



Issue 2 • February 2003

Contents:

- 1 - Editorial
- 2 - The GRIDS Center Software Suite:
Toward a National Middleware Infrastructure
- 4 - DATATAG:
Middleware Interoperability
- 7 - DATAGRID:
The European DataGrid Middleware Architecture
- 10 - GRIDLAB:
Grid Application Toolkit & Testbed
- 12 - UNICORE:
GRIDFTP - The UNICORE Environment
- 18 - GLOBUS TOOLKIT:
3.0 Delivers Grid Standards

IN THIS ISSUE:

Welcome to the second issue of the GRIDSTART Technical Bulletin. This bulletin is intended to be a useful source of articles and announcements relevant to Grid developers worldwide.

This issue has a number of articles from Europe and the US. We have articles describing the goals and current status of the DATATAG and GRIDLAB projects – excellent starting points if you haven't come across these projects before. We publish the announcement of the alpha version of the Globus Toolkit 3.0 – probably the most eagerly awaited announcement of the year. We also hear from Ian Foster and Tom Garritano on the establishment of the GRIDS Centre Software Suite resulting from the NSF Middleware Initiative in the US. If you're interested in the architecture of the Grid we have an article outlining the DATAGRID project architecture and also an in-depth look at the use of GridFTP within the UNICORE environment thanks to the EUROGRID project. A big thank you to everyone who has contributed to this issue!

This publication will only make a worthwhile contribution to the long and challenging road that lies ahead if it gets the support of the GRID community. That support means the submission of articles, news items and any other material you feel is of interest to the rest of us involved in making the GRID a reality. Please send your contributions to technical_bulletin@gridstart.org. We look forward to promoting your work and ideas in future issues.

GRIDSTART

Technical Bulletin

- Editorial -

The Challenges of 2003

A number of different factors are coming together to ensure that 2003 will be a key year in the development of the Grid. These factors include: the first final results of a number of development projects, releases of the first reference implementations of the Open Grid Services Architecture and accompanying services, the first demonstration, in Europe at least, of large-scale, stable Grid testbeds, and the second (or third) wave of proposals to many national and international funding bodies around the world.

In Europe, both at a European Union and national level, a number of projects will complete this year – most notably the first tranche of European Commission projects such as EuroGrid, Damien and DataGrid. This latter project is now demonstrating a useable Grid testbed for leading science, based around Globus 2, which is growing daily. This is a major achievement which should be applauded.

Many of the projects which were funded two or more years ago now face the challenge of OGSA migration. Globally, many projects that committed to Globus 2 have had no option but to stay with it during this transition. Their skills and design decisions are now being tested as they consider how best to migrate to OGSA in follow-on projects. In many cases this transition will be quite straightforward – a Grid Service is not that different from a daemon-based process for instance. But in some cases a major redesign of specific components will be required and this work will be ongoing throughout 2003 and beyond in many projects.

A number of national initiatives are now beginning to demonstrate Grid testbeds for their national user bases. Many such initiatives have found this problematic – despite early warnings from US-based projects that deploying Grids was as much a social problem as a technology challenge. We are all learning during this process and in 2003 we must begin to discuss the creation of national and international policies on a range of issues relating to the creation of Grid infrastructures worldwide.

Many new proposals for funding are now being written around the world. The OGSA platform gives an underlying framework on which to build. The challenge for 2003 for the Grid community worldwide is to understand which services are required for which domains and begin building higher-level services on top of the basic framework. Encouraging global cooperation in this regard is a key goal of the GRIDSTART project.

Given that we will not solve all of the challenges of Grid computing in 2003 - what should our goal for this year be? Perhaps a good starting point would be to concentrate on greater communication and best practise between projects both nationally and internationally. Only through such work will we be able to reach our ultimate goal of truly global Grids.





- Articles -

The GRIDS Center Software Suite: Toward a National Middleware Infrastructure

Ian Foster and Tom Garritano, University of Chicago and Argonne National Laboratory, January 2003, foster@mcs.anl.gov, garritano@mcs.anl.gov

Summary

A new generation of collaborative middleware is tackling great challenges of science and engineering research, from particle colliders generating petabytes of high-energy physics data to seismological studies that combine the latest computer modelling with remote access to massive “shake tables” for simulating earthquakes. Just as the World Wide Web was initially the sole province of researchers but soon migrated to industry, many observers predict that Grids are on the verge of moving beyond science and engineering to have widespread influence on business and mainstream computing users. This vision is within reach, thanks to the increasing affordability and speed of desktop computers and commodity networks that can be aggregated to deliver supercomputer-level performance.

