

1. Introduction

The ETF GT4 evaluation considered the state of the GT4 tool kit as part of the UK e-Science Engineering Task Force activities reviewing grid and web service middleware for production grid deployment within the UK e-Science community. The emphasis in the evaluation was to consider the strengths and weaknesses of the GT4 package and to identify issues that will be important if production deployment is recommended.

The evaluation reviewed the state of GT4 by installing the software, reviewing available documentation, creating sample GT4 grid services and testing inter-operation between GT2 and pre-WS GT4 components, and thus evaluating service capabilities, the documentation and the capabilities of the software.

1.1 Software Status.

A first GT4 release (GT4.0) was shipped on April 29 2005.

2. General Information

2.1 The Provider

GT4 is developed, maintained and supported by the Globus Alliance [1]. The core Globus Alliance team includes members at the Argonne National Laboratory (Chicago), University of Chicago, Edinburgh Parallel Computer Centre, the National Centre for Supercomputing Applications (Illinois), the Northern Illinois University's High Performance Computing Laboratory, the Royal Institute of Technology (Sweden), Univa Corporation (Chicago) and the University of Southern California's Information Sciences Institute (Marina del Rey California).

The Globus toolkit is an open source project and is thus tested and developed by a wider community than the Globus Alliance core team. The Globus Alliance maintains a collection of active discussion groups on middleware components, their installation and use, and the future development of middleware components. A bug tracking and search facility is used to register problems with GT4 components at [4].

2.2 Licensing

The Globus Alliance is *committed to a liberal, open source license* [1]. The Globus Toolkit is distributed *for use by anyone and for any purpose, without restriction* [1]. The full Globus Toolkit License used with GT4.0 is derived from the standard Apache license set out in [2] and reproduced in Appendix I. In addition, we are informed that Globus Toolkit will, in the near future, switch to the fully standard Apache license as set out in [2].

2.3 Supported Platforms

The Globus Alliance reports that the GT4 toolkit has been successfully built on Apple Mac OS X, Debian Linux, Fedora Core Linux, HP/UX, IBM AIX, Red Hat Linux, Sun Solaris and SuSE Linux [3] with implementations for 32 and 64 bit architectures. The Globus installation documentation and the installation email archives imply that the GT4 toolkit installation is most stable on Linux distributions with Red Hat and SuSE reporting the fewest problems [16]. Although most service implementations shipped with GT4 are marked as running on Unix only, all of the Java *clients* for those services are also supported on Windows [8]. In addition, Globus Alliance members report that work on porting other GT4 components to Windows is underway.

2.4 Support

The GT4 package is available free of charge by online download from the Globus Alliance web site [1]. The GT4 package is supported via online documentation [5,7], bugzilla bug/feature reporting and searching [4] and email discussion groups [6].

Commercial support for the GT4 tool kit is to be provided by Univa Corporation that aims to harness *the power of Globus to deliver enterprise-class products, technical support, and professional support to Governments and Universities* [13]. IBM provide commercial support for GT3 within the context of their Grid Toolbox product, and have spoken about providing similar support for GT4.

The email activity on the GT4 Friends email list implies significant GT4 experimentation and development activity within IBM, HP, Siemens and many smaller companies.

3. Systems Management

3.1 Documentation for System Managers

The GT4 documentation is well presented and provides general overview documentation and detailed documentation on particular toolkit components; the documentation is available at [5,7] and the documentation navigation page reproduced in Appendix II.

The documentation is available online and thus indexed by and easily searched using an online search engine. The GT4 pages provide a printer friendly button to make printing the documentation straightforward.

A criticism of the GT4 toolkit documentation is that it does require significant background reading to understand the range of software components available and the detailed configuration required to develop significant services. This is due to the range of issues that the toolkit addresses and that it relies heavily on other open source projects and tools. There is also a significant lack of examples that go beyond simple first steps in creating, managing and configuring services. At Globus week at NeSC the Globus team indicated that they would make available their own test examples in future GT4 releases.

3.2 Server Deployment

The GT4 package is straightforward to install with options to deploy a source or binary distribution.

