Tools and Support for Deploying Applications on the Grid

Fenglian Xu[*], M. Hakki Eres[*], David J. Baker[‡], and Simon J. Cox[*]

[*] *School of Engineering Sciences*
University of Southampton
Southampton, Hampshire, SO17 1BJ, UK
[‡] *Information Systems Services*
University of Southampton
Southampton, Hampshire, SO17 1BJ, UK
{F.Xu, Hakki.Eres, D.J.Baker, S.J.Cox}@soton.ac.uk

Abstract

Deploying applications on the Grid requires both a stable production quality infrastructure, and correct configuration of the user environment and applications. In this paper we discuss two support tools to assist in increasing the chance of successful execution of an application. The first is a set of scripts which we have developed and deployed on the UK e-Science Grid to check at a system level that resources are live and ready for use. These checks might be thought of as a 'heartbeat' for the Grid. The second is a support tool targeted at end users which allows them to verify that the specific resources they require are available and that their applications and user security settings are configured correctly. This verification gives a 'dial tone' for the application of the user. We demonstrate their deployment and use.

1. Introduction

Tools have a crucial influence on the ability of programmers to develop and deploy applications. Special consideration and support is required if the intended deployment environment is a large scale Grid [1]. The Geodise project [2] brings computer scientists and engineers together to deliver a suit of Grid-enabled tools for engineering design search and optimisation (EDSO) [3]. EDSO is a long and repetitive computation process, and in Geodise, compute and data Grid resources are called from a Matlab [4] script. We use the Globus Toolkit [5] and Java Commodity Grid Kit [6] as the middleware to handle the execution of codes on any Grid resource. We also use Globus toolkit functionality to transfer files between clients and servers. In this paper we focus on assisting the end user in deploying applications in this environment.

The Grid Engineering Task Force (GETF) [7], with members from each regional e-Science Centre (See Figure 1), guides the construction and testing of the production-quality e-Science Grid in the UK [8]. To address the integration issues of the infrastructure, the Grid Integration Working Group (GIWG) [9] was setup with the following objectives:

- successful deployment of Globus servers at regional e-Science Centres,
- contribution of Grid-enabled resources of regional e-Science centres to the UK wide Grid infrastructure,
- quality assurance of resources.

These objectives enable e-Science project members to access hardware and software resources easily and reliably, and allow them to carry out their research successfully.

The Grid Integration Test Script (GITS) [10] is used during various phases of the UK e-Science Grid to test Grid services between servers. The GITS enables integration tests to be performed between nodes on the Grid and allows monitoring of active members of the UK e-Science Grid.

Even though the GITS scripts are very useful for Grid node administrators, they do not provide real-time information about the status of the Grid for the end user, who wishes to use existing Grid resources for a particular application. The Grid tool allows a client to test all of the basic functionality of the Globus toolkit on the Grid resources, which will be used for their application. This tool enables a more detailed test of resources on the Grid immediately prior to launching their application and details such as the user's configuration and security settings. In this paper we

detail these two tools, illustrate their large scale usage on the heterogeneous UK e-Science Grid, and demonstrate their efficacy.
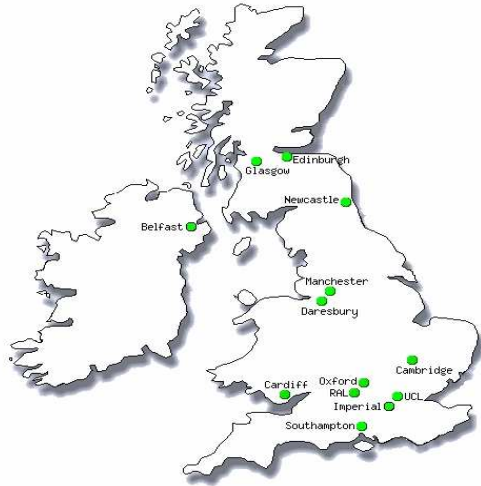

Figure 1. National and regional e-Science centres in the UK.

