

# GT4: What's in it for you?

LEE LIMING AND IAN FOSTER

At the time this column was written, Globus Toolkit version 4.0 (hereafter referred to as "GT4") is scheduled for official release in April 2005. The GT4 release cycle has consumed more than a year of effort from the Globus Toolkit development team, a collaboration of open source developers that includes employees of the core member organizations of the Globus Alliance as well as numerous individuals from other organizations. After several schedule "slips," a reasonable question you may ask is, "What took them so long, and what's in it for me?"

At the highest level, the changes between GT4 and GT3.2 fall into four broad categories:

- "User experience" improvements
- Performance improvements
- Upgrade to latest versions of Web services standards
- New features

While most of the written information about GT4 has focused on new features, some of the most profound changes in the GT4 are actually in the other areas. In this column, we describe these changes and their significance for you, the user.

This column represents the Globus Toolkit development team's understanding of what we have done in GT4 over the past year. Early GT4 users have confirmed much of what is described here, but independent reviews are yet

to be written, and we look forward with great interest to hearing what the broader community will have to say about this latest edition of the Globus Toolkit.

## User Experience Improvements

Without diminishing GT4's exciting new capabilities (which we describe in later sections), the improved user experience is probably the most profound change from previous versions. Intrepid users of the Globus Toolkit 1.0 series were accustomed to research software that would never see use in business or beyond narrow fields in science and engineering. The user community has since broadened and taken in less adventurous people from many walks of life, leading to increased expectations for usability. Until now, the toolkit has not kept pace with these rising expectations, leading to an overall sense that the Globus Toolkit is hard to use.

GT4 system administrators and users will experience two major improvements and will likely benefit from a third improvement without realizing it.

The first major improvement is that the documentation has been completely overhauled and significantly expanded, providing users with more consistency in the quality and content of documentation across all

toolkit components and considerably more documentation in general. The team took a top-down approach to defining the documentation requirements for the release. Required documentation elements (e.g., "Key Concepts") are provided for each component and for the toolkit as a whole. Other required sections include those on backward compatibility, tested platforms, a developer guide, a system administrator guide, and a user guide. This consistency across components has already been cited by early users as an important element in making the documentation more usable.

The development team has devoted much effort to documenting the GT4 code and its capabilities, thanks in large part to funding from IBM and the National Science Foundation (NSF)'s Middleware Initiative (NMI). Quantitatively, the GT4 documentation is more than double the size of previous releases. Qualitatively, the consistency improvements make it far easier to find the answer to a given question.

The second major improvement is a streamlined installation mechanism. The new installer allows users and administrators to pick the parts of the Toolkit they need installed; it then skips the other components, avoiding unnecessary configuration and administration tasks. The automatic system configuration tools,

autoconf, have been updated to the latest public releases so that more platforms are covered. If something stops a build from completing, restarting is much easier now.

Because the Globus Toolkit is used on a wide variety of systems, some quite exotic, we continue to solicit information from the user community about how to build the software on unusual systems. NMI's distributed software build system has greatly expanded the range of systems on which we test code during development.

The third major improvement—which users will benefit from without realizing it—is in how the development team tests the Globus Toolkit. Following a test coverage analysis, the development team added hundreds of new tests that are run automatically during development and prior to releases. These tests are run on a variety of popular hardware/OS platforms. Automated testing has enabled the team to find and fix many more bugs prior to release.

The team has also built testbeds for robustness (how well the software stands up to continuous use) and performance (how efficiently the software performs). All of this testing has yielded significantly more prerelease bug fixes, more performance enhancements, more documentation improvements, and—in general—better software.

The development team is not the only group testing GT4. The first public “preview” release of the GT4 Web services code was in July 2004. Subsequent preview releases in August,

October, December, and January added more toolkit components. These preview releases mean that when you start work with the April GT4 release, you will have been preceded by hundreds of early users who have already identified problems that the development team fixed prior to the official release.

## Functionality Changes

The purpose of the Globus Toolkit is to provide useful functionality for Grid system and application builders. While the user experience improvements mentioned earlier make this functionality easier to access, in the end it is *capabilities* that make the Globus Toolkit useful.

The non-Web services software in previous versions (GT2.4 through GT3.2) is still present in GT4. This software predates the Grid community's embrace of the Web services framework; but since many production Grids still use these components, we continue to provide limited support for them. Specifically, GT4 provides bug fixes and some improvements in documentation. The Globus Toolkit development team routinely tests the non-Web services components prior to release to ensure that they continue to work as expected. We expect to demonstrate during 2005 that the Web services components provide superior performance and functionality, encouraging migration.