3.2.1 Required Software

A GT4 installation requires

- Globus Toolkit installer, from Globus Toolkit [4.0 download page](#)
- J2SE 1.4.2+ SDK from Sun, IBM or BEA.
- Ant 1.5.1+ (1.6.1+ if using Java 1.5).
- C compiler. (If gcc, avoid version 3.2. 3.2.1 and 2.95.x are okay).
- GNU tar
- GNU sed
- zlib 1.1.4+
- GNU Make

- Sudo
- JDBC compliant database. For instance, PostgreSQL 7.1+
- gpt-3.2autotools2004.

and optionally requires

- IODBC (for RLS)
- Tomcat (for WSRF container)
- junit (for java unit tests)

All of these software packages are widely available and easy to obtain.

3.2.2 Deployment from Source

A source deployment consists of four steps

```
globus$ export GLOBUS_LOCATION=/usr/local/globus-4.0
globus$ ./configure --prefix=$GLOBUS_LOCATION
globus$ make
globus$ make install
```

as defined in [8]. The installation is compiled from downloaded source and thus will use the installed libraries available on the host machine; the configure step will report libraries that are missing and that are required if the installation is to succeed.

A range of installation options is outlined in [8] and reproduced in Appendix III.

The configuration of the compiled GT4 package requires

- a web service container certificate to be installed; and
- the definition of a grid-mapfile that assigns certificate names to local host user names.

At this point the Globus-provided web service container can be started that has example Globus services deployed.

3.2.3 A Binary Deployment

Binary deployments are available for Red Hat 9, Fedora Core 2/3, Debian Sarge, Solaris 9, SuSE 9 (Opteron) and SuSe 8 (Itanium). The appropriate binary package should be downloaded and installed using the commands defined below.

```
globus$ export GLOBUS_LOCATION=/usr/local/globus-4.0
globus$ ./configure --prefix=$GLOBUS_LOCATION
globus$ make
globus$ make install
```

3.2.4 Installation Report

The ETF evaluators report that installation on a Linux system is largely error-free and significantly easier than installed GT2 and GT3 services at a similar release stage. Further, the evaluators found that the installation document was accurate, well structured, easy-to-follow and comprehensive.

The ETF evaluation has installed on Fedora Core 1, 2 and 3, Red Hat Linux 7 and 9, Red Hat Linux Enterprise Edition 64 Bit, SuSE Linux without error.

The reviewers found some difficulty in installing the RLS component and additional problems for this component are reported on the email list. In addition, the reviewers found difficulty in deploying services into Tomcat/Axis rather than the default Globus container.

Adding new users to access a host follows the procedure used for GT2 and GT3 toolkits and requires that a user's certificate name be mapped to a local user account on the host machine¹. As was the case with the GT2 and GT3 toolkits, the management of a large-scale grid deployment is not addressed by the GT4 toolkit although the inclusion of MyProxy, the Community Authorisation Service, the Delegation Service and the Authorisation Framework packages does provide a basic set of functionality to build management services. There are also active groups in GGF, such as CDDLM [15], that are defining APIs for this purpose.

The GT4 software uses http and https as the transport protocol and as such should be significantly easier to configure for use through a firewall than GT2 and GT3 toolkits. An overview of the firewall requirements for GT4 is provided in [22].

The GT4 evaluators experimented with GT3.9.3, GT3.9.4, GT3.9.5 and the final GT4 toolkit release.

3.3 Client Deployment

The GT4 toolkit can be downloaded and installed in a variety of flavours that are focused on service development and deployment.

3.4 Account Management

As with GT2 based middleware, a user X509 certificate is mapped to a local user account and this mapping defined in the file `/etc/grid-security/grid-mapfile` [17].

3.5 Reliability

We have no quantifiable data on this. However, in deployed services for the Gridcast project migrated to GT4 from GT3.0.2, the GT4 based services showed greater reliability and significantly improved performance than their GT3 equivalents.

3.6 Distributed Management

The Globus toolkit does not provide any solution to distributed grid management but there are active GGF groups considering these issues. The *Community Authorisation Service* distributed with GT4 does provide a service framework for the creation of distributed community resource management systems and procedures.

3.7 External Services

No data available.

¹The gt4-friends mailing indicates that careful rendering of user DNs is required with capitalisation required if security is being used. We have not investigated this but there has been a number of errors reported on the list that have been tracked to conflict between lower and uppercase fields in the DN.

