Interactive Fusion Simulation and Visualisation on the Grid

Herbert Rosmanith, Jens Volkert
GUP Linz,
Johannes Kepler University Linz,
Altenbergerstr. 69,
A-4040 Linz, Austria/Europe

Ruben Valles, Fermin Serrano
Institute for Biocomputation and
Physics of Complex Systems (BIFI),
Corona de Aragon 42,
50009 Zaragoza, Spain

Marcin Plociennik, Michal Owsiak
Poznan Supercomputing and
Networking Center (PSNC),
ul. Noskowskiego 10,
61-704 Poznan, Poland

Abstract

*This paper describes the "Fusion Plasma Application" which is developed within the "Interactive European Grid" (*int.eu.grid*), making use of its architecture to produce interactive visualisation of a distributed simulation executing on the grid. In properly combining and, where necessary, adopting the building blocks "Migrating Desktop" (MD), "Roaming Access Server" (RAS), "Grid Visualisation Tool" (GVid) and "Grid Interactive Tool" (glogin), the* int.eu.grid *adds extensions for interactive visualisation to grid middleware in a novel way.*

1 Introduction

When running grid[1] jobs, the common approach is to submit, execute and, after the job has finished, analyse the result. When visualisation is needed, this can result in transferring all of the output from the grid to a local visualisation device. Modification of parameters of a running grid job is not possible at the time it is executing in this scenario. When changing parameters, the whole job has to be submitted again. Not only is this a waste of computing resources, but also requires the user to possibly wait hours for the result. Examining the behaviour of a grid job is possible only in a post-mortem fashion, interacting with the job while it is running is impossible. To find a solution for these circumstances, the "Interactive European Grid" (*int.eu.grid*) was formed. The task of this initiative is to provide support for demanding interactive applications. To accomplish this, the "Grid Interactive Desktop", which provides user-friendly access to the grid, has been extended with powerful visualisation possibilities. This will enable researchers to receive answer from the grid in fractions of a second, not in hours.

The application chosen to demonstrate the power of this infrastructure is the "Fusion Plasma Application". It simulates the thermonuclear process taking place in fusion devices, such as, for example, the TJ-II Stellerator [2][3] or the the planned "International Thermonuclear Experimental Reactor" (ITER) [4]. The advocates of nuclear fusion hope to solve the energy problems of mankind in a clean and environment-friendly manner. The fusion application has been ported to the *int.eu.grid* infrastructure, allowing it to make use of the high number of nodes in the grid and thus calculating the data of a great number of particles. Additionally, the simulation parameters can be changed interactively, allowing to view the results of the modification in almost real time.

This paper is organised as follows: Section 2 covers the evolution of the fusion application, Section 3 describes the architecture and the components of the *int.eu.grid* and the application executing on top of it. An overview of related work is given in Section 4, before a summary and an outlook on future work concludes the paper.

2 The Fusion Plasma Application

Generally speaking, the fusion application can be subdivided into two parts: a graphical interface and a computing core. In the original fusion application, which ran only locally, the GUI was implemented by using the "Fox Toolkit" and OpenGL to represent data retrieved from the computing core graphically.

The task of the computing core is to calculate the trajectories of particles inside the fusion core in a determined time, depending on initial parameters, such as velocity and position of each particle, the definition of the vacuum chamber, and the density of the particles inside the device. The sum of particles as a whole is called a "plasma".

The calculation of the trajectory of different particles during the simulation is solved using stochastic differential equations, which govern the evolution of the plasma. The computing core has to follow a great number of particles trajectories, to obtain averages of relevant magnitudes required to represent the features of the plasma simulation. Needless to say, that this requires enormous computing power. Therefore, solving this problem in the grid has become a necessity.

2.1 The Computing Core as a batch job

The computing core was successfully ported to the grid and has been running on the EGEE [5] infrastructure. Important results have been generated. Due to the batch job nature of this solution, the previously tight coupling of the visualisation and the computing core had to be disjoined. However, it is important to see how the particle trajectories behave during the simulation, to change related parameters, to steer the computing core and to obtain the results of the specific features of the simulation in each point in time. Therefore, joining together the visualisation and the computing parts of the program in a way similar to the original locally executing application is necessary.

