

# User Deployment of Grid Toolkits to Engineers

M.H. Eres, G.E. Pound, A.J. Keane, and S.J. Cox

School of Engineering Sciences, University of Southampton, Southampton SO17 1BJ  
{Hakki.Eres, G.E.Pound, Andy.Keane, S.J.Cox}@soton.ac.uk

## Abstract

Recently a full-scale deployment of the Geodise toolboxes has been performed in the Computational Engineering and Design Centre, University of Southampton. This deployment included the following steps: installation of required software components such as Globus, Condor, Oracle, and Websphere Application Server; installation of Geodise tools; configuring available software packages; training researchers via documents and workshops; and administering the whole infrastructure. In this paper we present our hands-on experience from deploying the toolboxes along with the challenges that may arise while Grid-enabling an existing computational cluster of an organisation. A special emphasis will be given to problems and proposed solutions of accessing Condor pools across a NAT firewall using the existing Grid infrastructure.

## 1. Introduction

The Geodise Project [1] is a research programme with the purpose of demonstrating a potential role for Grid-computing in the field of engineering design search and optimisation. In a series of demonstrators the Geodise project has married the disciplines of design optimisation and Computational Fluid Dynamics with state of the art computer science to tackle industrial design problems [2].

During the development of these demonstrators a number of Grid-enabled design tools have been developed. These tools include a number of generic toolboxes for the submission of computational jobs and data management, as well as Grid-enabled design optimisation packages [3][4]. Additionally we provide a high-level GUI that allows novice users to graphically configure an optimisation to run over the Grid with support from automated design advisors [5].

The Geodise toolboxes have been designed from a user-centered perspective, with the aim of reducing the barriers to entry for engineers wishing to access, and exploit, Grid resources. To this end Matlab [6] was chosen as the top-level scripting environment for all of the Geodise tools. Matlab is a mature cross-platform technical computing environment that is widely used in the field of engineering design. Matlab provides a familiar environment that has the power and flexibility to meet the requirements of engineers wishing to perform design optimisation across a range of disciplines [2][7][8].

Consequently the Geodise tools are of interest to colleagues in the Computational Engineering and Design Centre (CEDC) of the University of Southampton. The CEDC contains a number of researchers and post-graduate students working in the field of industrial design optimisation in partnership with companies such as BAE Systems, Rolls-Royce, and Fluent.

The scale of the optimisation problems tackled within the CEDC are computationally and data intensive, often requiring hundreds of automated CFD calculations to be run concurrently during a single optimisation. These characteristics make the problems studied within the CEDC well suited to Grid-computing. However the majority of staff within the CEDC typically have little or no experience with Grid-computing. The focus of the Geodise toolboxes upon usability, and the integration into the existing working practices of the engineer, makes them suitable to facilitate staff within the CEDC to exploit the Grid.

Therefore we undertook a deployment of the Geodise tools within the CEDC. This provided valuable experience about the requirements for, and problems arising from, the roll-out of Grid software. In this paper we summarise the steps undertaken to deploy the Geodise tool, and to support the new users of the software. We provide details of the problems posed by the existing Network Address Translation (NAT) firewall used by the CEDC. Finally we conclude with a summary of the challenges faced, and the lessons learnt, during this experience.

## 2. Geodise Tools

The Geodise project provides a number of tools to assist the engineer to perform Grid-enabled design search and optimisation. These tools include a number of services for data management and job submission, and crucially a set of Matlab toolboxes that provide the client functionality to the Grid services and resources.

The Computation Toolbox gives the Grid users essential Grid computing functionality such as certificate based authentication, job submission, and file transfer, along with numerous Grid related utility functions [3]. Some of this functionality is analogous to the technologies previously used to automate to execution in jobs internally within the CEDC (`rlogin`, `rcp`, etc.), however the Geodise tools provide seamless access to computational resources across the Grid.

The Database toolbox uses Grid technologies, Web services, and XML to enable users to archive files and data to a managed data archive. Archived files and data can be associated with custom metadata defined by the engineer. Datagroups which describe relationships between items in the archive can be created, and sophisticated queries can be used to locate and retrieve data stored in the archive [4]. Much of this data management functionality is unfamiliar to users within the CEDC and represents a departure from existing methods of organising and storing the results of optimisation experiments.