3.8 Scalability

3.8.1 Service Performance

The (largely) Java based GT4 middleware is built around and upon mainstream web technology, such Tomcat and Axis, and its performance is reliant on the performance of those web middleware solutions. The performance improvements in service execution in deployed GT4 services when compared to GT3 services suggests that service execution performance will be acceptable for most applications [18]. The development of supporting libraries for native C service implementations provides a route to develop applications that require higher performance. The packaging of the Reliable File Transfer service, Replica Location Service and (the much improved) GridFTP with GT4 provides a good mechanism for large-scale, managed and high-performance file movement.

3.8.2 GRAM

The GT4 GRAM implementation has been substantially re-engineered to allow for greater scalability both in terms of the number of jobs it can handle at any given time and in terms of response time for any specific job.

In both GT3 and GT4 toolkit releases, the main Globus server is responsible for submitting a user's job to whichever underlying cluster manager they is selected. For authentication, authorization and accounting reasons the Globus JobManager, the perl-based backend that controls the submission process, must run with the same UID as the user submitting the job.

In the original GT3, this was accomplished by spawning a new instance of the GT3 hosting environment as that user through the use of a custom `suid-root` binary. The new environment then runs the JobManager once to submit the job to the local batch system and then periodically re-invokes a `poll()` method to query the state of the job. The overhead that these layers incurred was substantial, usually taking tens of seconds on an unloaded twin-CPU machine to submit a job from the local host with the Java client provided. This approach did not scale with large numbers; as the number of polling processes and user-specific JVMs increased the pressure on CPU, memory, and kernel resources became excessive.

The new implementation removes the use of the `poll()` completely; instead of periodically querying the underlying job management system a separate module has been introduced, called a Scheduler Event Generator, or SEG. The SEG is responsible for continuously monitoring the state of the cluster manager (eg by monitoring a logfile generated by the cluster tools) and sends asynchronous state-change notifications when the state of any job updates. This allows for a much faster response to job state changes and can substantially reduce the overhead involved in constantly polling the state of the running system.

The new implementation also has a number of other improvements; rather than spawning a new JVM for the invoking user, a lightweight pre-compiled C application is used to drive the Perl JobManager modules directly - resulting in a much faster turnaround time and substantially reduced resource requirements. A new precompiled `globusrun-ws` command has been introduced in the latest development release, which allows for much for efficient *submission* of jobs by the client - it is no longer necessary to start and stop a JVM for every job. Finally, rather than use a custom `suid-root` binary to perform the user-switching operations, GT4 now simply uses `sudo`, a standard unix tool which allows for secure, fine-grained privilege selection.

David McBride completed a release of the Sun Grid Engine integrated with GT4 in the course of this evaluation [21].

4 User Experience

4.1 User Documentation

The GT4 documentation is comprehensive, well structured, accurate and presented in a consistent style—defining an overview of the key concepts before presenting detailed documentation on components. The documentation is a *significant* improvement over documentation for previous GT releases.

There is also a service-building tutorial by Borja Sotomayor [9]. A new initiative is being conducted directed by Sotomayor, called the Globus Documentation Project [10], aims to collate references to good documentation. Hopefully, this new project will provide a repository for Globus documentation.

A criticism of the documentation is that it tends to require experience with grid technology and previous versions of the toolkit to understand the layout, configuration and references within the pages. (This is an unfortunate consequence of the field and the technology used and there is no easy solution.)

Keyword searches are supported for the entire Globus site (rather than just the documentation) but these often direct users to the mailing list archive, itself a useful, yet unstructured, source of information.

4.2 Joining the Grid

Assuming that the definition of a *Grid* refers to a set of resources that have been grid-enabled by integrating them with the Globus Toolkit, the questions posed in the evaluation criteria are

How does the user join the Globus deployment?

- In a Globus deployment a set of services will be published on start-up of the service container all with the root name of <http://localhost:8080/wsrf/services> (if using the “no security” flag, if security is used then the protocol will be “https” and the default port will be 8443). The individual services will be identified depending on what name has been designated to them in their deployment descriptor file. To access these services the service URI will be specified somewhere in the client class, and this will be used to find the endpoint-reference (epr) address of the service resource. Once this is established, the client class can modify the properties of that resource.

How does the user authenticate to the container?

- To authenticate to the Globus container, the user must have security credentials in the form of a user certificate and private key. These can be issued from any certificate authority as long as that authority is recognised - this is done by pointing the correct environment variables to a root certificate and signing policy.

How does the user obtain the tokens necessary to authenticate to the container?