2.2 From batch to *int.eu.grid*

The use case of the fusion application inside the orignal interactive framework and the computing requirements demanded match perfectly with the services offered by the *int.eu.grid*. However, porting the fusion application requires some reorganisation of the software. The previously monolithic program was split into parts executing remotely and a local part, handling visualisation and the GUI. For remote execution, the computing core was re-implemented as a MPI application. The main objective is to distribute the calculation of particle trajectories to as many "worker nodes" (WNs) as possible, allowing to obtain more detailed results of the simulation. The assumption here is that the different trajectories don't interact. This issue is already taken into account inside the resolution of the equation, thanks to a parameter which represents a set of particles with the same density and temperature. This parameter is defined before starting the simulation. While MPI slave processes are busy doing calculations on the WNs, the MPI master process collects the datastreams from the slaves and compiles them, in a synchronised way, for visualisation. Since the MPI master process is executing on the grid, a method of interactively sending visualisation data and exchanging GUI events is needed. The solution to this problem is the covered in the next Section.

3 The *int.eu.grid* Interactive Architecture

Figure 1 shows how the fusion application fits into the *int.eu.grid* architecture. The application was redesigned, splitting it into a visualisation client, and into one or more MPI jobs remotely executing the fusion computing core. Both parts make use of the *int.eu.grid* extensions, which add support for interactive visualisation to the grid middleware.

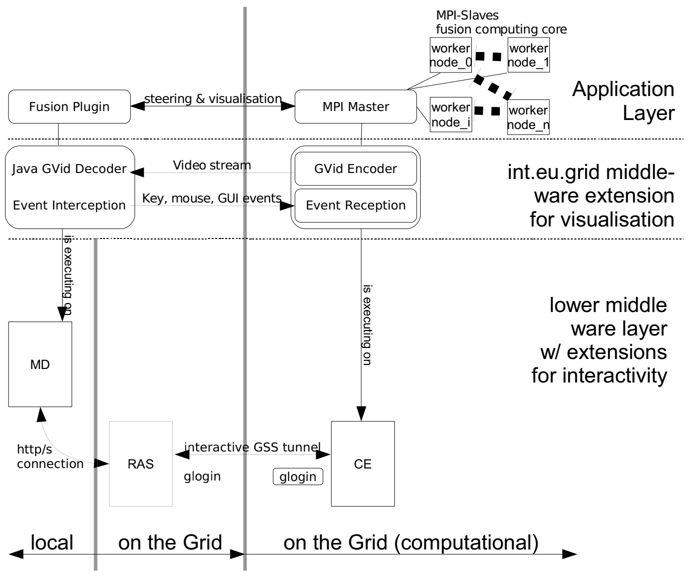

Figure 1. System Architecture of Fusion Plasma Application in *int.eu.grid*

In this figure, the horizontal axis represents networked communication of peers talking the same protocol, while on the vertical axis, the transition between the software layers are shown.

The significant parts of the *int.eu.grid* extensions are:

- the "Migrating Desktop" (MD)
- the "Roaming Access Server" (RAS)
- the "Grid Visualisation" (GVid)
- the "Grid Interactive Tool" (glogin)

An overview of these parts follows:

3.1 Migrating Desktop

The "Migrating Desktop" (MD) [6][7] can be tought as a user-friendly graphical shell, acting as a frontend for the user to access and execute jobs on the grid and to store and move data on grid storage elements. It provides built-in tools (such as a "Job Wizard", a "Grid Explorer") to allow users to work with the MD out of the box. MD is written purely in Java, making it platform independent. Since it does not access the grid "directly", that is, executing grid command line tools, but over the network my means of the Roaming Access Server, it does not depend on grid middleware being installed locally, and thus can allow accessing the grid even from handhelds, PDAs or embedded devices – assumed that the Java Runtime Environment for grid based computation. A big advantage of the MD is its extensibility, allowing developers to add applications by using well-defined interfaces for plug-ins. Thus, the MD can easily adopt external modules built for grid based computation.

