Integration and operational monitoring tools for the emerging UK e-Science Grid infrastructure

David Baker[1], Mark Baker[2], Hong Ong[2] and Helen Xiang[2]

[1]University of Southampton Information Systems Services
[2]University of Portsmouth Distributed Systems Group

Abstract

We have, during the course of the past year, been developing a software toolkit to allow the sites participating in the Portsmouth led OGSA Testbed to carry out operational monitoring of resources and to facilitate the integration of the Grid. In this document we present and discuss the design and implementation the tools we are using to test and monitoring Grid middleware installed over the testbed. In particular we comment on our initial experiences of using the tools for integration testing and operational monitoring. Finally we highlight current problems and suggest potential improvements to the tools, and additionally we make suggestions for ways in which the project could be developed and diversified in the future.

1. Background

In October 2003, two one-year projects were funded in the UK by the EPSRC via the e-Science programme. The initial objective of these projects was to install and test the Grid middleware based on the Open Grid Services Architecture (OGSA), and then deploy a number of Grid-based applications. One of these projects, the OGSA Testbed, is led by the Distributed Systems Group (DSG) [1], and involves participants from the Daresbury Laboratory, Manchester, Reading, and Westminster universities [2]. Originally, the Testbed partners were focussed on the Open Grid Services Infrastructure (OGSI), and in particular the implementation of the Globus Toolkit version 3 (GT3), for its platforms and application work. Since the announcement of the Web Services Resource Framework (WSRF) in January 2004, the remit of the Testbed has been extended to encompass the development of applications based on both Web and Grid Services standards.

The integration of the software components and hardware infrastructure is a vital aspect of building any e-Science grid infrastructure. Considerable experience has already been developed in this area by the Southampton led Grid Integration Working Group, and their project known as Grid Integration Test Script (GITS) [3], which defines a set of point-to-point "integration tests" for the Globus Toolkit version 2 (GT2). Southampton University is now collaborating with members of the OGSA Testbed with the following objectives:

- Extend the GITS to encompass the emerging OGSA-based platforms,
- Define and discuss the suite of tests and their capabilities that will best exercise the deployed services,
- Use the OGSA Testbed as a platform to develop and deploy the updated version of GITS (GT3GITS),
- Study the best means for further extending the GT3GITS toolkit to incorporate user deployed application-based services.

With these objectives in mind, we have been developing a toolkit to enable users to do integration testing and to perform operational monitoring of the testbed resources.

The remainder of this paper is organised as follows. Section 2 describes the key features, and design of the GT3GITS. In Section 3 we outline the tests provided by the GT3GITS. In Section 4 we outline our efforts to integrate the GT3GITS with OGSA-DAI, which is the infrastructure used for storing and retrieving daily published operational monitoring results that reside in distributed databases. In Section 5 we describe our initial experiences of using the GT3GITS in the Testbed to carry out integration testing, and operational monitoring. Finally in Section 6 we present and discuss potential future challenges, and opportunities in this area of work.

2. Development of the gt3gits test framework

The GITS toolkit is being designed and implemented on a GT3 Grid. We assume that the experiences gained with this grid infrastructure can be transferred and employed to provide a generalised framework for performing integration testing in future Web Services based Grids. In other words, although the detailed coding, and the specific tests employed in the toolkit will, of course, need to be changed in the future, we hope that our ideas, and design decisions will be portable to future grid infrastructures.

Some of the major features (design decisions) employed in the toolkit are discussed here.

- **Perl was the language of choice:** The toolkit was written in Perl since it is widely known, and portable to a range of systems.
- **Sanity checking:** To ensure that the toolkit does not waste time trying to check services that do not exist or are inaccessible, before a service is tested it is important to perform sanity checks on the remote host. First of all the toolkit performs a test to verify that the GT3 Container Registry on a remote host does exist, and is accessible. This test is done just once for each specified host. If this test is successful then the toolkit looks inside the Container for the specified service, and only if this test is successful is the service tested.
- **Flexibility:** The user can specify which services are tested on a per host basis.
- **Diagnostics:** The toolkit places any errors in a log file, and using a database of known error messages attempts to give the user a hint as to why a service test failed. The latter activity is in an early stage of development.
- **HTML output:** The user can optionally output results in HTML format for clear and easy access to test results.

3. Tests performed by gt3gits

The toolkit is available for download. The scripts may be copied and modified freely by any site under the terms of the GNU General Public License. In this section we briefly describe the tests done by the GT3GITS; readers interested to know more should consult the documentation [4]. Generally speaking we have designed tests that are appropriate to the services deployed and used in the Testbed, and they may be categorised as follows:

- **Registry Services:** Tests to find, and search remote GT3 Container Registries.
- **GRAM Services:** Tests that check that the MFMJFS and services relating to batch schedulers are working correctly.
- **RFT Services:** Tests that do file transfer using GSIFTP, and the RFT.
- **OGSA-DAI Services:** Tests that exercise the OGSA-DAI registry and data services,
- **Index Services:** Tests that examine the GT3 informational infrastructure.
- **Other/User defined Services:** The toolkit provides the basic infrastructure to deal with user-defined services. An example based on the counter service is given in [4].

