

Building Grid client applications with the g-Eclipse framework

Paweł Wolniewicz*

Author Poznan Supercomputing and Networking Center, Poland

E-mail: Pawel.Wolniewicz@man.poznan.pl

Ariel García

Institute for Scientific Computing, Forschungszentrum Karlsruhe, Germany

E-mail: mathias.stuempert@iwr.fzk.de

Mathias Stümpert

Institute for Scientific Computing, Forschungszentrum Karlsruhe, Germany

E-mail: ariel.garcia@iwr.fzk.de

Harald Kornmayer

NEC Laboratories Europe, St. Augustin, Germany

E-mail: kornmayer@it.neclab.eu

The g-Eclipse is an integrated workbench framework to access the power of existing Grid infrastructures. It can be used as a rich client application with user friendly interface to access Grid resources, but can also be used as a base for writing customised grid application using g-Eclipse model and g-Eclipse common grid library. Scientific application can use g-Eclipse to gain access to grid resources independently on specific grid infrastructure. Applications can enhance g-Eclipse with specific application editors, viewers and submission supports or can be extended to a Rich Client Platform application using g-Eclipse as a common grid library.

XII Advanced Computing and Analysis Techniques in Physics Research

November 3-7 2008

Erice, Italy

*Speaker.

1. Introduction

Over the last few years Grid infrastructures have become the backbone of those fields of science and research that require the solution of complex computational problems. At the same time, the commercial application of Grid technologies has led to new categories of offerings (e.g. on-demand computing) and are likely to play an important role in the *Software as a Service* landscape. However, the complexity of Grid infrastructures is often discouraging to application developers and impedes the use of Grid technologies in scientific application domains. The European Commission (EC) funded g-Eclipse project is therefore building an integrated workbench framework to access the power of existing Grid infrastructures, based on top of the Eclipse Platform which is widely popular development and integration environment. g-Eclipse achieved the status of an official Eclipse project.

Using g-Eclipse framework [1], access to the power of Grid infrastructures will become more intuitive and simpler than is possible nowadays. Grid application users will be able to access the Grid with standardised, but customised user-friendly interfaces. Grid resource providers can reduce the cost of operation as the complexity of the Grid is reduced with supporting tools. Grid application developers will be empowered to speed up the development cycle of new Grid applications.

g-Eclipse is not dedicated to any specific user community, and can be used for different user needs, different infrastructures, Virtual Organisations and applications. Example application integration was shown for chemical, pharmaceutical and aerospace applications. g-Eclipse is an open source project and user community is gathered around it, which allows for sustainable development.

There are two ways to prepare and manage grid applications job with g-Eclipse: enhance g-Eclipse with specific application elements or prepare Rich Client Platform application using g-Eclipse as a common grid library.

2. g-Eclipse overview

The g-Eclipse framework provides tools to customise Grid users' applications, to manage Grid resources and to support the development cycle of new Grid applications. The project aims for general Grid workbench tools that can be extended for many different Grid middlewares (such as gLite, UNICORE, Globus Toolkit), and comes with exemplary support for the gLite and GRIA middlewares. Furthermore an adapter to Amazon's Web services (i.e. S3 and EC2) is available. As a first outcome of this development Amazon's Simple Storage Service can be easily accessed from within g-Eclipse and instances in Amazon's Elastic Compute Cloud can be controlled. With this step, g-Eclipse introduces itself to clouds and is no longer only middleware independent but Grid independent. The important thing for application developers is that they do not need to care about the specific grid middleware supports. The same application can work with different grids or infrastructures without any changes.

The g-Eclipse platform is developed as a general framework that can be used by Grid users, Grid developers and Grid operators. The software developed in the g-Eclipse project consists of *Score Grid plug-ins* for the Eclipse platform. These enable and standardise the access of Grid infrastructures from within Eclipse, independent of the Grid middleware used. The graphical

user interface follows the Eclipse UI guidelines. The intuitive interface hides the complexity of using Grid services from the end user or application developer, which results in a low entry barrier especially for users and developers new to Grid technology.