3.2 Roaming Access Server

The Migrating Desktop is just a front end to the "Roaming Access Server" (RAS) [8], which intermediates the 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 is based on web services technology.

The RAS is a set of web services that mainly provide grid functionality. It is the integration level for interfaces of different middleware. It provides services like job submission service, job monitoring service, interactive session manager and interactive job channels forwarding service.

In order to support Grid data-intensive applications and to give easy and intuitive access to grid data, the "User Virtual Directory" has been created. This is an abstract filesystem that contains information about all user files independently of their physical location. Each branch in the Virtual Directory tree can be physically placed on a different location or can even be placed on storage with different way of accessing (e.g. ftp, gridftp or any other project native protocol). The Virtual Directory was designed to standardise data access, and to create a user-friendly, uniform view of the Grid and local resources. The Virtual Directory is currently compatible with the "Logical File Catalogue" (LFC) [9].

3.3 Grid Visualisation

The CrossGrid project [10] was the predecessor of *int.eu.grid*. It was oriented towards compute- and data-intensive applications which require the interaction with a person in a processing loop. As part of CrossGrid, the GVK [11] was examined. The topic of research was where to split the visualisation pipeline, so that parts execute on the grid, connected via the network with the rest of the pipeline executing locally. The pipeline starts with an application producing simulation data, the visualisation mapping process data will generate "Abstract Visualisation Objects" (AVO), producing intermediate geometry data, which are transformed into a displayable image by the rendering process. It turned out that, when splitting the visualisation pipeline after this process, the amount of geometry data generated by realistic simulations is too big to be transmitted over the network in realtime. Therefore, it has been suggested to shift the generation of images to the grid, and transfer ready rendered images only, eventually compressing each frame before transmission. The "Grid Visualisation Tool" (GVid) [12] pushes this idea even further and produces a compressed video stream by using the XviD library [13], which creates a discrete cosine transformed (DCT) data stream. An efficiently and thus highly compressed video stream is transferred through the network and displayed on one or more thin video clients, while the feed back is from a single master user, who interacts with the application.

Using this technique, GVid is able to provide an advanced Grid video service, which is capable of securely transporting visual data originating from arbitrary OpenGL and X11 applications, but can also be used in programs which adopt their visualisation routines to the GVid video transport library.

In *int.eu.grid*, the GVid has been extended to conform to the requirements of MD plug-ins, which involved creating a Java implementation of the GVid display client. This resulted in a Java package ready to be used by grid application developers.

3.4 Grid Interactive Tool

The "Grid Interactive Tool" (glogin) [14][15] was originally developed for the "Globus Toolkit 2" (GT2) [16]. Recently, support for web-services has been added, allowing *glogin* to execute on GT4 middleware. To overcome the limitations of the "Globus Gatekeeper" (an authorisation process deciding about access to the grid), *glogin* opens a separate communications channel. This channel is created such that it is within the firewall's port range. *glogin* invokes a remote copy of itself on the grid, using executable-staging when instructed. To speed up transmission of connection parameters, the remote *glogin* process forces emptying of the GASS-cache [17] by invoking a child-process and terminating the parent, since unfortunately, the GASS-cache offers no method for flushing its contents. Both *glogin* processes create a connection listener and try to reach each other, the connection established first will be kept, while the losing connection will be removed. The remaining connection is authorised using the standard grid authentication mechanism. An optional program is started on the "Computing Element" (CE) and its IO-traffic is intercepted. This traffic is secured by using methods of the GSS-API [18][19] and transported over the connection. Locally, the traffic can be received by the program starting *glogin*, allowing further processing (such as, for instance, video stream decoding with GVid).

In *int.eu.grid*, *glogin* has been modified to fit the needs of this infrastructure. The "local" *glogin* process is started by RAS, since the MD is the only frontend the user should be bothered with. For ease of development, *glogin* no longer invokes a remote copy of itself, but informs the RAS about its connection endpoint. When submitting a grid job, RAS will store the connection address in the job descriptor file, making it available to the remote *glogin* process. This makes the previously mentioned "connection establishment race" unnecessary. Since now there is no need for the remote *glogin* process to flush its information about connection parameters, the previously mentioned method is not in effect. This way, problems in the middleware which became evident [20] can be avoided.