- The most common method of obtaining such credentials is to contact the UK Grid Operations Centre [12], obtaining a p12 file from them and extracting the necessary user key and certificate following the instructions provided. Alternatives include setting up a local certificate authority using the Globus tool, SimpleCA, or by using OpenSSL directly, a security application provided with most Linux bundles. When using either of these tools, a root certificate must be set up and user certificates/keys created from this. Again the root certificate and a signing policy must be made available. The major drawback with this approach is that these certificates will only be trusted over the local domain whereas UK e-Science certificates are trusted on a national and international scale.

4.3 Legacy Application Integration

The integration of legacy applications and GT4 was considered through the integration of PERMIS software with GT4.

4.3.1 GT4 integration with PERMIS

PERMIS is a security middleware solution that provides an interface to make use of the standardised GGF authorization API in the Globus Toolkit. In the BRIDGES and DyVOSE projects at NeSC, PERMIS has been used to provide a Role-Based Access Control infrastructure to protect grid services in an effective yet flexible way. The original configuration for PERMIS was to be used with Globus 3.3 services implemented in these projects and it was therefore in the interests of NeSC to migrate PERMIS to GT4 functionality.

Code re-factoring within the PERMIS application was required as the PERMIS code made use of certain in-built OGSI classes that had to be changed to account for the WSRF structure. The code base of the WSRF services has changed (three service implementation classes instead of one) requiring the PERMIS client code and the deployment methods to be changed as well.

The development overhead associated with this re-factoring was significant. However, it was a “one time” effort and now that this issue has been solved, the application performs the same functions as before. Indeed the deployment and overall structure is simpler to understand – the location and use of the configuration files that make up the PERMIS service, in the context of Globus services, are more logical and structured than before.

In summary, the application appears to the user to function exactly as when run with previous Globus versions. The migration from v3.3 to v4.0 required significant investment of developer time but the resulting product is more structured and easier to deploy.

4.4 Migration between platforms

There are two² Globus based user communities—GT2 based and GT3 based.

4.4.1 Use of GT2 with GT4

The GT4 distribution bundles a collection of pre-WS components. The pre-WS components are, in effect, the components of the GT2.4 release. These components can be used to permit systems and clients using GT2 deployments to interact with systems and clients with GT4 deployments. An overview of GT2.4 compatibility with GT4 is provided in [19] and summarised below.

- **XIO.** XIO was introduced in GT 3.2 and it includes a compatibility layer to allow full backward compatibility for applications written using `globus_io`. The APIs for `globus_io` and XIO use the same event model and only significantly differ in connection establishment.
- **C Common Libraries.** No changes need be made with GT2.x releases.
- **GridFTP.**
GT 2.2 or earlier is deprecated and unsupported, and has a protocol incompatibility with later versions due to an error in the security code. GT2.2 clients will not work with newer servers (GT2.4.0 and greater) and new clients will not work with GT2.2 servers.

² There are also many flavours of GT2!

GT2.4.0 or greater, the migration for GridFTP is relatively painless with only new features added. No changes were made to the existing protocol or APIs, so any existing client or application built using the client APIs will work unchanged. Any combination of old/new client/server will work.

- **RLS.** Users of GT2 RLS using a deployed RLI must create the RLI database, which was not required in some GT2 versions of RLS. A script is provided to perform this task.
- **GRAM.** There are significant differences in architecture and implementation that will require modification to the GT2 code.
- **Command Line Tools.** Typical interactions with pre-WS GRAM service are using either the `globusrun` or `command, globus-job-*` commands. In WS GRAM, the `globusrun-ws` command implements the functionality of `run` commands using the XML Job Description language (instead of RSL) and permits additional command line arguments.
- **Developer API and RSL.** APIs and RSL fields have changed and a mapping table is provided in [19].
- **WS-MDS.** The basic functionality between GT2 MDS and GT4 WS-MDS is broadly similar. The implementation, architecture and standards have, however, changed significantly. A summary of the changes is defined in [19] and reproduced in Appendix IV.
- **GSI-OpenSSH.** No changes are required.
- **Pre-WS A&A.** No changes are required.
- **MyProxy.** No changes are required.

The status of the pre-WS components is defined by Ian Foster below³

The ****pre****-WS components of GT4 (our terminology for "GT2 in GT4") are all a supported part of GT4. Some specifics:

- GridFTP: this is our new server, which replaces the old server. GT4.0 GridFTP is backward compatible with GT3.2 GridFTP
- GRAM: GT4.0 pre-WS GRAM is backward compatible with GT3.2 pre-WS GRAM
- MDS: GT4.0 pre-WS MDS is backward compatible with GT3.2 pre-WS MDS.

