Experiences with MPI Application Development within int.eu.grid: Interactive European Grid Project

B. Krammer

HLRS - High Performance Computing Center Stuttgart,
Nobelstrasse 19, 70569 Stuttgart, Germany
email: krammer@hlrs.de
phone: (+49 0711) 6855890, *fax:* (+49 0711) 6855832

Abstract

Nowadays, a growing number of researchers and application developers profit by Grid infrastructures to run complex compute- and data-massive applications in a distributed way to get results in a reasonable time period. Applications come from many different scientific areas, such as medicine, physics or environment, etc. The European Grid-project int.eu.grid addresses the challenge of offering a production-level e-Infrastructure to interactive (MPI) applications. Within this project, the MPI libraries Open MPI and/or PACX-MPI are employed. Application support is also available through the MPI verification tool MARMOT, which will be described in this paper.

1 Introduction

Today's Grid technology [12] is advanced enough to offer users distributed computing power for large simulation runs in a more or less convenient way. Thus, many researchers and application developers profit by Grid infrastructures to run their complex compute- and data-massive applications in a distributed way to get results in a reasonable time period. Applications come from many different scientific areas, such as medicine, physics or environment and other research areas (from robotics to archaeology), and have increasing demand for grid-enhanced interactive use. However, it is, in many cases, still rather cumbersome to run interactive applications in a Grid environment, e.g. to run scientific simulations, visualize results and interactively change parameters. Another challenge is minimizing response times so that results can be obtained in almost real-time.

The objective of the European int.eu.grid project [5] is to deploy and operate an interoperable production-level e-Infrastructure for demanding interactive applications. This scientific initiative offers application developers access to distributed parallel (MPI [1, 2]) interactive computing and storage at the tera level through a user-friendly Grid interactive desktop with powerful visualization. While guaranteeing interoperability with existing large e-Infrastructures such as

EGEE [7], the initiative will exploit the expertise generated by the EU CrossGrid project [6].

Applications from three different research areas will be considered in the first phase of the project: medicine (e.g. ultrasound computer tomography, including real-time data storage, assisted diagnosis on brain images and support for clinical VO), physics (e.g. High Energy Physics, Astrophysics, and Fusion) and environment modelling (e.g. pollution simulation).

2 MPI within int.eu.grid

In int.eu.grid's predecessor project CrossGrid, the MPI library mpich [13] and the grid-enabled library mpich-g2 [14], respectively, were employed. On the int.eu.grid testbed, we are now using Open MPI [3, 15], which is a completely new MPI implementation that has been developed from scratch during the last years, combining technologies and resources from several other previous projects such as Lam-MPI, FT-MPI, LA-MPI, PACX-MPI. A growing number of partners world-wide is involved in the implementation of this Open Source project. Open MPI is a fully MPI-2 standard compliant implementation that supports a large number of platforms and is being used in a number of projects.

In addition to Open MPI, int.eu.grid aims at using the grid-enabled PACX-MPI library [4]. PACX-MPI enables users to run an MPI application across homogeneous, but also heterogeneous testbed sites. For the site-internal communication, a locally installed MPI library can be used, i.e. in our case Open MPI.

3 MPI support with MARMOT

Due to the complexity of parallel programming there is a need for tools supporting the MPI development and porting process. There are many situations where incorrect usage of MPI by the application programmer can automatically be detected. Examples are the introduction of irreproducibility, deadlocks and incorrect management of MPI resources such as datatypes, etc. Furthermore, the MPI standard leaves many decisions to the implementation, e.g. whether or not a standard communication is blocking. This implementation-defined behaviour may cause problems when porting an application from a local platform to the int.eu.grid testbed with a possibly different environment, regarding e.g. compilers or the MPI implementation to be used.

MARMOT [8, 9] is a library that uses the so-called PMPI profiling interface to intercept MPI calls and analyse them during runtime. It just has to be linked to the application in addition to the underlying MPI implementation, without any modification of the application's source code nor of the MPI library. The tool checks if the MPI API is used correctly and checks for errors frequently made in MPI applications, e.g. deadlocks, the correct construction and destruction of resources, etc. It also issues warnings for non-portable behaviour,e .g. using tags outside the range guaranteed by the MPI-standard. The output of the tool is available in different formats, e.g. as text log file or html/xml, which can be displayed and analysed using a graphical interface. MARMOT is intended to be a portable tool that has been tested on many different platforms and with different MPI implementations.