The Grid Research, Integration, Deployment and Support Center (GRIDS, <http://www.grids-center.org>) was formed as part of the National Science Foundation Middleware Initiative (NMI), with the goal of creating a stable middleware infrastructure to permit seamless resource sharing across virtual organizations.

Introduction

NMI and GRIDS grew out of workshops and white papers that identified the need for production-quality software based on open-source and open-standard approaches. The GRIDS Center Software Suite was first issued in mid-2002 via NMI Release One (NMI-R1), which included the Globus Toolkit™, Condor-G, the Network Weather Service and KX.509. NMI-R2 in October 2002 added GSI-OpenSSH, GPT (Grid Packaging Tools), and GridConfig. These components were chosen for their combined benefits -- they work together for maximum utility that is greater than the sum of its parts. (See <http://www.nsf-middleware.org/NMIR2>.)

Future GRIDS Center releases will follow regularly in April and October of each year. NSF funding of NMI extends to September 2004, with the strong possibility of added years. So users can rely upon this software distribution for timeliness, quality and a high degree of interoperability to make the most of on-line resources. GRIDS project organizers hope it will become the preferred source for major e-science projects such as GriPhyN, NEESgrid, TeraGrid, and others.

As part of NMI, GRIDS develops and seeks standard components and mechanisms for:

- Authentication, authorization, policy
- Resource discovery and directory
- Remote access to computers, data, instruments

GRIDS also promotes integration of these components with end-user tools (conferencing, data analysis, data sharing, distributed computing, etc.), with campus infrastructures, and with commercial technologies. The center's goals are to help define, develop, deploy, and support integrated software supporting 21st Century science and engineering applications.

GRIDS Center Software Suite

At the project's core is the GRIDS Center Software Suite, a collection of open-source software aimed at the national research, education, and scientific communities. For NMI-R2, GRIDS software supports Red Hat Linux 7.2 on IA32 or IA64 and Solaris 8.0 on 32-bit Sparc, with other platforms to be added in subsequent releases.

This suite includes the Globus Toolkit™ from the





Globus Project™, which is led by Argonne National Laboratory, the University of Southern California's Information Sciences Institute, and the University of Chicago. This software is the foundation on which users may build Grids and Grid-based applications. It has three key functions: Resource Management to allocate shared computers, storage, sensors, instruments, networks and software; Information Services to characterize these shared resources; and Data Management to let users access and organize information generated by the resources.

Condor-G is from the University of Wisconsin at Madison. It manages both a queue of jobs and a set of resources from one or more sites where those jobs can execute. Condor-G lets the user submit many jobs at once and monitor them with a convenient interface. It notifies the users when jobs complete or fail, and it maintains Globus Toolkit credentials that may expire while a job is running. Condor-G is also fault-tolerant -- if a machine crashes, the user can still perform all of these functions after the system reboots.

The Network Weather Service (NWS) is from the University of California Santa Barbara. It is a distributed, generalized system for producing short-term performance forecasts based on historical performance measurements. NWS has four functions:

- A Name Server directory used to bind process and data names with low-level contact information (such as TCP/IP port numbers or address pairs)
- A Memory Server for persistent storage of data
- A Sensor to gather performance measurements from a specified resource
- A Forecaster to predict the deliverable performance of a resource during a specified time

Each component runs in the background, listening on a dedicated port, which the user may specify if desired. The user would install one Name Server, one or more Memory Servers (which may reside on different machines), and a Sensor running on each machine whose resources would be monitored. These functions complement the Globus Toolkit's and Condor-G's functions of defining and managing an array of Grid resources.

The GRIDS Center software suite includes KX.509, a client tool developed at the University of Michigan under the auspices of a partner NMI team, EDIT (Enterprise and Desktop Integration Technologies). It lets sites easily convert Kerberos certificates to the X.509 format used by the Grid. Netscape and Internet Explorer will accept these credentials for secure https web activity.