The most significant non-Web services change to GT4 is the new GridFTP server. The GT4 GridFTP server has been rewritten from scratch. (The previous

implementation was a modified version of Washington University's wuftp server.) The new implementation is based on the high-performance Globus XIO communication library and features significantly better performance and capabilities not provided by the older code. It also eliminates a subtle licensing issue that caused concern for some people who wanted to redistribute the older server. The new server is fully compatible with both the GT3.2 server and the published GridFTP protocol, so older clients work with the GT4 server and GT4 clients work with the GT3.2 server.

The GT4 features that have generated the greatest attention are undoubtedly the new Web services capabilities. Like GT3, GT4 includes a programming model and associated tools that allow users to build and host Web services that represent “stateful” resources on the Grid. In other words, users can use Web services development tools and hosting environments to provide Grid interfaces to computation engines, storage systems, legacy applications, instruments and sensors, and other things that make up a Grid application. Once Grid interfaces are provided, users—and their partners and collaborators—can develop all kinds of applications that use them in creative ways.

New Web services features in GT4 include a more stable hosting environment, better performance and efficiency, broader programming language support, and a new state model that implements

the WSRF (Web Services Resource Framework) and WSN specifications. (See the March 2005 "On the Grid" column for more detail on WSRF and WSN and what they mean for the Grid.) All Web services included in GT3.2 are included in GT4, with the same or better capabilities. The WSDL interfaces for these services (used by Web service developers) have changed from their GT3.2 counterparts as a result of the introduction of WSRF and WSN.

GT3.2 provided tools for developing client programs in C or Java and provided Java classes for developing Grid services. GT4 expands both client and server programming support to include C, C++, Java, and Python.

GT4 also includes new system security capabilities. GT4 provides message-level security mechanisms that provide message protection for SOAP messages based on the WS-Security and WS-SecureConversation specifications. Developers can use GT4 to build applications and systems that are compliant with the WS-Interoperability Basic Profile and Basic Security Profile. Transport-level security mechanisms are also supported. A new authorization framework supports a variety of authorization schemes, including the familiar "grid-mapfile" access control list, an access control list defined by a service, a user-supplied authorization handler, and access to an authorization service via the SAML protocol. Security services distributed with GT4 include the MyProxy online credential repository and the

Community Authorization Service (CAS). In addition, GT4 interoperates with the Virtual Organization Management Service (VOMS) and PERMIS.

Finally, the GT4 Monitoring and Discovery Services ("MDS4") provide significantly enhanced functionality. Every GT4 Web services container is preconfigured with an MDS-Registry service that maintains information about services deployed in that container. MDS-Registry services can also be configured to monitor not only GT4 services but also, via a plug-in interface, any network-accessible resource or service. An extensible display component uses XSLT templates to define custom displays of MDS-Registry contents.

## Performance Improvements

The GT4 development team placed a high priority on improving performance and scalability, and the results are impressive. This column was written while GT4 performance tuning efforts were still in progress, but significant improvements had been achieved already.

As noted above, GT4 includes development support for C and C++ clients and services. This support is useful for Web service client programs that start up, do one thing, and then quit. For example, the GRAM job submission client is now written in C. By eliminating the need to start a Java Virtual Machine (JVM) to run this program, start-up cost is reduced by 80 percent.

GridFTP has long been the performance star of the

Globus Toolkit. As noted above, GT4 includes a completely new GridFTP server implementation. This server consistently performs at roughly 80 percent of the raw iperf performance on a network. This result has proved true on networks ranging up to one gigabit per second end-to-end, where iperf performed at roughly 940 Mbit/s and GridFTP performed at 750 Mbit/s. In our testing, data transfer rates have always been limited by the performance of the disk subsystem or the network interface card, never by the software. In a load test, a server running on a dual-processor Linux system supported 1800 clients simultaneously, sustaining a combined throughput equivalent that of an unloaded server, demonstrating that the server scales extremely well under heavy loads.

The new GT4 GridFTP server also supports *striping*, where the server runs on a cluster with a shared parallel filesystem and multiple nodes are used for individual transfers. On the TeraGrid's 30 Gb/s network, a striped transfer using 64 nodes at each end performed at 17 Gbit/s, limited only by the speed of the disk subsystem. (A memory-to-memory striped transfer using 32 nodes at each end sustained a rate of 27 Gbit/s—an amazing 90% of the theoretical limit!)