MARMOT supports the complete MPI-1.2 standard for C and Fortran applications and is being extended to also cover MPI-2 functionality [10, 11]. As it is only fully defined in the MPI-2 standard, the C++ language binding is currently not yet supported by MARMOT, but it is possible to run MARMOT with C++ applications that use the C-style language binding. MARMOT itself is written in C++ using the MPI C language binding. The only requirement to successfully run C++ applications with MARMOT and Open MPI is to build MARMOT and the application with the `-DOMPI_SKIP_MPICXX` compiler flag.

Being a piece of MPI code itself, portability problems also arise in the development of the MARMOT tool. In this case, adding support for Open MPI to MARMOT required some changes in MARMOT, as, for example, so-called "opaque" MPI resources such as communicators, datatypes, etc. are implemented differently in Open MPI than e.g. in mpich. Another difference is the implementation of Fortran wrapper functionality within Open MPI, and subsequently, in MARMOT.

MARMOT has been used successfully in the CrossGrid environment, and can similarly also be used in int.eu.grid. For the job submission, middleware developed by other tasks can be used, such as the CrossBroker [18] or Migrating Desktop [16], or glogin [17]. With the help of the CrossBroker it is possible to submit jobs to any site in the testbed that seems suited best. The testbed sites are spread among several European countries. The CrossBroker offers support for serial and for MPI jobs using mpich, mpich-g2, Open MPI and PACX-MPI, for interactive as well as for batch jobs. Migrating Desktop is a graphical user interface intended to offer easy access to Grid services such as job and file management. Support for applications and tools is provided by plugins. Migrating Desktop is platform independent as it is based on Java technology. Fig. 1 shows an application running on the Grid with Marmot; this Grid job has been submitted via Migrating Desktop from the user's desktop and can be monitored and steered at runtime.

Glogin is an interactive grid service that offers interactive bidirectional connections between a grid resource and the user's desktop machine. It proves useful in transmitting arbitrary data in and out of the Grid, e.g. for interactive visualization of results using the so-called GVid Grid-enabled video-service. Glogin also offers an interactive shell access to Grid resources, similar to an `ssh` command.

4 Applications

Applications are supported at all levels: adaptation of the legacy application to the MPI and Grid framework, access to a testbed for trials and pre-deployment, monitoring tools to assess the impact of executing in a Grid

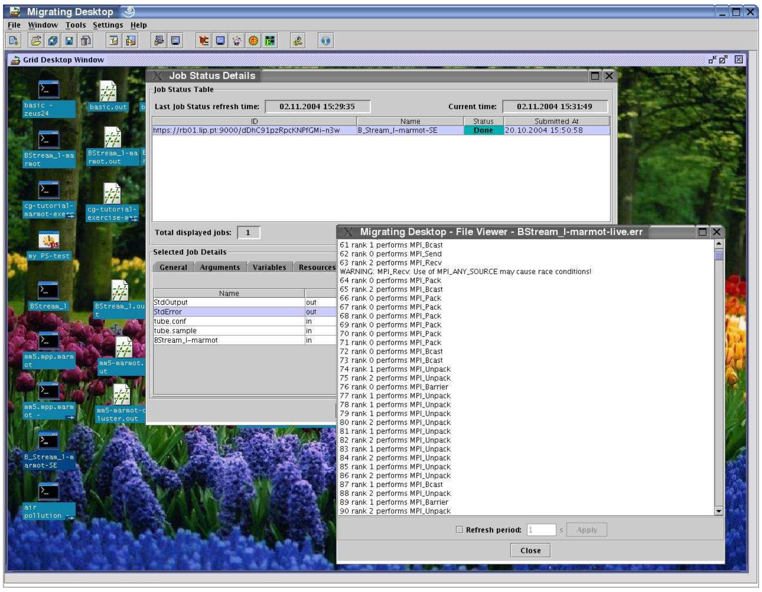

Figure 1: Watching a Grid application with MARMOT via Migrating Desktop

framework, support to setup a corresponding Virtual Organization, etc. The int.eu.grid project comprises a number of applications from different areas, e.g. medicine, physics or environment. Examples are