GSI-OpenSSH is a modified version of OpenSSH that adds support for Grid Security Infrastructure (GSI) authentication. Developed by the National Center for Supercomputing Applications (NCSA), it can be used to log into remote systems and transfer files between systems without entering a password. Instead, all operations can be authenticated using a valid GSI credential. GSI-OpenSSH can forward GSI credentials to the remote system on login, so the user is not required to manually create a new GSI proxy credential on that system.

NCSA's Grid Packaging Tools (GPT), provide a straightforward way to define complex dependency and compatibility relationships between packages. Installation of the software is simplified due to bundling with GPT, which lets users choose whether to install components collectively or individually. Finally, GridConfig from the San Diego Supercomputer Center (SDSC) is used to configure and finetune Grid technologies. It provides an easy way to generate and regenerate configuration files in native formats, and to ensure consistency within and among applications.

Conclusion

GRIDS Center Software Suite components were chosen by the NMI leadership for their collective value in creating and managing computational Grids that facilitate use of powerful on-line resources. The hoped-for result will be a national infrastructure that can be used by application communities to explore full-scale, meaningful Grid applications.

See http://grids-center.org/grids_primer.asp for application examples, with links to project web sites.

For more general information on Grid computing, see also -





“The Anatomy of the Grid: Enabling Scalable Virtual Organizations,” I. Foster, C. Kesselman, S. Tuecke. *International J. Supercomputer Applications*, 15(3), 2001, at <http://www.globus.org/research/papers/anatomy.pdf> and “The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration,” I. Foster, C. Kesselman, J. Nick, S. Tuecke; January, 2002, at <http://www.globus.org/research/papers/ogsa.pdf>.

In addition to support from NSF, the GRIDS software developers wish to acknowledge contributions from the U.S. Department of Energy, the Defense

Advanced Research Projects Agency and NASA.

Ian Foster (foster@mcs.anl.gov) is a professor of computer science at the University of Chicago and associate director of the Mathematics and Computer Science division at Argonne National Laboratory. Tom Garritano (garritano@mcs.anl.gov) is project manager for the GRIDS Center at U of C and ANL.

DataTAG: Middleware Interoperability Between European & U.S. GRIDS in High Energy Physics

Rosy Mondardini, Cristina Vistoli, J.P. Martin-Flatin and Olivier Martin,
20 January 2003, datatag-office@cern.ch



Summary

The objective of the DataTAG project is to create a large-scale intercontinental testbed for Grids in order to investigate i) issues related to advanced networking technologies and ii) Grid middleware interoperability between European and U.S. Grid domains. The use of computing resources and access to data distributed across the U.S. and Europe both require the integration (or at least the coexistence of) Grid services developed and implemented on both sides of the Atlantic. This article focuses on the middleware interoperability efforts, where the goal is to enable a selected set of applications in High Energy Physics to run across the transatlantic testbed.

Grid interoperability issues had never been investigated to such an extent in the past, which makes the DataTAG work both crucial and innovative. To extend the capabilities of national Grids and enable worldwide development programs, it is necessary to establish relationships with the projects developing Grid middleware and define a general strategy and common requirements. This program is done in collaboration with the EU-funded DataGrid project, several national Grid projects in Europe, and three major Grid projects in the U.S.: GriPhyN, PPDG and iVDGL.

Introduction

Several major international Grid development projects are underway in the European Union and in the U.S., with the common goal of providing transparent access to the massively distributed computing infrastructure. Some of them collaborate on the crucial issue of Grid interoperability, to extend the capabilities of each Grid domain and enable worldwide developments and use.

DataTAG is the largest of these projects in High Energy Physics. It cooperates with the EU-funded DataGrid project, several national Grid projects in Europe, and three U.S. Grid projects: GriPhyN,

PPDG and iVDGL (iVDGL itself is a collaboration of U.S. projects in High Energy Physics). The objective is to create a large-scale transatlantic Grid testbed that enables researchers to study two issues:

- high performance inter-Grid networking, which includes advanced network protocol engineering, sustained and reliable high performance data replication, end-to-end advanced network services, and novel monitoring techniques;
- interoperability between the Grid middleware layers such as information and security services.