OptionsMatlab is a Matlab interface to the Options design exploration system [9], and allows users to perform engineering design search and optimisation techniques directly from the Matlab environment. The Options system is already widely used within the CEDC, however the OptionsMatlab interface allows users to easily perform optimisations across resources upon the Grid.

The Workflow Editor is a graphical problem solving environment which allows users to construct and configure workflows for the Grid [5]. This tool is suitable for novice users who are unfamiliar with the Matlab environment. Generic optimisation workflows can be constructed by an expert user, and subsequently configured by new users to optimise their problem.

Geodise tools extensively utilize the Globus Toolkit 2.4.3 [10], which is widely used in the UK e-Science Grid for certificate-based authentication, job submission, and file transfer processes. For example, access to the UK National Grid Service [11] is provided via Globus GT2 middleware.

Geodise also uses Condor [12] as its workload management system for compute jobs. Condor utilizes wasted CPU power from idle desktop workstations by matching resource requests with resource offers. Additionally, Condor's "flocking" technology allows multiple Condor compute installations to work together. However, firewalls and more restrictive network security policies make collaboration between different sites difficult, if not impossible. One solution to this problem is provided by the Southampton eScience Centre job submission service which provides a Web-service interface to Condor pools [13].

## 3. Geodise Deployment

Prior to deployment in the CEDC the Geodise software was tested internally by the Geodise team. Software testing was primarily by the software developers, and subsequently during the development of a series of application demonstrators in which the Geodise tools are used to tackle industrial scale engineering optimisation problems [2]. This internal testing and exposure to genuine optimisation problems ensured that the Geodise toolboxes are sufficiently robust for deployment outside of the development environment.

The deployment of Geodise tools at the CEDC provided us an opportunity to review the prerequisites, installation procedures and configuration necessary to establish a working installation of the Geodise services and client tools.

When deploying Grid software within an organisation this software must conform to the requirements of the local IT policy, in particular to the security requirements of that policy. Local IT policies may limit which local resources, if any, are permitted to be exposed to the wider Grid. If the IT policy allows local resources to be exposed to the Grid, this may be upon condition of a trusted authentication and encryption scheme.

Today many organisations will attempt to restrict the traffic that can enter and leave their

local network to help protect against malicious activity. Consequently peer-to-peer systems, such as Condor, may not be able to function across a firewall that is deployed to protect the local network.

A number of computational and data resources within the CEDC were exposed as Globus servers. The purpose of this was to provide a single interface to computational resources inside of, and external to, the CEDC network. Additionally a Condor pool was created to exploit the computational power of idle workstations within the CEDC. Due to the NAT firewall none of the internal computational resources are accessible outside of the CEDC.

Matlab 6.5, which is required as the environment from which the user will access the Grid, was already installed within the CEDC. Users within the CEDC access a central Unix installation of Matlab. It was therefore straightforward to alter the installation to make the Geodise client tools available to all users with the CEDC. To date no conflicts in the Matlab environment have been reported by users in the CEDC.

Finally it was important that the deployment process was thoroughly documented. This documentation ensures that the local system administrators have the appropriate knowledge to maintain the installed software.

#### **4. User Training and Support**

This exercise has also provided a fresh supply of users of the Geodise software. To encourage the uptake of the Geodise toolboxes we have provided training and support for students and researchers within the CEDC. In addition to the existing documentation for the Geodise toolboxes provided via the Matlab help function, several tutorials and FAQs were developed to meet the needs of users who are unfamiliar with either the Grid or the Geodise tools.

The Geodise team have also supplied a number of training sessions for new users within the CEDC. This training had the initial aims of educating new users about the key concepts of Grid computing, such as user certificates, and increasing their awareness of the Grid resources available and details of the client functions. Further training will help to develop the skills of users in the CEDC to employ the Geodise software, and to effectively

exploit the Grid resources that are available to them. The ultimate aim of the user training is to develop a user community within the CEDC that is Grid-aware, and is able to develop applications without input from the Geodise team.

User training took the form of a number of user workshops. These workshops are an opportunity for the Geodise team to demonstrate the functionality available from the Geodise software. More importantly the workshops were used to develop worked examples of Grid-enabled optimisations with the user's applications. These worked examples proved an effective medium for the exchange of information between users and the Geodise team.