- a medical imaging application based on the reconstruction by numerical techniques of an image, using as input the data measured by an Ultra Sounds Scanner; these data are distributed for analysis over the grid
- a medical application for diagnostic on on brain images : 50 sequential images are obtained in a CT scan, and using as reference two structures defined by the doctor, three parameters are obtained for each voxel, and used to precisely define the brain zone affected. The application allows visualization and annotation of DICOM images, the three corresponding brain maps, and the final diagnostic map. Moreover, the interactive session can be shared with another doctor or medical team.
- a flood forecasting application, which tries to predict oncoming floods based on series of simulations of meteorological, hydrological and hydraulic conditions in the target area, including several simulation steps (with more choices of used models for each step), attached pre-processing and post-processing and visualization tools; interaction with workflow according to output data of previous simulation is possible
- pollution modelling as the movement of individual independent particles (Lagrangian model), distributed among a number of processors, which calculate the individual trajectories. For every particle the position is stored

in a file which is used for interactive visualization purposes.

- visualisation of Wave Model BaltWAM runs, possibly with running several model setups with different parameters in parallel

Some of the applications in the int.eu.grid project have already been part of the CrossGrid project. Experiences verifying these applications with the MARMOT tool have been described for example in [8].

One new application is the Fusion Application, which visualizes the behaviour of plasma inside the magnetic field of a Fusion Reactor. The plasma is analyzed as a many body system consisting of N particles, distributed among a number of processors, which calculate the individual trajectories independently. For every individual particle, and at every step of the algorithm, the position, velocity, and other magnitudes of the particle are stored. A graphical interface allows to interactively visualize the plasma trajectory in the reactor and to interactively change simulation physical parameters such as the velocity and direction of particles, electromagnetic fields, number of particles, etc., see fig. 2.

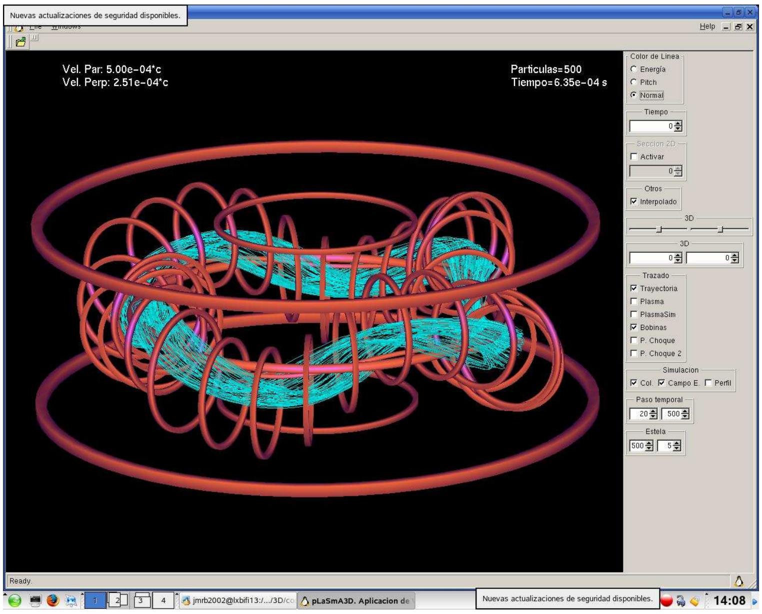

Figure 2: Visualization of Plasma in Fusion Reactors

This is a challenging task because of the great amount of data and the short response times that are required to visualize and interact with the application in almost real time. For the visualization, all processes need to transmit the information related with the position and velocity of the particles in a continous and synchronized way to a central machine, which will send the output as a video stream to the client's desktop, after rendering. This is implemented as a master-slave-algorithm using MPI.

The fusion application is written in C++ but can be run using Open MPI and MARMOT without a problem as it is using C language binding and MPI-1.2 calls only. One error that can be identified with MARMOT is a deadlock caused by some processors hanging in `MPI_Finalize`, due to an `exit(0)` construct before calling `MPI_Finalize` on one of the processes.

5 Conclusions

In this paper, we have given an introduction to the European Grid project int.eu.grid with special respect to the MARMOT MPI verification tool. The int.eu.grid project aims at offering distributed resources with powerful visualization to researchers in a user-friendly and interactive way. Application developers working on big or complex scientific problems are encouraged to run their compute- and data-massive applications on the int.eu.grid infrastructure to get results in a reasonable time period.