Additional arguments for input and output redirection have been added to *glogin*, too. By specifying "named pipes" for input and output, the RAS is able to reconnect to *glogin* in case of a software failure. In the presence of a named input pipe, *glogin* does not terminate the communication channel in case of an end-of-file condition, but rather restarts the IO-operation. This implies that *glogin* does not know when to terminate the program it is executing on the CE either. Therefore, the end-of-job condition has to be exchanged at the upper software layers.

3.5 Putting the pieces together

A first release of joining these software pieces and porting the fusion plasma application to this new infrastructure is already available, although development is still in progress. In Figure 2, a screenshot of the "Fusion Plugin" executing in the MD can be seen.

Physical parameters and other simulation data can be changed from this graphical interface. This can be done initially at submission time, but also during the execution of the program. Examples of properties that can be changed are:

- number of particles to simulate
- place, speed and direction when launching particles
- whether collisions are allowed
- if the electrical field is activated

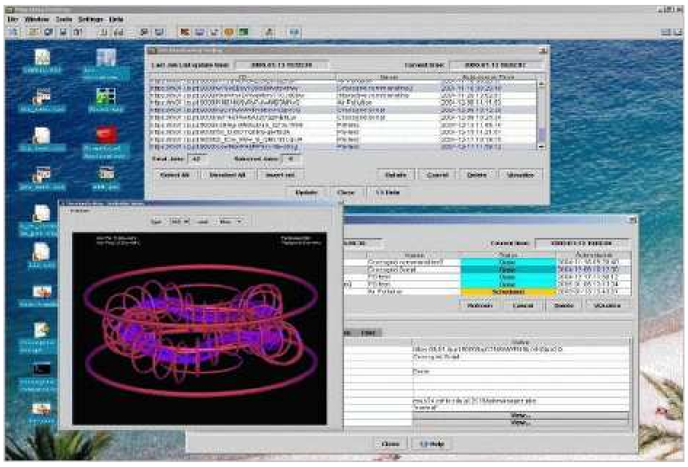

Figure 2. View of Fusion Plasma Application running in the Migrating Desktop

4 Related Work

The "Bloodflow Visualisation" (BFVS) [21][22] developed in the CrossGrid Project aimed at grid-based interactive visualisation, too. By allowing a human in the processing loop, interactivity was introduced. However, the time required for once cycle spent in the interaction loop was substantially long. The intermediate geometry data produced by the visualisation toolkit used (VTK [23]) was stored on disk, to be transferred to a local workstation for offline visualisation later. Data transfer was carried out by using gridftp. As a result, the responsiveness of the system was very low, as was the amount of interactivity achieved. In contrast to this, *int.eu.grid* offers real time interactive visualisation.

With respect to interactivity, several utilities have been built. One example is the ubiquitious "secure shell" (SSH) [24]. In the grid environment, the SSH has been expanded to support grid related authentication, resulting in a GSI enabled SSH (GSI-SSH). Within the Globus Toolkit 4 (GT4), the GSI-SSH is now available by default. The disadvantage of this approach is that a GSI aware ssh server (ssh daemon, sshd) needs to be installed, configured and activated on the target host. In contrast to this, *glogin* represents an ultra-light weight approach with no additional installation requirements. Additionally, executing jobs by means of GSI-SSH results in access to the host without being subject to limitations such as grid job policies and scheduling. Therefore, GSI-SSH cannot be used as a replacement for the usual job submission mechanism.

The "Interactive Grid Architecture for Application Service Providers" (I-GASP) [25] aims to provide interactivity for the grid. It uses the "Virtual Network Computing"

(VNC) [26] to display graphical output. Since VNC was not designed for transmitting compressed video streams, it cannot work as efficiently as the solution presented in this paper.

The many grid portals which have been built can roughly be compared to the Migrating Desktop. For instance, they offer job submission and monitoring services, functions which can be found in the MD, too. Grid portals often announce an "interactive environment", however, the interactivity offered usually only refers to the portal software itself, as can be seen in e.g. the "Grid ENabled Interactive Environment" (GENIE) [27]. In contrast to this, the *int.eu.grid* provides interactivity not only for a grid environment, but also for the grid jobs.