The remainder of this paper is organised as follows. In Section 2, we describe and outline the GITS. In Section 3, we describe the system architecture and implementation of the Grid tool. In Section 4, we illustrate the functions for the Grid tool. In Section 5, we demonstrate the functions of the Grid tool in Geodise Workflow Editor environment. Finally, conclusions will be drawn in Section 6.

2. Grid Integration Test Script

As the number of Grid resources grows, performing quality assurance tests between every resource becomes more time consuming and burdensome, but increasingly important. The GITS aims to simplify these tasks and allow system administrators to report the status of the Grid easily and reliably. GITS is written in Perl [11] to achieve portability and usability. GITS has a number of integration and operation monitoring tests which are appropriate to the Grid services supported by the UK e-Science Grid. These tests include, for example, point-to-point tests to check authentication, job submission, and file transfer, and tests to check the status of the Monitoring and Discovery Service (MDS) [12][13] functionality of Grid nodes. Every test result can either be saved in HTML or XML format which provide excellent presentation and archive options for system administrators. An example of HTML results is given in Figure 2.

Monitoring and analysing the test results can be a difficult and time consuming process for the node administrators. The GITS Web Service [14] allows Grid node administrators to perform their tests, save the results in XML format, and archive their results to a central database. This centralised archive facility makes published results available to the whole UK e-Science community, and it proved to be very useful for administrators to maintain and monitor the whole test results. However, the end users are only interested in the connections from their organization to the nodes in the UK e-Science centre.

Figure 2. An example of the HTML output from GITS.

3. Design and Implementation of the Grid Tool

The Grid tool is a stand-alone application which can be easily integrated within other graphical user interface (GUI) applications. It is platform independent as it is implemented in Java [15] and it can be run on any compatible Java virtual machine. In its current configuration the system is designed to be able to run a Matlab script remotely on a Grid-enabled distributed system. Its three layered architecture is shown in Figure 3.
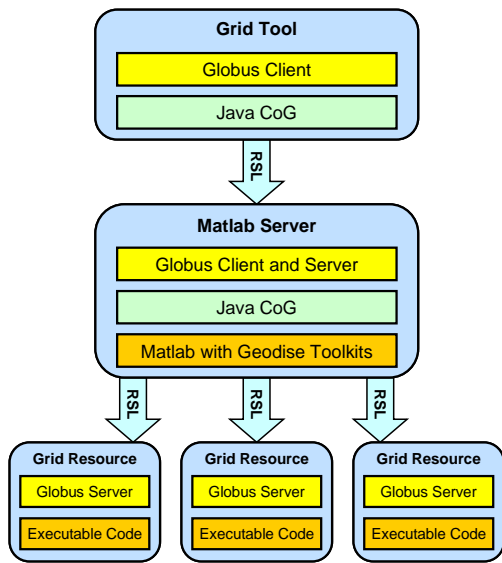

Figure 3. Architecture of the Grid tool.

The top layer is a GUI which provides tools for authentication, Grid utility checking and Grid resource management. The Java Swing API is used to implement the graphical components of the GUI. The middle and lower layers are all Grid resources with installed software to perform a certain task. In the middle layer Matlab is required to execute the Matlab script file and user defined Matlab components for intensive computing purposes. Especially the Geodise toolkits are Matlab scripts or functions to enable Grid computing and data transferring on the Grids. The executable components in the third layer can be any third-party or user supplied code to perform a computational task.

The Grid tool also requires the Java CoG 1.1 API which enables the user to use GSI, GRAM, and GridFTP protocols [16][17] from the Globus toolkit. Both GridFTP and GRAM require that the user has a valid Grid certificate. The Grid certificate can be propagated into different layers which mean that the user only needs to sign-in once to access all Grid resources. In Figure 3, the top level is a front GUI and it provides an entry for the user to interact with the system. If the user creates a valid certificate from the GUI, it is then automatically propagated to the second level so that the Matlab server can validate the proxy certificate and allows the user to perform job submissions and file transfers to the low level Grid resources. The implementation of communication between the client and Globus server is via the Resource Specification Language (RSL) [18].

