) was introduced. VO members join the VO to share resources and data for a common goal. VO management, on behalf of all members, negotiates conditions of using the resources with the providers.

In EGEE infrastructure there are several VOs officially supported by the project and much more regional and smaller VOs maintained outside the project, but using the infrastructure. They may support a single experiment (like ATLAS VO – high energy physics project), gather scientists using the same computational package (for example GAMESS VO) or even working in the same scientific domain (like biomed VO or CompChem VO). There are also some initiatives to enable the possibility of grid newcomers to join and test their applications in a Grid environment. Example of such a VO would be VOCE – Virtual Organization for Central Europe.

Grid applications

When discussing Grid systems, we have to answer the question, which kinds of applications are best suitable for running them on the Grid infrastructure and most easily adapted to such an environment. After describing some general remarks we will give examples of applications that are successfully run on current Grids.

General hints for application designers and developers come from the nature of a Grid system, which is a large-scale distributed system, consisting of heterogeneous resources, without a central control and with diversified network performance. Most grid systems offer a batch processing capability, where a single unit of application is a job processing some input files and writing the output to another set of files. A job may consist of multiple processes, possibly distributed between many Grid sites, and communicating using any supported mechanism. Most grid systems do not guarantee the successful execution of a batch job, which may fail unexpectedly. Therefore there is a need to provide some fault tolerance mechanisms on top of existing infrastructures. Another popular approach is based on the service oriented architecture [Foster05], where a grid application may consist of invocations of services offered by others. Such services may provide various computational or data access functionalities, e. g. offering remote access to scientific computing software libraries or databases.

The first class of applications most naturally fitting the Grid environment are the so called embarrassingly parallel problems, which can be solved by running a set of independent copies of an identical program, differing one from

another by some parameter or input data. The processes do not communicate between them at runtime and may be run on many distributed resources in parallel without the need for high-speed network. Such applications may be sometimes naturally fault-tolerant, which means that a failure of some percentage of jobs does not influence the final result. To this class of applications belong e. g. Monte Carlo simulations, where the loss of some jobs has no statistical significance.

Parallel-distributed applications form another class. They consist of multiple processes, which communicate with each other at runtime for synchronization and data exchange purposes. It is well known fact, that the scalability of such applications, which may be very good on a massively parallel machine or a single computing cluster, may decrease substantially when executing on distributed grid resources, which are more loosely coupled. However, such applications may be also ported to a grid, but only when the communication volume between processes is reduced to a minimum.

The third interesting class of applications can be described in terms of a scientific workflow. A workflow is a set of activities (may be jobs, services) executed in a specified order, when subsequent steps depend on the results of previous ones. Such a process may be represented as a graph defining activities and their dependencies. Workflows can be very useful for representing tasks performed repeatedly and may consist of such steps as data access, preprocessing, many simulation runs, and data analysis.

Adaptation of applications to the Grid

There are two main ways of adapting applications to the Grid environment, depending on the selection of batch- or service- oriented architecture.

In the case of batch processing systems [Nabr03], there are a few steps required to transform a standalone program into a job which can be submitted on the Grid. First, the program should be transformed into a single executable or a script, which reads some input files and writes its output also to disk files. The Grid middleware is able to transfer the executable and all input files (often called an “input sandbox”) to the machine on the Grid and execute it. The user can check the status of the job and retrieve output when ready. It is important to consider all dependencies such as libraries or additional data, which need to be either pre-installed on the Grid machines or transferred together with the sandbox. It is also possible to use the files which are stored and registered in a Grid-enabled storage system, so they can be accessed by the computing job. The process of submitting the jobs may be controlled manually by a user by command line or portal interface, or may be automated by tools for managing bulk submission of jobs.

When the Grid is based on service oriented architecture [Foster05], the application should be decomposed into services which can be deployed on the Grid. A service can be defined as software offering some well defined functionality (operations) which are accessible for invocation over the network using a published interface and protocol. Per-

haps most often the Web services standards such as SOAP [SOAP] and WSDL are used to interface the services in Grid systems. These standards allow the implementation of a service in many programming languages, so an existing program or library can be quite easily wrapped into a service. When there are many services installed on Grid sites and available to users, it is possible to orchestrate their operations to form a parallel processing application or a workflow where outputs of previous steps are used as inputs to subsequent services.