5 Conclusion and Future Work

Using *int.eu.grid*, the Fusion Plasma Application can simulate a great number of ion trajectories in a reasonably short period of time. Adding interactivity, the physical parameters describing the plasma (such as density, temperature and electric field) can be modified during the simulation. The joint benefit of interactivity and visualisation enables the user to instantly recognise which effect which changes in the physical parameters of the model have on the individual orbits of thousands, possibly millions of ions. In devices with such a complex magnetic configuration as TJ-II, this is a valuable information when studying the transport.

We have shown the architecture of the *int.eu.grid* and how it can be applied to support demanding interactive applications. For verification of the correctness and usefulness of *int.eu.grid*, the Fusion Plasma Application has been ported to this infrastructure. The power of the *int.eu.grid* results in the proper combination of its basic building blocks. By demonstrating this power, we hope that the *int.eu.grid* will attract the attention of current and future research projects.

6 Acknowledgments

We thank our colleagues for giving their valuable help and expertise to this project: Paul Heinzlreiter, Thomas Köckerbauer, Markus Baumgartner, Bartosz Palak, Sven Stork, Enol Fernandez, Elisa Heymann, Miquel A. Senar, Stuart Kenny, Brian Coghlan and Isabel Campos Plasencia. Additionally, we want to thank Paul Heinzlreiter for proofreading.

The work described in this paper was supported in part by the European Union through the FP6-2005-Infrastructures-7- contract No 031857 project "Int.eu.grid".

References

[1] Ian Foster, Carl Kesselman (editors): *The Grid: Blueprint for a New Computing Infrastructure*, Morgan-Kaufman, 1999.

[2] F. Castejon, J. M. Reynolds, J. M. Fontdecaba, R. Balbin, J. Guasp, D. Lopez-Bruna, I. Campos, L. A. Fernandez,7 D. Fernandez-Fraile, V. Martin-Mayor, A. Tarancon: *Ion Orbits and Ion Confinement Studies on ECRH Plasmas in TJ-II Stellerator*, Fusion Science and Technology, 50(3), pp. 412-418, 2006

[3] F. Castejon, L. A. Fernandez, J. Guasp, V. Martin-Mayor, A. Tarancon, J. L. Velasco: *Ion kinetic transport in presence of collisions and electric field in TJ-II*, 33rd European Physical Society Conference on Plasma Physics, Roma, 2006

[4] *ITER: International Thermonuclear Experimental Reactor*, http://www.iter.org

[5] *EGEE: Enabling Grids for E-sciencE*, http://www.eu-egee.org

[6] Miroslaw Kupczyk, Rafal Lichwala, Norbert Meyer, Bartosz Palak, Marcin Plociennik, Maciej Stroiski, and Pawel Wolniewicz: *The Migrating Desktop as a GUI Framework for the "Applications on Demand"*, Concept M. Bubak et al. (Eds.): ICCS 2004, LNCS 3036, pp. 91-98, 2004. Springer-Verlag Berlin Heidelberg, 2004

[7] Miroslaw Kupczyk, Rafal Lichwala, Norbert Meyer, Bartosz Palak, Marcin Plociennik, Maciej Stroiski, Pawel Wolniewicz: *The Migrating Desktop - the General Entry Point to the Grid*, 6th CARNET Users Conference, pp. 27-29 September 2004, Zagreb, Croatia

[8] Miroslaw Kupczyk, Rafal Lichwala, Norbert Meyer, Bartosz Palak, Marcin Plociennik, Pawel Wolniewicz: *Roaming Access and Migrating Desktop*, Conference materials - Cracow '02 Grid Workshop, December 11-14, 2002, Krakow, Poland, pp. 148-154

[9] Sophie Lemaitre: *The LCG File Catalogue (LFC) General Description*, https://uimon.cern.ch/twiki/bin/view/LCG/ Lfc-GeneralDescription

[10] Marian Bubak, Jesus Marco, Holger Marten, Norbert Meyer, Marian Noga, Peter A.M. Sloot, and Micha Turala: *CrossGrid - Development of Grid Environment for Interactive Applications*, EU Project, IST-2001-32243, Technical Annex. http://www.eu-crossgrid.org