4. Functions of the Grid Tool

The Grid tool consists of a main frame with a menu bar and a tree table (see Figure 4). The menu bar contains a file menu and a tool menu. The file menu enables the user to open a compute resource configuration file, save the modified compute resource configuration file, and exit the Grid tool. The contents shown in the tree table is loaded from an initial XML configuration file which can be modified by either manually editing the resource configuration file, or directly within the Grid tool.

Figure 5 shows an example of XML configuration file. The content of the tree table contains a root node: compute resource which includes a Matlab node and a Globus node. The Matlab node consists of a list of Matlab servers so that the user can select a validated one for use and it requires that Matlab is installed on the Grid resource. The Globus nodes are composed of a list of servers with Globus Toolkit server software. These Globus nodes enable remote job execution and file transfers to be performed on a distributed system. The tool menu provides a full set of functions to verify authentication, job submission, and file transfer operations from the client to different Grid resources.

4.1. Authentication Test

GSI is the Globus Grid security infrastructure which is used for both GRAM and GridFTP. The clients must have a valid Grid certificate to run a remote job on a Grid compute resource and to transfer a file to (or from) a Grid resource. Here, we assume that the user has a valid Grid proxy and that the user's Grid proxy is mapped to a valid user account by the system administrator of the Grid resource. The Grid tool provides the user with the following menu items:

− Generate proxy: Displays a Grid proxy initialisation window for the user to enter their Grid certificate password.
− Delete proxy: Deletes a valid Grid proxy after obtaining a confirmation message from the user.
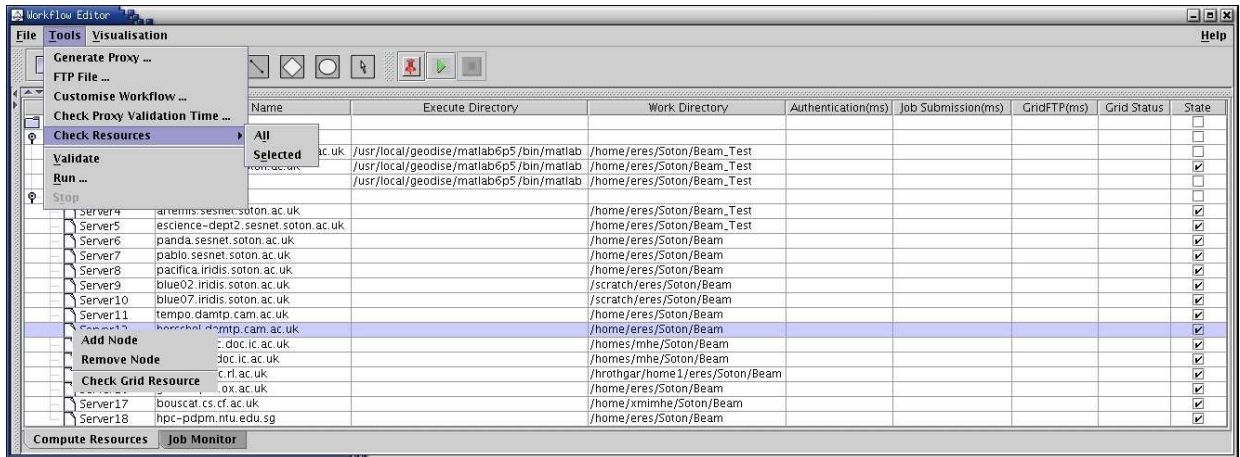
Figure 4. The main window of the Grid tool showing tools menu and individual Grid node popup menu.

− Check proxy validation time: If a valid proxy exists, displays a message dialog with valid Grid information, such as subject, issuer, and time left etc. If there is no valid proxy available, it warns the user with a message dialog.