In terms of support, pre-WS MDS is specified as "deprecated" in GT4.0, meaning it won't be in GT4.2 and that support will cease from GT4.4 onwards.

We don't have a plan yet to deprecate pre-WS GRAM. Our goal is to move major U.S. projects to WS GRAM as soon as possible, given that WS GRAM appears to be superior to pre-WS GRAM in just about every respect. Continued support depends in large part on how quickly the projects that we are funded to support make that transition. I haven't discussed this with my colleagues, but I would hope that we could stop supporting pre-WS GRAM a year from now.

To test the inter-operability of GT2.4 and the pre-WS components of GT4, BeSC conducted inter-operability tests using the Level 2 R-Matrix GT2 test application. The application typically runs for 2 hours, creates around 30+ interacting parallel tasks and moves 200+M of data with 100+ file movements. The application does not use GT2 MDS but focuses on data movement (GridFTP) and jobs execution (`globus-job-run`) and thus the above migration issues between GT2 MDS and WS-MDS were not tested.

The tests used a 32-node blade cluster at BeSC. The application was executed as a GT2.4 application using compute nodes with pre-WS components of a GT4 installation, as a pre-WS GT4 application using GT2.4 compute nodes and as a combination of GT2.4/pre-WS

³ From a private email exchange.

middleware with some compute nodes using GT2.4 middleware and other pre-WS middleware from a GT4 installation.

The application was executed using our standard collection of test data for R-Matrix. No problems were encountered and all runs completed successfully.

4.4.2 Migration between GT2 to GT4

There is currently no guide available to provide a systematic mechanism for GT2 to GT4 migration. The GT2 and GT4 toolkit models are considerably different and significant re-development of the code is to be expected in such a migration. BeSC is actively converting the R-Matrix GT2 package it developed for Level 2 grid testing to be a GT4 application—the effort has proven significant thus far although making R-Matrix work within a GT4 environment (as a pre-WS application) require almost no effort.

4.4.3 Migration between GT3 and GT4

A guide has been written in the course of this evaluation that outlines a systematic approach to GT3 to GT4 conversion—the guide is the collective view of the Gridcast developers that migrated the Gridcast GT3.0.2 code base to GT4. The guide [11] is a work in progress and has been used in covering 250k of Java 3.0.2 code to GT4 that has been successfully deployed.

The Gridcast conversion required significant changes to service descriptions and deployment definitions that, for the Gridcast code base, could be automated by using a collection of Perl scripts⁴ to create the new GT4 service form from the original GT3 form. The GT3 service required very few changes to convert it to be a GT4 service.

The Gridcast conversion exposed some problems with the deployment of services within the Tomcat/Axis service container—these were mainly path and location issues that were time consuming to location although they were relatively easy to correct. The email lists have discussions of Tomcat problems with the GRAM services [23].

4.5 Usability

Once security has been set up, using the example services is fairly straightforward. The counter client provided as a demonstration service is an easy introduction to the services provided. There is also a Java client that can be invoked to create a service resource, which can then be manipulated using the various new command line “wsrf-“ commands.

However, as always, the ease of use of this (and any) product depends largely on the documentation that is available. The Sotomajor tutorial provides a useful introduction and explains clearly the procedures and reasons behind using web services and the reason for the conceptual move from OGSA to WSRF.

A tutorial has been written that covers slightly more advanced topics in grid/web service development. It covers implementation of basic security methods, role-based access control with PERMIS and implementation of logging techniques using the log4j project [20].

⁴ These create the new WSDL from GWSDL, the JDNI deployment descriptor, the server deployment definition, GT4 web service and the resource home. These scripts took approximately 3 hours to create.

4.6 Security from the user's perspective

Security requirements in this version of the toolkit are much more stringent and therefore, probably much more effective (as there has not been widespread deployment of the toolkit yet, there are very few builds “in the wild” on exposed networks). Security workarounds for “lazy” implementations of the toolkit are much more difficult to put into practice. For instance, to avoid having to obtain credentials, the full-source toolkit must be started with a “No security” option explicitly specified. Even if this less secure option is chosen, various services cannot be run without secure credentials being presented to the server process from the client, so any insecure clients are effectively barred from running services that demand security.

