

Architecture of the Grid for Interactive Applications^{*}

Marian Bubak^{1,2}, Maciej Malawski¹, and Katarzyna Zajac¹

¹ Institute of Computer Science, AGH, al. Mickiewicza 30, 30-059 Kraków, Poland

² Academic Computer Centre – CYFRONET, Nawojki 11, 30-950 Kraków, Poland
{bubak,malawski,kzajac}@uci.agh.edu.pl
phone: (+48 12) 617 39 64, fax: (+48 12) 633 80 54

Abstract. In this paper we present the current status of the CrossGrid architecture. The architecture definition follows from the specification of requirements and design documents. It consists of descriptions of functionality of new tools and Grid services and indicates where interfaces should be defined. The components of the CrossGrid architecture are modular and they are organized in the following layers: applications, supporting tools, application development support, application-specific Grid services, generic Grid services, and fabric. We also present an analysis of the possible evolution of the CrossGrid services towards the OGSA service model.

Keywords: Grid, architecture, services, interactive applications, OGSA

1 Introduction

The CrossGrid Project [6,5] is oriented towards applications which are characterized by interaction with a person in a processing loop. Each application requires a response from the Grid to an action by a human agent in different time scales: from real time through intermediate delays to long waiting periods. The applications are simultaneously compute- and data-intensive. The following applications are currently being developed: simulation and visualization for surgical procedures, flooding crisis team decision support system, distributed data analysis in high-energy physics, air pollution combined with weather forecasting.

The medical application is a distributed near-real-time simulation with a user interacting in virtual reality and other interactive display environments. A 3-D model of arteries is the input to a blood flow simulation and the results are presented in a VR environment. The user alters the layout of the arteries and the effects are analysed in near-real-time through VR.

Flood forecasting requires meteorological simulations of different resolutions, from mesoscale to storm-scale. Next, selected hydrological models are used to determine water discharges from the affected area, and with this information

^{*} This work was partly funded by the European Commission, project IST-2001-32243, CrossGrid [6]

hydraulic models simulate flow through various river structures to predict the impact of the flood. The results of simulations can be interactively analysed, some of them accepted and forwarded to the next step, some rejected or re-run with modified inputs.

Distributed data analysis in high-energy physics addresses access to large distributed databases in the Grid environment and development of distributed data mining techniques suited to the high-energy physics field. Data mining services based on supervised and unsupervised learning with processing on worker nodes and on the database side will be elaborated.

2 The CrossGrid Approach to Interactive Applications

The medical application requires a distributed environment consisting of simulation, interaction and visualization components which will allow the user to change simulation parameters in near-real time. For that purpose, we propose an approach to problem solving environments consisting of a set of Grid services which allow the setup and interactive steering of complex Grid applications, with separate modules for simulation and visualization. This environment is a consistent, non-complex framework in which complex systems can be composed from reliable sub-units. The core of the proposed approach is an event system, the functionality of which will be based on existing standard infrastructures for distributed interactive simulations, called the High Level Architecture (HLA).

A second type of interactive applications is required for flood prediction and protection system. An interactive Grid system fulfilling the needs of this application should allow experts to prepare cascades of meteorological, hydrological and hydraulic simulations basing on the assumption that each preceding step of the cascade produces input for the next simulation. After each of the steps is completed, the expert should be allowed to decide whether there is a need for the next simulation step in the cascade to be performed.

High Energy Physics and air pollution modeling applications require support from a Grid interactive system that allows for on-line progress monitoring of their results in order to help operators decide about further job execution (i.e. interrupting the execution or letting it finish). To track the progress of a distributed job execution on-line, the Interactive Session Manager (ISM) is proposed which communicates with the user interface using XML, sets up distributed jobs consisting of Interactive Session Workers, and collects information from all nodes to build a global result for the user.

3 Layers of the CrossGrid Software

In the first approximation the architecture of the software that constitutes CrossGrid is shown in Fig. 1.

The top layer comprises applications that are developed within the project.

The second layer represents tools that facilitate development of applications. There are MPI verification tools, performance analysis and prediction tools, as

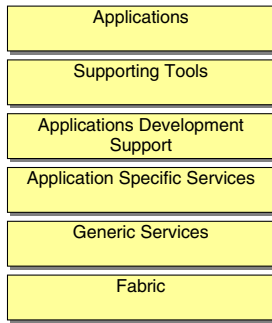


Fig. 1. Layers of the architecture.

well as Grid benchmarks. The tools are adapted to the Grid environment where the developed applications are running.

The third layer is built of such supporting software as communication libraries (MPICH-G), HLA and GVK. They are used at runtime.

The two lower layers are for Grid services: generic and application-specific. They form the middleware of the Grid and to some extent they exploit existing software developed by such projects as EU DataGrid and Globus Toolkit. There are also new Grid services developed by CrossGrid: Scheduling Agents, Data Access, Grid Monitoring, Roaming Access and Interaction services.

The bottom layer is the fabric, which consists of operating systems, queuing systems cluster management systems and others. The Fabric layer is encapsulated by the software developed within Globus and DataGrid.