The framework integrates functionality for the following use cases:

Grid Virtual Organisation management A functionality that allows the dynamic creation and management of Virtual Organisations and their resources. This includes authentication and authorisation of users on remote Grid resources.

Grid job management One of the standard actions on the Grid is the submission of Grid jobs and their monitoring. This functionality supports Grid users and developers to create and manage Grid jobs independent of the middleware.

Grid file management The management of distributed data in a Grid environment is based on filenames, which are maintained by catalogues. The access and management of local and foreign resources are seamlessly integrated into the Eclipse Workbench.

Grid application deployment A functionality that supports Grid application developers and Grid application users with the deployment of their applications on the Grid will be developed.

Grid visualisation tools A visualisation framework offers functionalities that allow the visualisation of scientific and numerical calculations.

Grid workflow builder An increasing number of Grid applications consist of many interlinked Grid jobs forming a complex workflow. A middleware independent functionality to support the creation and management of such workflows using a graphical editor is provided.

Grid command console Many existing Grid systems use a command line interface (CLI) to interact with Grid resources and most Grid users are familiar with these CLIs. Therefore g-Eclipse delivers a command console to access the Grid.

g-Eclipse provide API and User Interface modules for all above functionalities.

3. Application inside g-Eclipse

Many of scientific grid application does not require graphical user interface. They are just command line application located in the remote grid machines and users do computation by submitting jobs to the Grid specifying applications input data. Computation results can be stored on grid for further processing or can be downloaded to local user machine. For some applications local machine can be involved in preprocessing of input files (e.g. reformatting input data) and postprocessing of result (e.g. graphical visualisation). Such kind of application can be dealt with by enhancing g-Eclipse with a set of optional elements.

Using this method application users can benefit from a user-friendly graphical interface for accessing grid infrastructure. Standard g-Eclipse way of working with Grid projects is used and only few optional elements should be added.

The easiest way is to provide XML-based application descriptions to specify application parameters in much more friendly way than just filling "Executable" and "Arguments" fields in a Grid

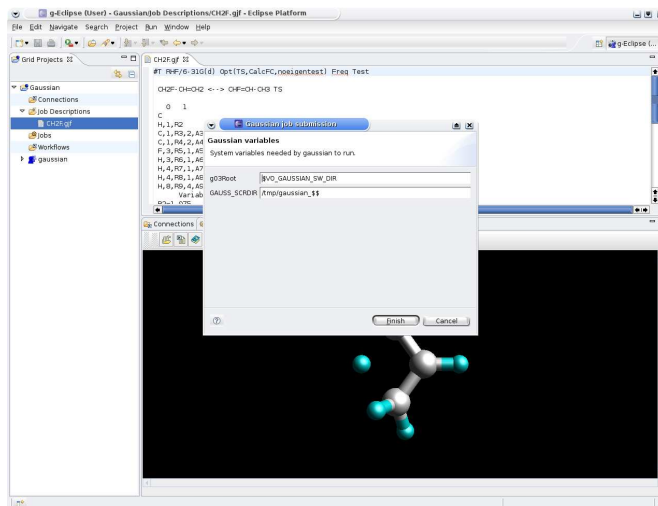


Figure 1: g-Eclipse enhanced with Gaussian input file editor and visualisation.

job description language. XML description is converted to a form of graphical dialog containing fields for entering application parameters. g-Eclipse offers tools to share such job description documents within application communities by using repositories like CVS or SVN. The same is valid for workflow description files. With the help of the g-Eclipse framework application developer or integrator manager can prepare applications, both simple or workflow, and share them with the application users group.

More advanced way of integration application can be achieved by providing more advanced plug-ins like input file editor or result visualisation. g-Eclipse provide visualisation functionality that can be used by application plug-ins. For some application also submission support plug-in is necessary to prepare proper job description and input scripts.