The flip side to this rigorous security is that, even though a certain level of security is demanded, the documentation and processes that allow a user to set up such security can be difficult for new users. Difficulties have been encountered when attempting to write, compile and run secure clients (by setting encryption and signature demands on the client stub).

Examples of new security requirements are:

- A global-security-descriptor file that names the certificate/key credentials that the container will use to start up.
- That the same credentials are used to shut the container down. This is a rather syntactically involved process that can easily go wrong.
- Security credentials being present before simple example services, such as the counter service, can be invoked. This is particularly problematic for beginners, who will invariably use the counter service as their first attempt to use the container.
- Stub properties can be applied using either direct coding or security descriptor files on the client side.
- A SAML (Security Assertion Markup Language) Callout parameter is available that allows easy and flexible integration with OGSA-authz compliant tools, such as PERMIS [14].
- New authorisation methods are available such as *identity*, where the client identity subject must match that specified in the WSDD file, and *userName*, where a JAAS login module authorises the user based on the username and password that they supply.
- Resource level descriptors are now available that allow specification of authorisation of the service resource. This makes for security policies (i.e. specification of grid-map files) that are more flexible than was possible with previous toolkit versions.

The security measures used with GSI have changed their focus to transport-level mechanisms rather than message-level mechanism. This is with a view to improving performance levels when invoking secure services. The GSI implementation requirements on simple service stubs are not very different to those required for previous versions of the toolkit (as described above – with properties being assigned either directly or using security descriptors).

Server side security is specified as in previous versions – by pointing to a security descriptor that denotes what type of authentication is to be used on a per-method basis and what identities the various subjects run as.

On the stripped down versions of the toolkit (WS-core binary, source and the ‘wsrf’ module), the security demands are more relaxed, which enables users to gain an understanding of the basic features of the toolkit without grappling with obscure security issues.

4.7 Verification

When dealing with security issues, it is not always easy to discern whether the error messages returned are coming from Globus or the underlying security concerns of middle-ware such as

SSH. Clarification of these messages was raised as a concern at Globus-Week UK and it has been noted by the Globus team that more work is required in this area.

Beyond security concerns, most error messages encountered, in NeSC testing, have nearly always related back to problems beyond Globus (“Connection refused”, SOAP messages, HTTP messages, etc.) In original alpha versions of this toolkit, there were some client-side error messages that were quite generic (“HTTP server fault”). But, in later versions, similar problems now seem to convey more information when they produce exception messages.

5. Developer Experience

5.1 Development documentation

The documentation exists but so far it is mainly in reference format on the Globus website. There are few guides and tutorials to aid in learning the development tools that are available. The mailing lists, combined with developers’ innate ability to work round problems seem to have brought about the most effective solutions in the evaluation.

5.2 Languages and tool support

The evaluation has coded (for the most part) in Java—the documentation and guides for Java are well developed. Other supported languages are C and Python. The documentation for C is less comprehensive but is adequate for its purpose. The documentation for Python does not seem as comprehensive as that for Java and C⁵.

Support for the integration of Globus services and portal development, such as GridSphere, is in development but is not currently available. An initial implementation of GridSphere Portlets supporting GT4 is completed and under test but there is no information on a release schedule⁶.

5.3 Developing new services

Simple services can be developed quickly with relatively little understanding of the web service model. Extended services require more investment of time and a change in thinking compared to the “OGSA way” of implementation. The WS-RF structure does make the service paradigm commensurate with that of regular web services.

6. Conclusions

The GT4 distribution documentation is well structured, well presented and accurate. The documentation includes overview documentation on the toolkit, installation procedures and options, and GT4 component-by-component descriptions. The documentation is a significant improvement on previous GT releases.

The deployment of the toolkit can be from a binary distribution available for most of the popular Linux distributions or from a source distribution. A binary installation can be performed quickly making the process of grid deployment relatively speedy. A source distribution can take two+ hours to be completed. The installation of GT4 software is straightforward and well documented. The installation process is a significant improvement on previous GT releases and it does not require significant grid experience or training to perform.

⁵ We do not have Python experience so it is hard to judge!