4 Architecture Definition

The Fig. 2 shows the second version [1] of the architecture of the CrossGrid Project, i.e. the general overview of components and their relationships. The dependencies on external components, such as DataGrid and Globus software are presented as well. The components are placed in the appropriate layers defined in Sec. 3

Components directly dependent on the DataGrid project [7] are Scheduling Agents and the Data Access package. These services extend the basic functionalities of the DataGrid software. The former will provide addressing for parallel and interactive jobs and the latter will extend the existing systems by optimization of access to tape- resident data.

The point that will enable access to the Grid from portals is the Roaming Access Server. It will also offer users the possibility to access the same working environment from any workstation by means of a Migrating Desktop. It will provide a set of components that will interface with the underlying Grid services. The application developers' task is to write application- specific plugins that can be placed in a portal (see Fig. 3). Work on specification of plugin capabilities and APIs is now in progress.

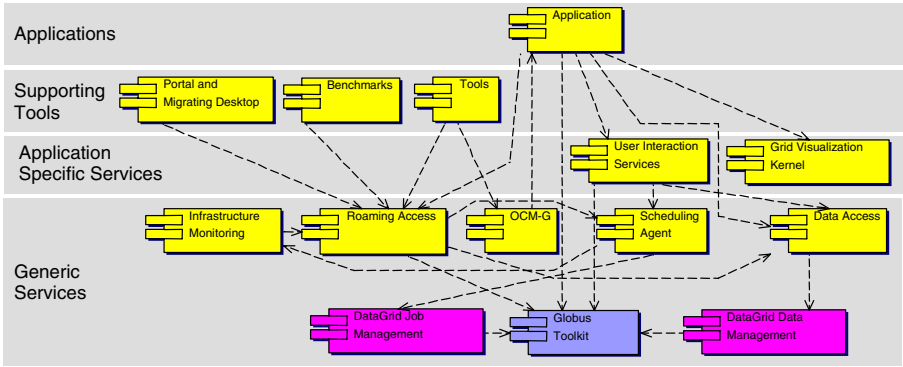


Fig. 2. Architecture overview.

Under a common name of monitoring, there exists a set of services used for different kinds of information gathering and processing. We have on-line monitoring of running applications [3] that provides data for the performance analysis tool [4]. The Jiro-based monitoring system gathers data about infrastructure and there is also a system for monitoring network traffic through the use of specialized hardware. Data from these services will be used by schedulers to make decisions about finding resources where jobs are run.

There is also a dependency between applications and tools. Current application kernels are used for development and testing of tools, and, finally, tools will be used to facilitate the application development process.

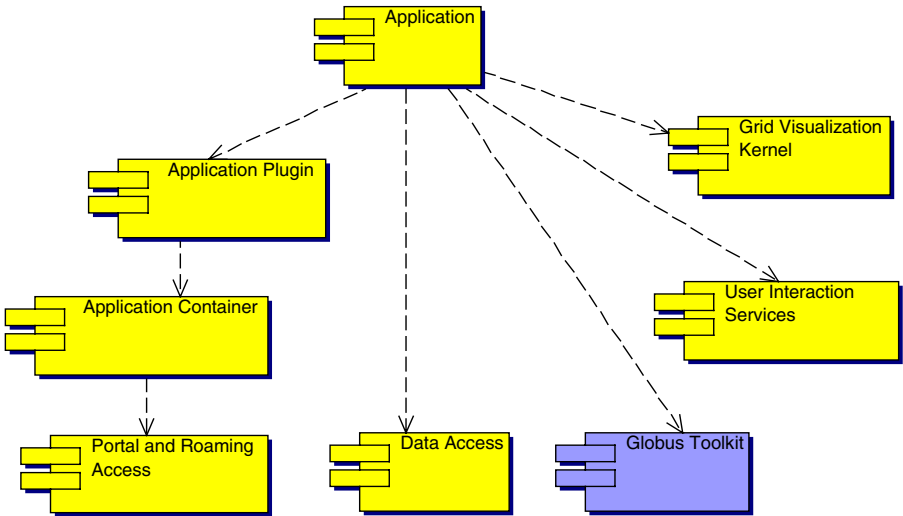


Fig. 3. Application-centric view of the architecture.

5 Evolution towards OGSA

The evolution of the CrossGrid architecture is inspired and based on the schedule of the development of Globus 3.0 (GT3) and OGSA technology. The OGSA [8] specification is scheduled to be defined in 2003 and GT3 is going to be released later this year. We believe OGSA will become a standard technology for building Grids and in our opinion, the Grid services that are being developed in CrossGrid may also be designed and implemented as OGSA-compliant, giving significant contribution to the Grid community.

OGSA which was recently proposed and is actively developed by the Globus team, is intended to be a basic technology for building Grid systems. Through Web services, actual interoperability and implementation independence may be achieved. The basic service semantics included in the Open Grid Services Infrastructure (OGSI) are used as building blocks in Globus Toolkit 3.0 (GT3). The Global Grid Forum OGSA Working Group is discussing a higher-level set of services that are going to be used in Grids and that can be integrated into the OGSA framework [9].