There are a number of pay-offs for the Geodise software developers from this investment in user training and support. New users will invariably use software in a variety of new and unexpected way, providing valuable stress-testing of the software. User feedback has been important in identifying missing functionality, usability issues and software bugs. In this way feedback has been essential in the development of tools that are of value to engineers performing design search and optimisation. For example following a demonstration of the Geodise software a discussion of the user's requirements exposed the requirement for specific higher-level Matlab functions that would assist engineers to Grid-enable their optimisation problems.

#### **5. Accessing External Resources**

The network of the CEDC is behind a NAT firewall. The firewall rewrites IP addresses of network packets, and makes accessing CEDC resources from external networks impossible. This is an effective solution for network security, but it also brings quite restrictive limitations to Grid computing. For example, Globus job submissions with call-backs, joining external Condor pools, and using Condor's flocking technology is impossible for researchers from the CEDC.

During parallel design optimisation, job submissions are typically in batch mode, thus allowing jobs to be run concurrently. Therefore call-backs from the Globus server to the client machine are not required by the Geodise tools. Instead the Geodise toolboxes utilise simple

client-server communication which are unaffected by the NAT firewall.

Alternative mechanisms were provided to access Condor pools external to the firewall utilizing Globus GRAM servers and a job submission Web service [13] (Figure 1). Both techniques require that both servers are members of the external Condor pool, and are allowed to make job submissions to the pool. These techniques can also be used within the local area network to access a local Condor pool by using Geodise tools from the Matlab environment or by using the Workflow Editor.

Condor performs its resource matching process by using submit description files where the users define their requirements for operating system, memory, executable files, etc. The Condor master uses these requirements to match the user's job to suitable resources in the Condor pool. Alternative mechanisms to submit jobs to a Condor pool must support these rich job descriptions.

The `gd_condorsubmit` Matlab function submits the user's job to a Condor pool via a Globus server. This function requires as an argument a Matlab structure that describes the job that the user wishes to submit to the Condor pool. The function uses this structure to generate a Condor job description file. The function assumes that the field names in the input structure follow the same convention as Condor. For example, the following Condor submit description file

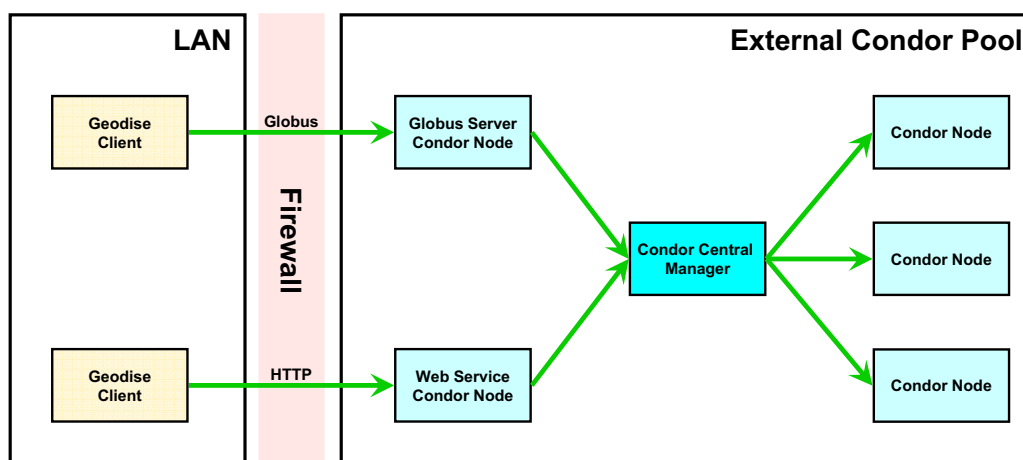
```
# Sample submit description file
executable = foo
log = foo.log
```