[11] Dieter Kranzlmüller, Paul Heinzlreiter, Herbert Rosmanith, Jens Volkert: *Grid-Enabled Visualization with GVK*. European Across Grids Conference 2003: pp. 139-146

[12] Thomas Köckerbauer, Martin Polak, Thomas Stütz, and Andreas Uhl: *GVid - video coding and encryption for advanced Grid visualization*, in J. Volkert, T. Fahringer, D. Kranzlmüller, and W. Schreiner, editors, Proceedings of the 1st Austrian Grid Symposium, volume 210 of books@ocg.at, pp. 204-218, Schloss Hagenberg, Austria, 2006. Austrian Computer Society.

[13] *The XviD project homepage*, http://www.xvid.org

[14] Herbert Rosmanith, Jens Volkert: *glogin - Interactive Connectivity for the Grid*. Distributed and Parallel Systems: Cluster and Grid Computing (DAPSYS 2004), Austrian-Hungarian Workshop on Distributed and Parallel Systems, pp. 3-12, Budapest, Hungary, 2004.

[15] Herbert Rosmanith, Jens Volkert: *Traffic Forwarding with GSH/GLOGIN*. 13th Euromicro Workshop on Parallel, Distributed and Network-Based Processing (PDP 2005), pp. 213-219, Lugano, Switzerland, 2005

[16] Ian Foster, Carl Kesselman: *Globus: A metacomputing infrastructure toolkit.* The International Journal of Supercomputer Applications and High Performance Computing **11**(2) (1997), pp. 115-128

[17] Joseph Bester, Ian Foster, Carl Kesselman, Joseph Tedesco, Steven Tuecke: *GASS: A Data Movement and Access Service for Wide Area Computing Systems*, Sixth Workshop on I/O in Parallel and Distributed Systems, May 5, 1999.

[18] John Linn: *Generic Security Service Application Program Interface - Version 2, Update 1,* Internet Engineering Task Force, RFC 2743, January 2000

[19] John Wray: *Generic Security Service API Version 2: C-bindings*, Internet Engineering Task Force, RFC 2744, January 2000

[20] Herbert Rosmanith, Peter Praxmarer, Dieter Kranzlmüller, Jens Volkert: *Towards Job Accounting in Existing Resource Schedulers: Weaknesses and Improvements,* High Performance Computing and Communication (HPCC 2006), pp. 719-726, Munich, Germany, 2006

[21] Alfredo Tirado-Ramos, Hans Ragas, Denis P. Shamonin, Herbert Rosmanith, Dieter Kranzlmüller: *Integration of Blood Flow Visualization on the Grid: The FlowFish/GVK Approach*. European Across Grids Conference 2004: pp. 77-79, Nicosia, Cyprus, 2004.

[22] Peter A.M. Sloot, Dick G. van Albada, Elena Zudilova, Paul Heinzlreiter, Dieter Kranzlmüller, Herbert Rosmanith, Jens Volkert: *Grid-based Interactive Visualisation of Medical Images*, S. Norager (editor), Proceedings of the First European HealthGrid Conference, January 203, pp. 57-66. Commission of the European Communities, Information Society Directorate-General, Brussels, Belgium, 2003

[23] *The Visualisation Toolkit*, http://www.vtk.org

[24] Tatu Ylönen: *SSH Secure Login Connections over the Internet*, Sixth USENIX Security Symposium, pp. 37-42 of the Proceedings, SSH Communications Security Ltd. 1996.

[25] Sujoy Basu, Vanish Talwar, Bikash Agarwalla, Raj Kumar: *Interactive Grid Architecture for Application Service Providers*, Mobile and Media Systems Laboratory, HP Laboratories Palo Alto, Technical Report, July 2003

[26] Tristan Richardson, Quentin Stafford-Fraser, Kenneth R. Wood and Andy Hopper: *Virtual Network Computing*, IEEE Internet Computing, 2(1) pp. 33-38, 1998

[27] *GENIE: Grid ENabled Interactive Environment*, http://www.npaci.edu/online/v5.5/genie.html