The first step toward OGSA, which is independent of the changes in evolving Grid Service framework and may be considered even for the first system prototypes is the usage of the Web services technology to expose external interfaces of the services. This may be applied to Roaming Access, Scheduling and Data Access services in CrossGrid.

The second step may consist of using specific extensions of Web services that are present in the OGSA model. These are mechanisms for dynamic service creation, lifetime management, introspection and information (service data) management based on XML. Dynamic service creation and lifetime management mechanisms can be used when it is necessary to control the state of some processes, e.g. user sessions in a portal, data transfers or a running simulation. The service data model can be applied to monitoring systems that can be used as information providers to other services. We plan to discuss such possibilities during the second year of the project.

6 Status of the First Prototype

During the first year of the project the work concentrated on definition of software requirements, design and implementation of the first prototypes of software. According to the project plan, the software components are not yet fully integrated, but they are tested separately on local cluster infrastructures.

The prototypes of portal and migrating desktop show the basic functionality of job submission and output retrieval. Scheduling agents extend the functionality of DataGrid resource broker by possibility of selecting multiple resources needed by parallel applications. Infrastructure monitoring services enable tracing of basic parameters of computing nodes and switches as well as network traffic in a cluster system. Additionally, the application monitoring system can gather and propagate the information of events inside the running program, such as MPI

calls. Data access package introduces the system of components for handling different types of tertiary storage.

The prototypes of development tools include a library for MPI code verification, that produces a trace of a program and attempts to resolve some of the possible problems, like deadlock. The prototype benchmark from the grid benchmark suite is based on High Performance Linpack and is running on the cluster platform. The performance analysis tool makes usage of the application monitoring system and can show such basic metrics of the application running on cluster as the communication volume between processes and CPU time consumed.

The MPI based HEP application computing training of neural networks was used to successfully demonstrate the usage of the portals, monitoring systems and development tools. The remaining applications are to be integrated with the CrossGrid tools during the second year of the project. During this phase the migration from the clusters to the Grid environment is scheduled.

7 Concluding Remarks

This paper presents a view of the CrossGrid architecture, its components, their functionality and relations between them as well as with components from Globus and DataGrid projects. The components of the CrossGrid software will be developed according to the iterative improvement approach consisting of fast prototyping, design, implementation, testing on testbeds, evaluation and further improvement. All software is produced according to the evolutionary life-cycle model with well-established phases of specification, development and validation. The first prototype of the CrossGrid middleware, tools and applications is available to the scientific community from the CrossGrid software repository [10].

Acknowledgements. We wish to thank M. Turała, M. Garbacz, P.M.A. Sloot, D. van Albada, L. Hluchy, W. Funika, R. Wismüller, J. Kitowski, and J. Marco for discussions and suggestions, and to P. Nowakowski for his comments. This research is partly funded by the European Commission the IST-2001-32243 Project CrossGrid.

References

1. Bubak, M., M. Malawski, K. Zajac: Towards the CrossGrid Architecture. In: D. Kranzmueller, P. Kacsuk, J. Dongarra, J. Volker (Eds.) Recent Advances in Parallel Virtual Machine and Message Passing Interface, Proc. 9th European PVM/MPI Users' Group Meeting, Linz, Austria, September/October 2002, LNCS 2474, pp. 16–24.
2. CrossGrid - Development of Grid Environment for Interactive Applications. Annex 1 – description of Work. <http://www.eu-crossgrid.org>

3. Balis, B., Bubak, M., Funika, W., Szepieniec, T., and Wismüller, R.: An Infrastructure for Grid Application Monitoring In: D. Kranzlmüller, P. Kacsuk, J. Dongarra, J. Volker (Eds.) Recent Advances in Parallel Virtual Machine and Message Passing Interface, Proc. 9th European PVM/MPI Users' Group Meeting, Linz, Austria, September/October 2002, LNCS 2474.
4. Bubak, M., Funika, W., and Wismüller, R.: The CrossGrid Performance Analysis Tool for Interactive Grid Applications. In: D. Kranzlmüller, P. Kacsuk, J. Dongarra, J. Volker (Eds.) Recent Advances in Parallel Virtual Machine and Message Passing Interface, Proc. 9th European PVM/MPI Users' Group Meeting, Linz, Austria, September/October 2002, LNCS 2474.
5. Bubak, M., Marco, J., Marten, H., Meyer, N., Noga, N., Sloot, P.M.A., and Turała, M.: CrossGrid – Development of Grid Environment for Interactive Presented at PIONIER 2002, Poznan, April 23-24, 2002, Proceeding, pp. 97–112, Poznan, 2002
6. CrossGrid Project: Deliverables of first year of the project: <http://www.eu-crossgrid.org>
7. DataGrid Project: <http://www.eu-datagrid.org>
8. Foster, I., Kesselman, C., Nick, J.M., and Tuecke, S.: The Physiology of the Grid. An Open Grid Services Architecture for Distributed Systems Integration, January 2002, <http://www.globus.org>
9. Global Grid Forum: <http://www.ggf.org>
10. CrossGrid Software Repository <http://gridportal.fzk.de>