MARMOT is a useful verification tool in the process of MPI application development. Future work on the tool includes full support for MPI-2, and full integration into the int.eu.grid framework including PACX-MPI support.

Acknowledgements

The research presented in this paper has partially been supported by the European Union through the IST-031857 project "int.eu.grid" (may 2006 – april 2008). Special thanks go to the developers of the applications. MARMOT is being developed by HLRS in collaboration with ZIH Dresden.

References

1. Message Passing Interface Forum. *MPI: A Message Passing Interface Standard*, June 1995. `http://www.mpi-forum.org`.
2. Message Passing Interface Forum. *MPI-2: Extensions to the Message Passing Interface*, July 1997. `http://www.mpi-forum.org`.
3. E. Gabriel, G. E. Fagg, G. Bosilca, T. Angskun, J. J. Dongarra, J. M. Squyres, V. Sahay, P. Kambadur, B. Barrett, A. Lumsdaine, R. H. Castain, D. J. Daniel, R. L. Graham, and T. S. Woodall. Open MPI: Goals, Concept, and Design of a Next Generation MPI Implementation. In *11th European PVM/MPI*, volume LNCS 3241, pages 97–104. Springer, 2004.
4. R. Keller, B. Krammer, M.S. Müller, M. M. Resch, and E. Gabriel. Towards efficient execution of MPI applications on the Grid: Porting and Optimization issues. *Journal of Grid Computing*, Vol. 1 (2), pp. 133-149, 2003.
5. int.eu.grid: Interactive European Grid Project - EU IST Project (031857). `http://www.interactive-grid.eu/`
6. crossgrid: EU Project (IST-2001-32243). `http://www.crossgrid.org`
7. EGEE: Enabling Grids for E-sciencE. `http://www.eu-egee.org/`

8. B. Krammer, M. S. Müller, and M. M. Resch. MPI Application Development Using the Analysis Tool MARMOT. In *ICCS 2004*, volume LNCS 3038, pages 464–471. Springer, 2004.
9. B. Krammer, M. S. Mueller, and M. M. Resch. Runtime checking of MPI applications with MARMOT. In *ParCo 2005*, Malaga, Spain, September 2005.
10. B. Krammer, M. S. Müller, and M. M. Resch. MPI I/O Analysis and Error Detection with MARMOT In *Proc. of EuroPVM/MPI 2004*, Budapest, Hungary, September 19-22, 2004. LNCS Vol. 3241, pp. 242 - 250, Springer, 2004.
11. B. Krammer, and M. M. Resch. Correctness Checking of MPI One-sided Communication Using MARMOT In *Proc. of EuroPVM/MPI 2006*, Bonn, Germany, September 17-20, 2006. LNCS Vol. 4192, pp. 105 - 114, Springer, 2006.
12. I. Foster, C. Kesselman (Eds.): The Grid. Blueprint for a New Computing Infrastructure. *Morgan Kaufmann Publishers*, San Francisco, 1999.
13. mpich Homepage. `www.mcs.anl.gov/mpi/mpich`
14. mpich-g2. `http://www3.niu.edu/mpi`
15. Open MPI Homepage. `http://www.open-mpi.org`
16. M. Kupczyk, R. Lichwala, N. Meyer, B. Palak, and P. Wolniewicz M. Plociennik. Mobile Work Environment for Grid Users. In *Proceedings, 1st European Across Grids Conference 2003*, volume 2970 of *Lecture Notes in Computer Science*, pages 132–138, Santiago de Compostela, Spain, 2004. Springer-Verlag.
17. H. Rosmanith, D. Kranzlmüller. glogin - A Multifunctional, Interactive Tunnel into the Grid In *Proc. Grid 2004, 5th IEEE/ACM Intl. Workshop on Grid Computing*, IEEE Computer Society, Pittsburgh, PA, USA, 266-272 (Nov. 2004).
18. E. Heymann, M. A. Senar, A. Fernandez, and J. Salt. The EU-CrossGrid Approach for Grid Application Scheduling. In *Proceedings, 1st European Across Grids Conference 2003*, volume 2970 of *Lecture Notes in Computer Science*, pages 17–24, Santiago de Compostela, Spain, 2004. Springer-Verlag.