These Grid resources are not restricted to the UK e-Science Grid resources. They can be any Grid resource as long as the user has registered with that particular system and has been authorised to use the compute resource.

```
<?xml version="1.0" encoding="UTF-8"?>
<computeResources>
    <matlabServer>
        <Matlab1 executePath="/usr/local/geodise/matlab6p5/bin/matlab" flag="on"
            hostName="escience-dept2.sesnet.soton.ac.uk"
            workPath="/home/eres/Soton/Beam_Test"/>
        <Matlab2 executePath="/usr/local/geodise/matlab6p5/bin/matlab" flag="off"
            hostName="artemis.sesnet.soton.ac.uk" workPath="/home/eres/Soton/Beam_Test"/>
        <Matlab3 executePath="/usr/local/geodise/matlab6p5/bin/matlab" flag="off"
            hostName="localhost" workPath="/home/eres/Soton/Beam_Test"/>
    </matlabServer>
    <globusServers>
        <Grid1 flag="on" hostName="artemis.sesnet.soton.ac.uk" workPath="/home/eres/Soton/Beam"/>
        <Grid2 flag="on" hostName="escience-dept2.sesnet.soton.ac.uk"
            workPath="/home/eres/Soton/Beam"/>
        <Grid3 flag="off" hostName="panda.sesnet.soton.ac.uk" workPath="/home/eres/Soton/Beam"/>
        <Grid4 flag="off" hostName="pablo.sesnet.soton.ac.uk" workPath="/home/eres/Soton/Beam"/>
        <Grid5 flag="off" hostName="pacifica.iridis.soton.ac.uk" workPath="/home/eres/Soton/Beam"/>
        <Grid6 flag="off" hostName="blue02.iridis.soton.ac.uk" workPath="/scratch/eres/Soton/Beam"/>
        <Grid7 flag="off" hostName="blue07.iridis.soton.ac.uk" workPath="/scratch/eres/Soton/Beam"/>
        <Grid8 flag="off" hostName="tempo.escience.cam.ac.uk" workPath="/home/eres/Soton/Beam"/>
        <Grid9 flag="off" hostName="herschel.damtp.cam.ac.uk" workPath="/home/eres/Soton/Beam"/>
        <Grid10 flag="off" hostName="pioneer.lesc.doc.ic.ac.uk" workPath="/homes/mhe/Soton/Beam"/>
        <Grid11 flag="off" hostName="viking.lesc.doc.ic.ac.uk" workPath="/homes/mhe/Soton/Beam"/>
        <Grid12 flag="off" hostName="grid-data.rl.ac.uk" workPath="/home/ngs0132/Soton/Beam"/>
        <Grid13 flag="off" hostName="hrothgar.esc.rl.ac.uk" workPath="/hrothgar/home1/eres/Soton/Beam"/>
        <Grid14 flag="off" hostName="grid-compute.oesc.ox.ac.uk" workPath="/home/ngs0132/Soton/Beam"/>
        <Grid15 flag="off" hostName="gatekeeper.ox.ac.uk" workPath="/home/eres/Soton/Beam"/>
        <Grid16 flag="off" hostName="bouscat.cs.cf.ac.uk" workPath="/home/xmimhe/Soton/Beam"/>
        <Grid17 flag="off" hostName="grid-data.man.ac.uk" workPath="/home/ngs0132/Soton/Beam"/>
        <Grid18 flag="off" hostName="grid-compute.leeds.ac.uk" workPath="/home/data01_a/ngs0132"/>
        <Grid19 flag="off" hostName="hpc-pdpm.ntu.edu.sg" workPath="/home/eres/Soton/Beam"/>
    </globusServers>
</computeResources>
```

Figure 5. An example XML file describing the Grid resources for the Grid tool.