4. GitsDaiMiner (GDM)

The results from running the GITS are stored locally at each Grid site. Potentially the output can be linked to a Web page for external viewing. It was felt that a more flexible means of storing, retrieving, and displaying distributed GITS data would be advantageous.

One objective of the OGSA Testbed project is the exploration and testing of grid-related software. The OGSA-DAI project [5] is concerned with constructing middleware to assist with access and integration of data from separate data sources via the Grid.

It was felt that OGSA-DAI would be an appropriate vehicle to provide a unified GITS service across the Testbed. Hence the conception of GitsDaiMiner (GDM), which would not only provide a useful application of OGSA-DAI, but would also offer us a potentially flexible way of storing and presenting GITS-related data in a secure and transparent manner.

The basic idea with GDM is that each Testbed site executes a script that pushes GITS results into a local database (relational or XML). The GDM Grid service is deployed at each site and linked to the local GITS database. A GDM Web portal is then used to query and display GITS output. The architecture of GDM is shown in Figure 1.

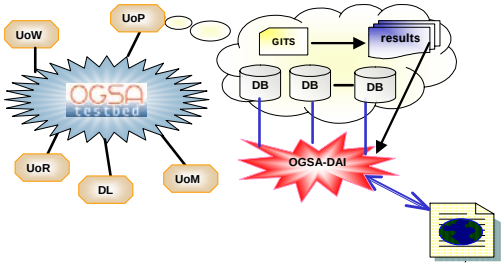

Figure 1: GitsDaiMiner (GDM) Architecture

5. Some early experiences of using GT3GITS in the testbed

This project is in a very early stage of development, and therefore our experiences are so far quite limited. Nevertheless we can offer useful comments in following areas:

- **Firewalls and port ranges:** GT3 is not particularly 'firewall friendly', and so a common source of problems is related to firewalls, and port range usage. Fortunately these issues are relatively easy to spot, and are easily resolved by getting administrators to modify their firewall rules and/or setting their port range correctly.
- **User interface:** The method of providing user defined tests is a little clunky (see the counter service example in [4]). In due course we will need to devise a schema for providing user test data using xml, for example.
- **Diagnostics:** Unfortunately GT3 can be ambiguous and very verbose when things go wrong.
- **Data overload:** For easy access to test data we decided to output the results in html format. An example of the html output from the toolkit is shown in Figure 2. Unfortunately as the number of sites, hosts, and services deployed in a grid grow the amount of data to analyze will soon become unmanageable. In this respect the ability to store and retrieve test data from a database is invaluable.

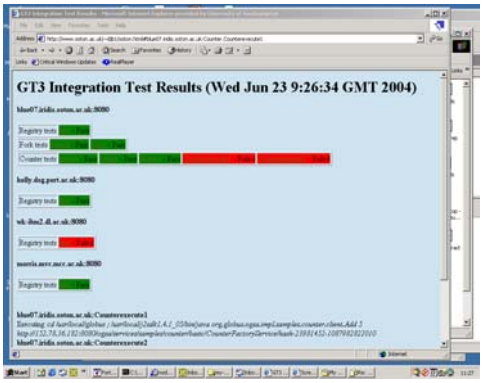

Figure 2: HTML output from the toolkit

6. Future challenges and opportunities in this area of work

In this section we present and discuss some strategic areas, which we feel need to be addressed. There are at least four potential development areas:

- **The toolkit infrastructure:** This project is very much in its infancy. In the coming months we will concentrate on making both the GT3GITS toolkit, and the GitsDaiMiner robust and reliable software.
- **Improving the user interface:** As mentioned in Section 5, the method of providing data for user defined tests is a rather static and less then ideal. This is fine for proof of concept and simple examples (for example the counter service). We do, however, need to design a more elegant interface, probably with Java Server Pages and Java Script. In addition, it is clear that the output data from the toolkit should be marked up in XML for a number of obvious reasons. This implies designing or adopting an XML naming schema for the data.
- **Providing better diagnostics:** If the toolkit is to be gainfully used by both users, and administrators then we must work at providing better diagnostics. Our limited experiences of using the toolkit in a GT3 environment have shown that this is a difficult, and demanding area to address
- **Supporting future Grids:** This is certainly our biggest challenge. We hope, however, that the experience we have gained by doing integration testing and operational monitoring in the current GT3 environment will stand us in good stead to provide a generalised framework for the future.

Acknowledgements

We would particularly like to thank Xiao Dong Wang (CCLRC Daresbury Laboratory) for his contributions to the GT3GITS toolkit. In addition we would like to thank all our colleagues in the OGSA Testbed for their ongoing support, contributions and suggestions.

References

[1] DSG, http://dsg.port.ac.uk/
[2] OGSA Testbed, http://dsg.port.ac.uk/projects/ogsa-testbed/
[3] GITS, http://www.soton.ac.uk/~djb1/gits.html
[4] GT3GITS, http://www.soton.ac.uk/~djb1/gt3gits.html
[5] OGSA-DAI, http://www.ogsadai.org.uk/