Biomedical Application Examples

Starting from parameter study application, we could give the example of drug discovery application. Before the new drug can be applied to standard medical procedures, the complicated experiments must be performed ensuring positive reaction upon its presence in the organism. The large part of these analyzes is performed *in silico*. Simulations of different systems of pharmacological character are able to estimate the positive and/or negative consequences of the presence of potential drug in the human organism. This is why the preliminary analysis *in silico* is important for cost lowering as well as for shortening the time of experimental part of research. Particularly, protein molecules are in focus of the attention of pharmacologists because of their role in functioning of our body. The grid system, assumed to be the result of the project accomplishment, would be used for large scale computing on protein structure in the context of drug design. This kind of research is done in the frame of EUChinaGRID Project [EUChG] by our team.

Another exercise worth mentioning was done using Wisdom Initiative [Wisdom] application adapted to the grid – in April 2006, 300,000 possible drug components was tested against the avian flu virus H5N1 using the EGEE Grid infrastructure. The goal was to find potential compounds that can inhibit the activities of an enzyme on the surface of the influenza virus, the so-called neuraminidase, subtype N1. Using grid in this case helped to mobilize resources that was able to complete in 25 days computations that in case of using a single processor would take almost 19 years.

As an example of a medical application which includes parallel programs running on the Grid, may serve the environment for the simulation and visualization of a cardiovascular system [Tirado04]. The system is designed to support a medical doctor in surgery planning. The application con-

sists of a Lattice-Boltzmann blood flow simulation which is a parallelized code running on a high performance cluster on the Grid. It is connected to the visualization system, which allows a user to dynamically interact with the simulation running on remote resources. A doctor can see simulated results of e.g. adding a bypass before the surgery begins. Additional steps like data acquisition from a CT scan, pre-processing and segmentation are also supported.

Many examples of biomedical applications which can be run on a Grid using the workflow approach are developed in the course of myGrid Project [myGrid, Goble03]. The project created a general purpose workflow engine which can orchestrate the services responsible for data access and computation steps of the application on the Grid. Example usages include gene identification and sequence analysis in drug discovery, analysis of clinical records, discovery and analysis in medical imaging systems, and text mining in scientific literature.

References

- [EGEE] EGEE Project: <http://www.eu-egee.org/>
- [Grid] I. Foster, C. Kesselman, S. Tuecke: The Anatomy of the Grid: Enabling Scalable Virtual Organizations. International J. Supercomputer Applications, 15(3), 2001.
- [EUChG] EUChina Project: <http://www.euchinagrid.org/>
- [Tirado04] A. Tirado-Ramos; P.M.A. Sloot; A.G. Hoekstra and M. Bubak: An Integrative Approach to High-Performance Biomedical Problem Solving Environments on the Grid, Parallel Computing, (special issue on High-Performance Parallel Bio-computing) vol. 30, nr 9-10 pp. 1037-1055, 2004.
- [mygrid] myGrid Project: <http://www.mygrid.org.uk>
- [gLite] gLite middleware: <http://www.glite.org/>
- [Goble03] C.A. Goble, S. Pettifer, R. Stevens and C. Greenhalgh: Knowledge Integration: In silico Experiments in Bioinformatics in The Grid: Blueprint for a New Computing Infrastructure Second Edition eds. Ian Foster and Carl Kesselman, 2003, Morgan Kaufman, November 2003.
- [Foster05] I. Foster. Service-Oriented Science. Science, vol. 308, May 6, 2005..
- [SOAP] SOAP Version 1.2, W3C Recommendation 24 June 2003, <http://www.w3.org/TR/soap>
- [Nabr03] J. Nabrzyski, J.M. Schopf, J. Weglarz (Eds). Grid Resource Management, Kluwer Publishing, Fall 2003.
- [Wisdom] WISDOM Initiative homepage: <http://wisdom.eu-egee.fr/avianflu/>