By embedding application into g-Eclipse developer do not need to provide client interface or grid resources management. All elements, menus and actions to submit job, manage job, update status or transfer files are provided by g-Eclipse framework.

Example integration of Gaussian [2] application was prepared. The editor for g98 formatted input files was prepared to allow users to work with the format they already know. No experience with Grid job description language is needed. g98 file can be submitted to the Grid and all necessary files, e.g. JSDL or submission script, are created on the fly. Gaussian files can be also visualised with additional Gaussian viewer. The provided plug-ins base on an open source JMol editor and are plugged into g-Eclipse after minor modifications. An example view of integrated element was shown on 1.

4. Application on top of g-Eclipse

The second approach enhances the g-Eclipse framework with new plugins for application specific job description languages and submission methods. Many applications use job description files in the specific format (e.g. running Gaussian jobs requires ".inp" file with all data). The

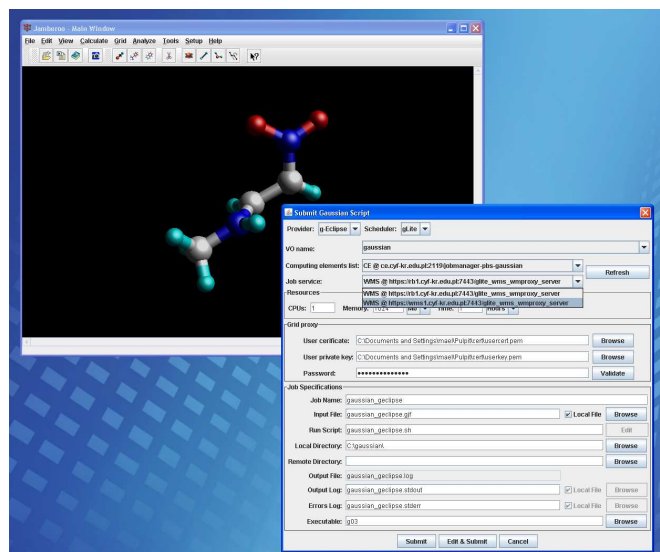


Figure 2: JMol application enhanced with Grid access.

new plugins registers such files as grid job descriptions and supporting editors (textual or graphical ones) which can handle this file format. Using the existing g-Eclipse infrastructure, such job description are submitted to the Grid by transforming the application specific description in a middleware specific one.

The last method is the most advanced and can be used to prepare custom application. By extensively using the Eclipse extension mechanism in combination with object-oriented design patterns, the framework can be easily extended by application-specific implementations. The application is not plugged into g-Eclipse, but rather built on top of it. It follows the approach of the Rich Client Platform (RCP) of Eclipse. Dedicated applications can be build by using parts of g-Eclipse as a common library to handle resources and grid access. RCP application can use the g-Eclipse core model only, or can use some GUI components, too. With such an approach existing applications can be gridified. The g-Eclipse core model is grid middleware independent. By electing the a appropriate set of plug-ins, the same application can access resources from different grid middleware. The g-Eclipse model defines interfaces for accessing virtual organisations, information system, job submission services, job monitoring services and grid file systems. There is also support for managing authentication tokens, storing submitted jobs locally, automatic notifications about job statuses and many others. If the results of computation needs to be visualised then another plug-in can be added that will handle the output file format and will visualise it, possibly using g-Eclipse visualisation.

References

- [1] H. Kornmayer, M. Stümpert, K. H. Gjermundrød, P. Wolniewicz, *g-Eclipse - A Contextualised Framework for Grid Users, Grid Resource Providers and Grid Application Developers*, ICCS 2008, Lecture Notes in Computer Science 5103, Springer 2008, pp. 399-408
- [2] M. J. Frisch at all, *Gaussian 03, Revision C.02*, Gaussian, Inc., Wallingford CT, 2004