With respect to Grid middleware interoperability, the main goal of the DataTAG project is to enable a selected set of applications to run across the transatlantic testbed. These applications belong to several experiments in High Energy Physics: Alice, Atlas, BaBar, CDF, CMS, D0 and Virgo/Ligo.

High Energy Physics is a typical example of Virtual Organization (VO). Thousands of physicists from hundreds of research institutes, laboratories and universities worldwide collaborate to design, create and operate detectors located at CERN in Europe or at U.S. laboratories such as Fermi Lab (FNAL) and SLAC. Moreover, during the data analysis phase, they need to pool their computing, storage and networking resources to analyze up to several petabytes of data (1 petabyte = 1000 terabytes = 1,000,000 gigabytes), thereby contributing to understand and identify specific interoperability issues.

Interoperability Approach -

One of the core DataTAG activities concentrates on the interoperability of middleware software developed by three major Grid projects in High Energy Physics: DataGrid in Europe, and GriPhyN and PPDG in the U.S. GriPhyN and PPDG share the same middleware (VDT v1), which is different from DataGrid Release 1 but relies on the same software (Globus and Condor) for implementing basic services.

The three projects have similar requirements imposed on them by the High Energy Physics experiments. As a result, they set similar requirements on their respective middleware packages. When the experiment collaboration accesses storage or computing resources in a Grid environment, it requires:

- resource discovery and security mechanisms;
- reliable data transfers;
- scheduling of computational jobs;
- scheduling of data movement operations;
- monitoring of the entire system for detecting faults;
- fault correction.

The above mechanisms and services are for the most part application independent and can therefore be developed as independent mechanisms. This

common observation led DataGrid and GriPhyN/PPDG to build software architectures where specific data-intensive Grid mechanisms are built on top of generic services that make up the basic Grid infrastructure.

This approach led to the definition of a three-tier interoperability framework. The *bottom layer* includes all (in principle unlimited) Grid resources, e.g. computers, storage systems and network devices.

The *middle layer* is the basic Grid infrastructure. It is a set of protocols and core services used for resource access, data movement, name resolution, authentication, authorization, resource discovery, resource management, etc. These protocols and services guarantee interoperability between different Grid domains. One implementation, the Globus toolkit, is the de facto standard for Grid systems and was adopted by DataGrid and GriPhyN/PPDG.

The *top layer*, built on top of the core services, includes several flavors of Grid scheduling, job submission, resource discovery and data handling. They must guarantee interoperability with the core services and their coexistence within the same Grid domain. These are the services of particular interest to the High Energy Physics experiments.

Interoperability Framework: GLUE -

The framework that embraces the Grid middleware interoperability efforts in the U.S. and Europe is called GLUE (Grid Laboratory Uniform Environment). The main goal of GLUE is to achieve "basic, minimum functional interoperability and integration" by establishing relationships between the European and U.S. partners, the middleware projects and the High Energy Physics experiments.

The management and development staff are provided by the DataTAG and iVDGL projects. Several joint groups with European and U.S. staff have been created on an ad hoc basis and are working in different areas.

Most of the GLUE activities are in the area of core services. The initial work focused on the Information Service Infrastructure and on the first solutions adopted to manage the authentication and authorization of Grid users. This effort has already





produced results, e.g. the definition of a common schema (information service) for computing and storage resources. The first inter-Grid testbed between Europe and the U.S. is already in place.

The interoperability between European and U.S. Grid domains was demonstrated for the first time in public during the Information Society and Technology (IST2002) conference, which took place on 4-6 November 2002 in Copenhagen, Denmark. It was performed on the GLUE testbed, which included 10 sites in Europe (Germany, Italy, Portugal, Spain, Switzerland and UK) and eight sites in the U.S. The same demonstration was given during the Super Computing (SC2002) conference held on 16-22 November 2002 in Baltimore, MD, USA.

Current and future work will have to demonstrate the ability to i) move data from storage services operated by one site to another, and ii) perform some level of cross-organization and community-based authorization. User jobs submitted from one of the resource brokers will have to be sent to the computing element that is the most suitable across the entire combined Grid service. Logging, bookkeeping, accounting and information retrieval should also work across Grid domains in the future.