is described in Matlab as

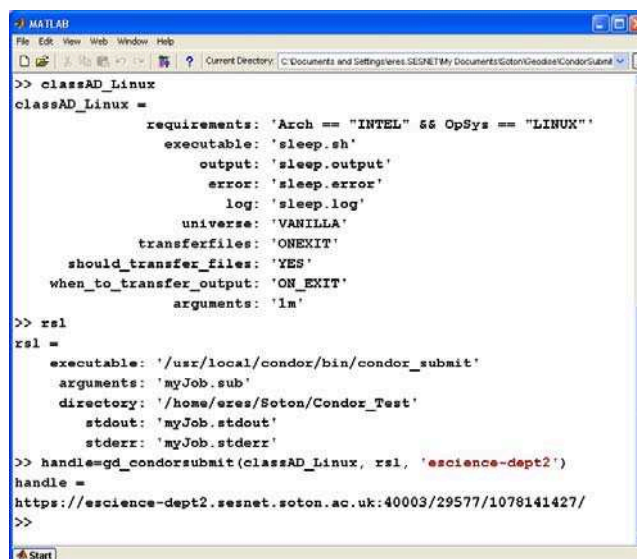
```
% Sample Matlab structure
classAD.executable = 'foo';
classAD.log = 'foo.log';
```

The second argument to the `gd_condorsubmit` function is a Matlab structure that defines the RSL [14] command that is responsible for the submission of the user's job to the Condor pool. This structure must contain details about the location of the `condor_submit` command on the Globus server, the name of the working directory at the Globus server, and the standard output and error files.

Finally the user must specify the name of the Globus server that will mediate the submission of the user's job to the Condor pool. The Globus server must be a member of the Condor pool and have permission to submit jobs to the queue. The `gd_condorsubmit` function also performs basic syntax checking on the Condor and RSL input structures. For example, users are warned if wrong options of Condor file transfer mechanisms are set on systems with (or without) a shared file system. After successful completion the command returns a unique job handle that can be used to query the status of the submitted GRAM job. The Matlab environment during a Condor job submission process is illustrated in Figure 2.



**Figure 1** Accessing an external Condor pool through a firewall by using a Globus server or a Web service. Local clients can not join or flock to the Condor pool because of the firewall, and Globus callbacks are also blocked.



```

MATLAB
File Edit View Web Window Help
Current Directory: C:\Documents and Settings\eres\My Documents\Soton\Geodise\Condor\Subnet

>> classAD_Linux
classAD_Linux =
    requirements: 'Arch == "INTEL" && OpSys == "LINUX"'
    executable: 'sleep.sh'
    output: 'sleep.output'
    error: 'sleep.error'
    log: 'sleep.log'
    universe: 'VANILLA'
    transferfiles: 'ONEXIT'
    should_transfer_files: 'YES'
    when_to_transfer_output: 'ON_EXIT'
    arguments: 'lm'

>> rsl
rsl =
    executable: '/usr/local/condor/bin/condor_submit'
    arguments: 'myJob.sub'
    directory: '/home/eres/Soton/Condor_Test'
    stdout: 'myJob.stdout'
    stderr: 'myJob.stderr'

>> handle=gd_condorsubmit(classAD_Linux, rsl, 'escience-dept2')
handle =
https://escience-dept2.sesnet.soton.ac.uk:40003/29577/1078141427/
>>

```

**Figure 2** The Matlab environment during a Condor job submission process. Here, the user defines a ClassAD and RSL as Matlab structures and makes a job submission to the Globus server.

By contrast the Southampton eScience Centre (SESC) Job Submission web service is a bespoke interface to Condor pools. As with job submission via a Globus server the web service uses a client-server model that is firewall friendly (Figure 2).

The web service interface allows the user to describe all of the requirements of the job that they wish to run. These requirements are converted into a Condor job description file. The user's executable and specified input files are automatically transferred to the web server using the DIME protocol. The web service performs the job submission to the condor pool and returns a job ID to the user that may be used to query the status of the job. As with the previous mechanism the server hosting the web service must be a member of the Condor pool.

The SESC Job Submission Web-service is implemented on the .NET platform, however a Java client to the service allows client functionality to be provided from the Matlab environment. User authentication with the web-service uses X509 certificates mediated by the WS-Security protocol.

These different mechanisms resolve some of the issues associated with accessing a Condor pool across a firewall. The first approach, in which job submission is mediated by a Globus server, is suitable for situations in which both technologies have been deployed together.