4.2. Grid Resource Verification

A Grid resource is specified by a host name, executable directory, and work directory. The resource checking process measures the time required for authenticating the user on the remote Grid resource, running a test job, and transferring a file to (or from) it. Additionally, it verifies that the executable and work directory exist on the Grid resource. The end user can verify all available Grid resources at once, or verify selected Grid resources in the tree table, or verify a single Grid resource (See Figure 4).

Figure 6 shows the measurement results of selected Grid resources. Details of each resource metric are as follows:

− Executable directory: Check if the executable code exists under the specified directory on the Grid resource.

− Work directory: Check if the work directory exists for storing output files on the Grid resource.

− Authentication: Time elapsed in milliseconds to authenticate the users Grid proxy to the Grid resource successfully.

− File transfer: Time elapsed in milliseconds to transfer a short test file from the client to the Grid resource successfully.

− Job submission: Time elapsed in milliseconds to run a job on the Grid resource successfully.

− Grid status: OK if the above verifications are successful, failed otherwise.

Figure 6. The main window of the Grid tool showing Grid test results.

Property	Host Name	Execute Directory	Work Directory	Authentication(ms)	Job Submission(ms)	GridFTP(ms)	Grid Status	State
computerresources								
matlabServer								
Server1	escience-dept2.sesnet.soton.ac.uk	/usr/local/geodise/matlab6p5/bin/matlab	/home/eres/Soton/Beam_Test					
Server2	artemis.sesnet.soton.ac.uk	/usr/local/geodise/matlab6p5/bin/matlab	/home/eres/Soton/Beam_Test	393	1083	1722	ok	✔
Server3	localhost	/usr/local/geodise/matlab6p5/bin/matlab	/home/eres/Soton/Beam_Test					
globusServers								
Server4	artemis.sesnet.soton.ac.uk		/home/eres/Soton/Beam_Test	236	674	1939	ok	✔
Server5	escience-dept2.sesnet.soton.ac.uk		/home/eres/Soton/Beam_Test	167	695	1202	ok	✔
Server6	panda.sesnet.soton.ac.uk		/home/eres/Soton/Beam	428	914	2184	ok	✔
Server7	pablo.sesnet.soton.ac.uk		/home/eres/Soton/Beam				failed due ...	✔
Server8	pacifica.iridis.soton.ac.uk		/home/eres/Soton/Beam	216	579	30898	ok	✔
Server9	blue02.iridis.soton.ac.uk		/scratch/eres/Soton/Beam	393	1084	1571	ok	✔
Server10	blue07.iridis.soton.ac.uk		/scratch/eres/Soton/Beam				failed due ...	✔
Server11	tempo.damtp.cam.ac.uk		/home/eres/Soton/Beam				failed due ...	✔
Server12	herschel.damtp.cam.ac.uk		/home/eres/Soton/Beam	273	1191	1356	ok	✔
Server13	pioneer.lesc.doc.ic.ac.uk		/homes/mhe/Soton/Beam	711	1642	990	ok	✔
Server14	viking.lesc.doc.ic.ac.uk		/homes/mhe/Soton/Beam	159	954	975	ok	✔
Server15	hrothgar.esc.rl.ac.uk		/hrothgar/home1/eres/Soton/Beam	182	1297	1104	ok	✔
Server16	gatekeeper.ox.ac.uk		/home/eres/Soton/Beam	197	1994	31017	ok	✔
Server17	bouscat.cs.cf.ac.uk		/home/xmimhe/Soton/Beam				failed due ...	✔
Server18	hpc-pdpm.ntu.edu.sg		/home/eres/Soton/Beam	2220	3083	14592	ok	✔

4.3. Grid Resource Management

The Grid resource management data consists of information about all the Grid resources which can be accessed by the user. All the Grid resource information is stored in an XML configuration file. The user can either edit the XML configuration file, or modify it within the Grid tool. The user can add/remove resources from the tree table (see Figure 6), or modify the contents of the highlighted row in the tree table, such as the host name and the working directory. Finally, the resources to be tested prior to deployment can be selected.