⁶ Private email communication from Michael.Russell@aei.mpg.de

The development of simple GT4 grid services is relatively easy and is well documented in the Globus documentation and in the Sotomayor grid service tutorial [9]. The Java GT4 services require additional definitions to be created and managed but this has not proven a significant additional development effort. The development of larger-scale services has, in this evaluation, exposed gaps in the documentation but these have not proven to be significant obstacles and are understandable given the GT release schedule. Deployed GT4 services have demonstrated good reliability and significant performance improvements on their GT3 equivalents. BeSC has two live GT4 user field deployments that have demonstrated better reliability and significant performance improvements on their GT3 equivalents.

GT2.4 clients can inter-operate with servers and clients using the pre-WS components of a GT4 deployment. This appears to be one route to provide users with the capabilities of GT2 and GT4 using a single middleware deployment—although migration of GT2 MDS to WS-MDS is likely to require significant additional effort. GT3 to GT4 migration has not proven to be a significant effort. The migration of code from GT2 to GT4 is still a largely unexplored area but is likely to require significant re-coding effort.

References

1. www.globus.org
2. <http://www.globus.org/toolkit/legal/4.0/license-v3.html>
3. <http://www.globus.org/toolkit/downloads/4.0.0>
4. <http://www.bugzilla.globus.org>
5. <http://www.globus.org/toolkit/docs/4.0/index.html>
6. <http://www.globus.org/about/subscriptions.html>
7. http://www.globus.org/toolkit/docs/4.0/doc_overview.html
8. <http://www.globus.org/toolkit/docs/4.0/admin/docbook/ch04.html>
9. <http://gdp.globus.org/gt4-tutorial>
10. <http://gdp.globus.org>
11. <http://www.qub.ac.uk/escience/howtos>
12. <http://ca.grid-support.ac.uk>
13. <http://www.univa.com>
14. <http://sec.isi.kent.ac.uk/permis>
15. <http://sourceforge.net/projects/deployment>
16. <http://www.globus.org/toolkit/docs/4.0/admin/docbook/ch03.html#s-platform>
17. <http://www.globus.org/toolkit/docs/4.0/admin/docbook/ch05.html>
18. http://www.globus.org/toolkit/docs/development/3.9.5/perf_overview.html
19. http://www.globus.org/toolkit/docs/4.0/migration_guide_gt2.html
20. <http://www.nesc.ac.uk/hub/projects/etf>
21. <http://www.lesc.ic.ac.uk/projects/SGE-GT4.html>
22. <http://www.globus.org/toolkit/security/firewalls/Globus%20Firewall%20Requirements-6.pdf>
23. http://bugzilla.globus.org/bugzilla/show_bug.cgi?=3218

Appendix I: The Globus Toolkit Public License

Globus(R) Toolkit Public License

Version 3, March 2005

<http://www.globus.org/toolkit/license.html>

(This license is identical to the Apache License, Version 2.0, January 2004, <http://www.apache.org/licenses/LICENSE-2.0>, except for the addition of an explicit U.S. Government disclaimer and limitation of liability in paragraphs 7 and 8.)

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,

any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor and the U.S. Government provide the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You and third parties are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor or the U.S. Government be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor or the U.S. Government has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Globus Toolkit Public License to your work.

To apply the Globus Toolkit Public License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Globus Toolkit Public License, Version 3 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.globus.org/toolkit/license.html>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Appendix II GT4 Documentation

The Globus Documentation Project – <http://gdp.globus.org/>

Toolkit-level documentation

- [Documentation Overview](#)
- [Release Notes](#)
- [GT4 Key Concepts](#) [pdf]
- [Installing GT 4.0 \(System Administrator's Guide\)](#) [pdf]
- [Platform Notes](#)
- [Best Practices](#)
- [Coding Guidelines](#)
- [Public Interface Guides \(APIs, command line, etc\)](#)
- [Migrating from GT2 to GT4](#)
- [Migrating from GT3 to GT4](#)
- [Samples](#)
- [Overview and Status of Current GT Performance Studies](#)
- [Release Version Scheme](#)

Component Level Documentation

- [GT 4.0 General](#)
 - [GT 4.0 Release Notes](#)
 - [GT 4.0 Key Concepts](#)
 - [GT 4.0 System Administrator's Guide](#)
 - [GT 4.0 Developer's Guide](#)
 - [GT 4.0 User's Guide](#)
 - [Overview and Status of Current GT Performance Studies](#)
 - [GT 4.0 Resource Properties Summaries](#)
- [GT 4.0 Common Runtime Components](#)
 - [Common Runtime Components: Key Concepts](#)
 - [Java WS Core](#)
 - [C WS Core](#)
 - [XIO](#)
 - [C Common Libraries](#)
- [GT 4.0 Security \(GSI\)](#)
 - [Security: Glossary](#)
 - [Security: Key Concepts](#)
 - [WS A&A](#)