Once the GLUE collaboration has established the minimum requirements for achieving the interoperability of Grid middleware, future software designed by the involved projects should commit to maintaining the achieved interoperability level.

Conclusions

The first and foremost requirement for Grid interoperability is the need to access Grid resources, wherever they are, by using common protocols, common security, common authentication and authorization basic mechanisms, and common information describing these resources. Achieving this across European and U.S. Grid domains is a complex issue. It involves not only the study of the different Grid services provided by middleware, but also the setup of strong relationships between partners on both sides of the Atlantic and collaborations with the High Energy Physics experiments providing the applications.

DataTAG and its European and U.S. collaborators have already met key objectives to this end. More activities are ongoing to set up the organization and to define the interoperability approach, after having analyzed the available software and the different European and U.S. end-users' needs.

The definition of the GLUE framework was an important step, which enabled the specification and implementation of a preliminary common Grid infrastructure. The evaluation of the core services of the European and U.S. Grids was performed in collaboration with the High Energy Physics experiments. Their prompt feedback and their rather urgent need to integrate their applications with Grid middleware made this work fast paced and very concrete.

Two Grid middleware packages have been considered so far in the DataTAG project: DataGrid release 1 and VDT v1. Both DataGrid and iVDGL are still ongoing, and a close collaboration with these projects is needed to ensure that interoperability improves as new versions of these software packages are released.





The European DataGrid Middleware Architecture

DataGrid Project Office, January 2002, rosy.mondardini@cern.ch

Summary

The aim of the EU DataGrid (EDG) project is to support advanced scientific research, which requires intensive computation and analysis of shared large-scale datasets across widely distributed scientific communities. Such requirements are emerging in many scientific disciplines, including High Energy Physics, Biology, and Earth Sciences.



This article describes the architecture of the middleware software and the elements of its design. Based on the one proposed by Ian Foster and Carl Kesselman [1], the EDG architecture is “multi-layered”, with 16 implemented services.

The software is open source and based on the Linux Red Hat platform, and supports a range of standard machine profiles (Computing Element, Storage Element, User Interface, Resource Broker, Worker Node and Network Monitoring Node). To simplify integration and interoperability with other Grid projects, de-facto industry standards, notably the Globus toolkit, are integrated in the architecture and standard protocols and formats are used whenever possible.

A set of common shared services are available to all certified users with valid X.509 PKI certificates issued by a Certificate Authority trusted by EDG.

Introduction

The design of the multi-layered EDG architecture [2] follows the requirements of scientific disciplines such as High Energy Physics, Earth Observation and Bio-Informatics. These research communities need to access and process large amounts of data over massively distributed computing infrastructure. In particular, they will need:

- access to very large volumes of archived data products;
- storage and retrieval of processed data of different types from a variety of diverse distributed sources;
- access to, and support for, a large number of shared, heterogeneous computational resources, widely distributed across different organizations and centers throughout Europe and the rest of the world;
- use of high-throughput data transmission over wide areas networks between such computation centers.