5. Application Example

In this section, we will demonstrate how the Grid tool is used in Geodise Workflow Editor environment (see Figure 7). The Workflow Editor is a front end GUI tool mainly for workflow construction and runtime job management.

The workflow construction workspace is a working area for the users to construct a workflow and to instantiate a workflow. A workflow is defined by a list of nodes, these are task nodes, condition nodes, loop nodes, joint nodes and connection nodes. The task node is one of the components from the tree view. The connection nodes are used to link the nodes. The construct workflow is a concept workflow and the instantiated workflow is a concrete workflow. The process of converting a construct workflow to a concrete workflow is performed in the workspace. The construction can be done by:

− drag and dropping components from the component tree to form task nodes. The connection is done by selecting a connection mode and making a connection node which links two non-connection nodes,
− browsing an existing workflow from a local file system,
− searching an existing workflow from a database.

After a workflow is constructed and configured, it is deployed to computer resources for Grid computing. To increase the chance of successful execution, it is essential to know which compute resources are available prior to deployment. The user can either select partially or all the compute resources to check the availability and then make a decision to choose some nodes with an acceptable performance on remote job submission and file transfer processes. The workflow will then be ready to be deployed on these selected Grid resources.

6. Conclusions and Future Work

The Grid integration test script has proved to be a valuable tool for Grid resource administrators for building and maintaining the UK e-Science Grid. It also provides useful published information about the status of the Grid for users and administrators alike. In the future, we plan to make the existing GITS tool more 'user friendly' by adding a means of also delivering diagnostic information to the user, beyond the raw test results.

The Grid tool is designed for an end user to verify that the resources they intend to use are live and configured correctly. It is thus possible to validate the status of Grid resources before actually committing a computationally expensive simulation. In the future we plan to add

additional testing capabilities to ensure that the tests performed mirror the actual job as closely as possible.

The tools we have described and our approach are aimed at encouraging and assisting users to deploy applications on Grids, firstly by helping system administrators to monitor their Grid, and secondly by allowing users to get some confidence that their application is properly configured and will run. Exciting opportunities are offered by the Grid for science and engineering. This requires not only new functionality, but also reliability for users. Tools and environments play a critical role in delivering reliability, without which the potential of this new functionality will not be realized.
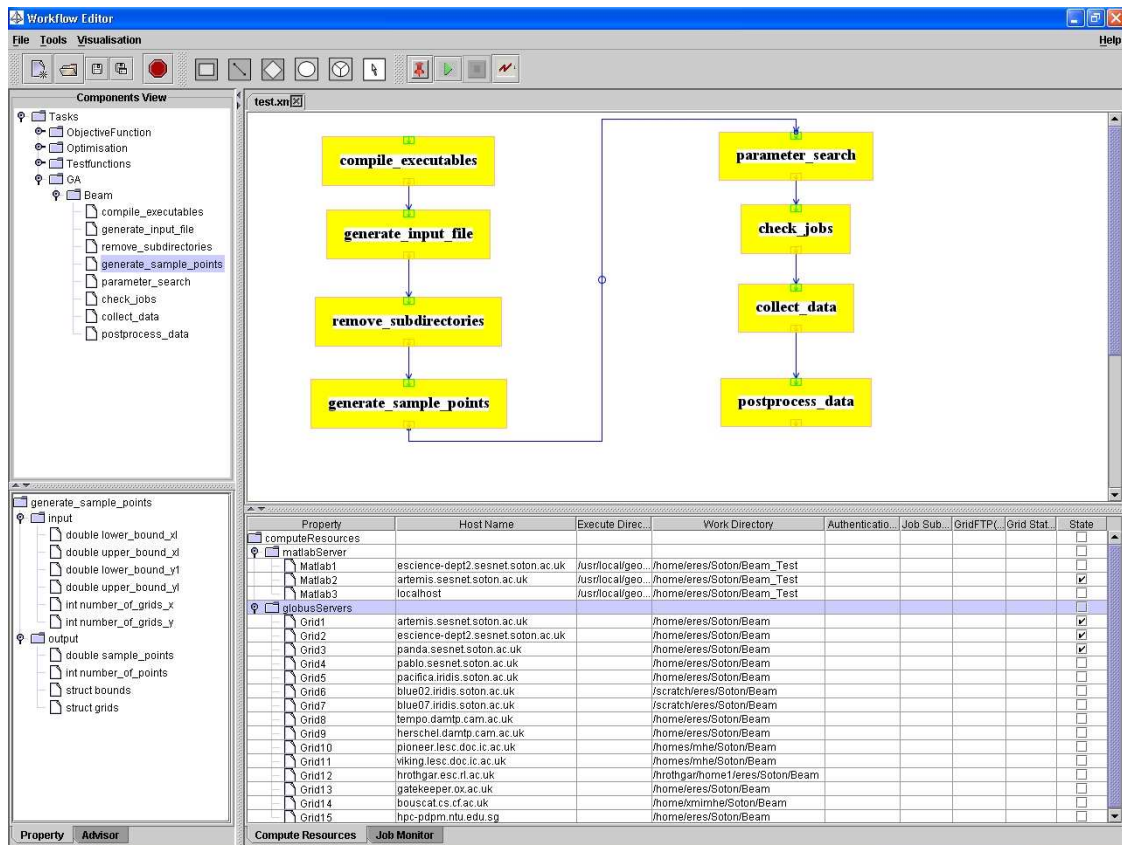