- [Community Authorization Service \(CAS\)](#)
 - [Delegation Service](#)
 - [Authorization Framework](#)
 - [Message/Transport-level Security](#)
- Credential Management
 - [MyProxy](#)
 - [SimpleCA](#)
- Utilities
 - [GSI-OpenSSH](#)
- [Pre-WS Authentication & Authorization](#)
- [GT 4.0 Data Management](#)
 - [Data Management: Key Concepts](#)
 - [RFT](#)
 - [GridFTP](#)
 - [RLS](#)
- [GT 4.0 Information Services](#)
 - [Information Services: Key Concepts](#)
 - WS MDS (MDS4)
 - [Aggregator Framework](#)
 - [Index Service](#)
 - [Trigger Service](#)
 - [WebMDS \(Tech Preview\)](#)
 - [Pre-WS MDS \(MDS2\)](#)
- [GT 4.0 Execution Management](#)
 - [Execution Management: Key Concepts](#)
 - [WS GRAM \(GRAM4\)](#)
 - [WS Rendezvous](#)
 - [Pre-WS GRAM \(GRAM2\)](#)

Appendix III Installation Options

1. Optional Features:
2. `--enable-prewsmds` Build pre-webservices mds. Default is disabled.
3. `--enable-wsgram-condor` Build GRAM Condor scheduler interface. Default is disabled.
4. `--enable-wsgram-lsf` Build GRAM LSF scheduler interface. Default is disabled.
5. `--enable-wsgram-pbs` Build GRAM PBS scheduler interface. Default is disabled.
6. `--enable-i18n` Enable internationalization. Default is disabled.
7. `--enable-drs` Enable Data Replication Service. Default is disabled.
8. [...]
9. Optional Packages:
10. [...]
11. `--with-iodbc=dir` Use the iodbc library in dir/lib/libiodbc.so.
12. Required for RLS builds.
13. `--with-gsiopensshargs="args"`
14. Arguments to pass to the build of GSI-OpenSSH, like
15. `--with-tcp-wrappers`

APPENDIX IV: GT2 MDS versus GT4 MDS

Description	GT2 Version	GT4 Version
Format of data describing a resource	LDAP data hierarchy	XML data document
Query language	LDAP queries	XPath queries
Wire protocol for queries	LDAP	WS-ResourceProperties
APIs used for queries	LDAP APIs	WS Core APIs
Command-line clients used for queries	grid-info-search	wsrf-get-property, wsrf-get-properties, wsrf-query
Available GUIs	Various LDAP browsers	WebMDS
Wire protocol for subscription/notification	Not supported	WS-Notification
APIs used for subscription/notification	Not supported	WS Core APIs
Security support	SAML-based security using X.509 user, proxy and host certificates	HTTPS-based security using X.509 user, proxy and host certificates
Queryable index of aggregated information	GIIS, which publishes data using the LDAP-related standards listed above	WS MDS Index Server, which publishes data using the WSRF-related standards listed above
Queryable source of non-aggregated information	GRIS, which uses information providers to gather data from services and then publishes that data the LDAP-related standards listed above	Individual web services, which publish data about their own resources using WSRF-related standards listed above.
Index registration mechanism	MDS servers (GRIS's and, in some cases, GIIS's) register themselves with a GIIS. An MDS server is configured to register itself to a remote index by editing the local MDS server's <code>grid-info-resource-register.conf</code> file, providing information about the location of the remote index to register to and timeout values for the registration	WS MDS Index servers maintain aggregating service groups that include registration information (timeout values, the mechanism to use to acquire information, and additional mechanism-specific parameters) The registration is accomplished by adding an entry to an aggregating service group via the <code>mds-servicegroup-add</code> command. In addition, services may be configured to register themselves to the default index server running in the same container.
Mechanism used by an index to collect information	GIIS's send LDAP queries to remote serves.	WS MDS Index servers use a plugin-based architecture to support several mechanisms to collect information. The Globus Toolkit supplies plugins that support collecting information via polling (resource property queries), subscription/notification, and by program execution.