Performance of the GRAM job submission service was a major (and justified) criticism of GT3.0 and GT3.2. Performance testing of GT4's GRAM service shows significant improvement. Since the initial GT4 implementation in mid-2004, design improvements and

## Something for everyone

Everyone who currently uses the Globus Toolkit should find something new to be happy about in GT4.

### Web service developers

- Improved WS Core performance
- Upgrade to latest Web service specifications

### Security architects

- Support for HTTPs, WS-Security, WS-SecureConversation, and WS-I Basic Security Profile
- Addition of MyProxy and CAS
- Interoperability with VOMS, PERMIS, and other SAML-based authorization systems

### Network engineers

- Extremely fast GridFTP server that includes a striped server configuration
- Reliable File Transfer and Replica Location Services that optimize network use

### Computational scientists

- Remote job submission service that scales well under high loads
- Improved and expanded documentation and training materials

profiling activities have improved performance by more than a factor of ten. The GT4 job submission service supports at least 10,000 concurrent job submissions on a reasonably configured system. The service can process up to 70 independent jobs per minute under normal (multiple user) scenarios. For scenarios in which a single user must submit many jobs at a high rate, a new delegation service streamlines security processing to attain faster rates. In addition, GT4 GRAM uses the RFT service (described next) in place of the old GASS service to manage data staging, eliminating redundant code and also giving the system administrator control over the maximum number of concurrent staging operations.

GT3 introduced the Reliable File Transfer (RFT) service. RFT accepts requests for GridFTP transfers between systems and processes them in sequence until they are complete, allowing the client application to continue working during the transfers. At the time this article was written, the development team had accomplished a 300 percent improvement in the number of requests that the RFT service could handle with default configuration settings for the underlying Java Virtual Machine. (At that time, the service could manage about 21,000 concurrent requests.) The development team conducted tests on the RFT service that required transferring more than 500,000 files. As another example of GT4 service scalability, we note

that the Replica Location Service (RLS) has been used by the LIGO Scientific Collaboration for some time now to manage 40 million replicas across 10 sites.

The Web services development environment for Java (WS Core) also improved greatly during the GT4 development cycle. From the first implementation in early 2004 to the time this article was written the messaging latency of the Web services environment (the time required to move a Web service message from the network interface to the service handler and return a response to the network interface) was reduced by 80 percent. The development team changed the default authentication method from WS Security (a Web services specification) to HTTPs (an older specification). This change significantly improved the time required to authenticate a service request—particularly in streaming scenarios—which had a major impact on all of the GT4 Web services tools. WS Security continues to be supported for applications that need it.

## Related Tools

A strength of the Globus Toolkit has always been the many complementary tools developed by other groups. The number of such tools continues to grow, and with GT4 we dedicate a section of the Globus Toolkit Website to describing their features and how they can be used in conjunction with GT4 to address application requirements.

Among the many featured tools are the Nimrod-G parameter study

system, the Condor-G job submission tool, the Ninf-G remote procedure call system, the Grid Resource Management System (GRMS) metascheduler, and the Open Grid Computing Environment (OGCE) software for portals. Related software includes GRAM interfaces for Condor, Platform's Load Sharing Facility, Sun Grid Engine, and the Portable Batch System, as well as group membership and authorization services accessible via the new authorization framework.

## Conclusion

GT4 has been in the hands of early users for nearly a year now and has been tested extensively by both the Globus Toolkit development team and hundreds of early adopters. These early users report major improvements relative to previous Globus Toolkit releases in the areas of user experience, features, and performance.

We are grateful for the substantial investment in usability and performance made by the National Science Foundation and IBM, combined with the core research and development support provided by NSF, DOE, NASA, DARPA, the European Commission, the UK Research Councils, IBM, and Microsoft. Moreover, we believe that the personal investments in time and effort to learn about and use GT4 in hundreds of science, engineering, and business projects will clearly pay off in personal and professional accomplishments.

Continued support of Grid computing is possible only when success stories are known and shared with others. We'd like to hear

about your success stories! (Problems are always of interest to us as well.) We encourage users to send e-mail to [info@globus.org](mailto:info@globus.org) with interesting experiences using GT4.

*Globus is a registered trademark held by the University of Chicago. This work was supported in part by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under Contract W-31-109-ENG-38 and under Contract DE-AC03-76SF0098 with the University of California; by the National Science Foundation; by the NASA Information Power Grid program; and by IBM.*

*Lee Liming is manager of the Distributed Systems Laboratory (DSL) at Argonne National Laboratory and the University of Chicago.*

*Ian Foster is the Associate Division Director of the Mathematics and Computer Science Division, Argonne National Laboratory and Professor of Computer Science, University of Chicago.*