Figure 7. The main window of the Geodise workflow editor with integrated Grid testing functionality.

Acknowledgements

References

[1] B. Tierney, R. Aydt, D. Gunter, W. Smith, V. Taylor, R. Wolski, and M. Swany, "A Grid monitoring architecture", *The Global Grid Forum*, 2002.

[2] The Geodise Project, http://www.geodise.org/.

[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* (in print), 2004.

[4] Matlab 6.5, http://www.mathworks.com/.

[5] The Globus Project, http://www.globus.org/.

[6] G. von Laszewski, I. Foster, J. Gawor, and P. Lane, "A Java commodity Grid toolkit", *Concurrency and Computation: Practice and Experience*, 13, 2001, pp. 645–662.

[7] The Grid Engineering Task Force, http://www.grid-support.ac.uk/etf.

[8] R. Allan, et al., "Building the e-Science Grid in the UK: Middleware, applications and tools deployed at Level 2", *Proceedings of UK e-Science All Hands Meeting*, 2003, pp. 488–493.

[9] The Grid Integration Working Group, http://www.grid-support.ac.uk/etf/wg/integration-tests.html.

[10] The UK Grid Integration Test Script, http://www.soton.ac.uk/~djb1/gits.html.

[11] Perl, http://www.perl.org/.

[12] K. Czajkowski, S. Fitzgerald, I. Foster, and C. Kesselman, "Grid information services for distributed resource sharing", *Proceedings of the Tenth IEEE International Symposium on High-Performance Distributed Computing*, 2001.

[13] The Globus Monitoring and Discovery Service, http://www.globus.org/mds/.

[14] GITS Web Service, http://vermont.mvc.mcc.ac.uk/gqec/.

[15] Sun Microsystems Inc.: Java 2, http://java.sun.com/.

[16] R. Butler, D. Engert, I. Foster, C. Kesselman, S. Tuecke, S., J. Volmer, and V. Welch, "A national-scale authentication infrastructure", *IEEE Computer*, 33, 2000, pp. 60–66.

[17] B. Allcock, J. Bester, J. Bresnahan, A. Chervenak, I. Foster, C. Kesselman, S. Meder, V. Nefedova, D. Quesnel, and S. Tuecke, "Data management and transfer in high-performance computational Grid environments", *Parallel Computing Journal*, 28, 2002, pp. 749–771.

[18] The Globus Resource Specification Language (RSL) v1.0, http://www-fp.globus.org/gram/rsl_spec1.html.