However the Web-service approach provides a bespoke interface that handles the

details of a job submission transparently from the user. The service based architecture may be of relevance to organisations that wish to provide access to a resource (i.e. a Condor pool) via a defined interface, without granting clients user privileges on a server.

## 6. Conclusions and Future Work

The process of a full-scale deployment of Geodise toolboxes has proven to be very useful in helping the project team to understand the detailed prerequisites for the integration of Grid-technologies into an established IT infrastructure. This work demonstrated that a pragmatic approach is needed in which the Grid software is deployed within the constraints created by the existing IT architecture and policy.

A considerable amount of installation and configuration time was spent on the underlying components, such as Globus, Condor, Oracle, and Websphere Application Server installations. In contrast, the time spent on the installation, configuration, and maintenance of the Geodise toolboxes was minimal. This was due in part to the development of client tools within the standard Matlab environment.

A considerable amount of effort was devoted to meeting the human factors requirements of system administrators and users during the deployment and adoption of the software. The effort devoted to training and support for the new users of the Geodise tools has also provided tangible benefits for the

project team in feedback about the tools provided and an improved understanding of the needs of the end-user.

Future work involves generating installation scripts for Geodise toolboxes to simplify the deployment of software. High Throughput Computing promises to meet many of the needs of engineering design search and optimisation. We shall therefore seek to improve access to external Condor pools, in part by exploring Condor/G's native Grid access functionality that may provide seamless access between Condor pools across the Grid. Additionally, we plan to support higher level Matlab functions tuned for particular engineering design search and optimisation processes by aggregating several simpler Grid functions. These higher level functions can use the underlying services and resources more transparently by hiding details from the end-user. These higher level blocks will also be made available in our workflow editor.

## References

- [1] The Geodise Project.  
<http://www.geodise.org/>
- [2] W. Song, A.J. Keane, and S.J. Cox, CFD-Based Shape Optimisation with Grid-Enabled Design Search Toolkits. Proceedings of UK e-Science All Hands Meeting 2003, pp. 619-626.
- [3] M.H. Eres, G.E. Pound, Z. Jiao, J.L. Wason, F. Xu, A.J. Keane, and S.J. Cox. Implementation and utilisation of a Grid-enabled problem solving environment in Matlab. Future Generation Computer Systems, 2004. (In print).
- [4] J.L. Wason, M. Molinari, Z. Jiao, and S.J. Cox. Delivering Data Management for Engineers on the Grid. Euro-Par 2003 Parallel Processing, Lecture Notes in Computer Science, 2003, pp. 413-416.
- [5] F. Xu, M.H. Eres, and S.J. Cox. Workflow Tool for Engineering Design Search and Optimisation Process on Grid Computing Environments, 2004. (In preparation).
- [6] Matlab. <http://www.mathworks.com/>
- [7] M. Molinari, K.S. Thomas and S.J. Cox. Electromagnetic design search and optimisation of photonic bandgap devices on distributed computational resources. Proc. of 5<sup>th</sup> International Conference on Computation in Electromagnetics 2004, pp. 103-104.
- [8] A.R. Price, G. Xue, A. Yool, D.J. Lunt, T.M. Lenton, J.L. Wason, G.E. Pound and S. J. Cox. Tuning GENIE Earth System Model Components using a Grid Enabled Data Management System. Proceedings of UK e-Science All Hands Meeting 2004, (in press)
- [9] A.J. Keane. The Options Design Exploration System: Reference Manual and User Guide - Version B3.1.  
<http://www.soton.ac.uk/~ajk/options.ps>
- [10] The Globus Project.  
<http://www.globus.org/>
- [11] The UK National Grid Service.  
<http://www.ngs.ac.uk/>
- [12] The Condor Project.  
<http://www.cs.wisc.edu/condor/>
- [13] G. Xue, M. Fairman, G.E. Pound, S.J. Cox. Implementation of a Grid Computation Toolkit for Design Optimisation with Matlab and Condor. Euro-Par 2003 Parallel Processing, Lecture Notes in Computer Science, 2003, pp. 357-365.
- [14] Globus Resource Specification Language (RSL) v1.0.  
[http://www.globus.org/gram/rsl\\_spec1.html](http://www.globus.org/gram/rsl_spec1.html)