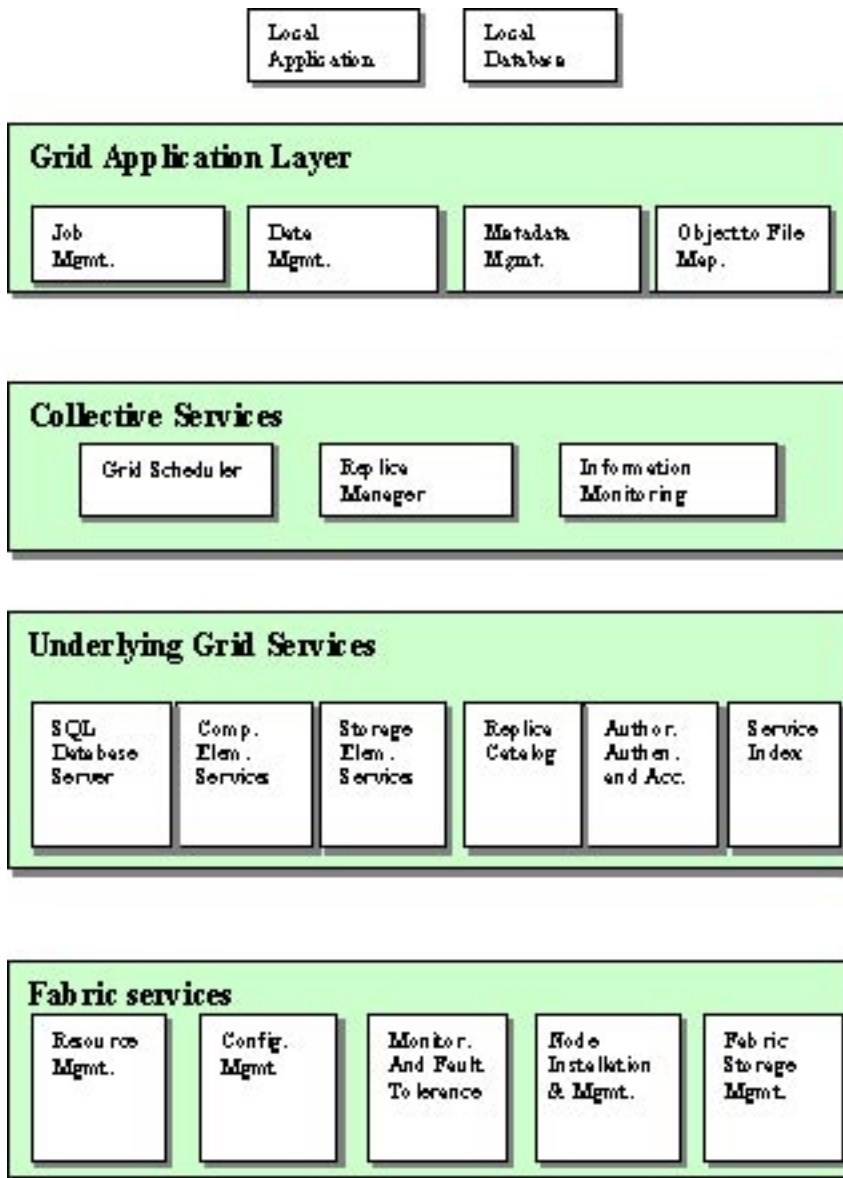
The EDG Grid architecture is shown in Figure 1 [3], where, from bottom to top, the layers are: Fabric

Services, underlying Grid Services, Collective Services, Grid Application and, at the top, local applications and databases, which represent the end user machine.

Sixteen services have been implemented by the middleware developers, based both on new modules produced within DataGrid and on the use of the Globus 2 toolkit (www.globus.org) for basic Grid infrastructure services. Globus provides the essential services for secure and authenticated use of both operating system and network connections to safely transfer files and data and allow interoperation of distributed services. The middleware also relies on software from the Condor project (<http://www.cs.wisc.edu/condor/>) and on general open source software such as OpenLDAP (<http://www.openldap.org>).

The DataGrid architecture provides a secure environment based on a Public Key Infrastructure. Each user and each service owns a credential, consisting of a pair of keys, one private and one public. Identity and keys are included in a X.509 certificate signed by a trusted Certification Authority (CA).





- **User Interface (UI):** is the access point for the Grid users, which allows them to submit jobs and retrieve their output. A job is defined using the JDL language, which specifies the characteristics, requirements and preferences of a job.
- **Resource Broker (RB):** its main task is to find the Computing Element that best matches the requirements and preferences of a submitted job, considering also the current distribution of load on the Grid. Once a suitable Computing Element is found, the job is passed to the Job Submission Service for the actual submission. Additionally, the Resource Broker allows cancelling a job and retrieving the output once a job has completed.
- **Job Submission System (JSS):** is responsible for the actual job management operations (submission, cancellation and monitoring) once the Resource Broker has chosen a suitable Computing Element for running the job.
- **Information Index (II):** is an index that collects information from the information servers running on the various Grid resources. It is used by the RB to perform the job-resources matchmaking.
- **Logging and Bookkeeping (LB):** stores a variety of information about the status and history of submitted jobs.

The middleware development is divided into six functional areas: Workload Management, Data Management, Grid Monitoring and Information Systems, Fabric Management, Mass Data Storage, and Network Monitoring.

Workload Management System (WMS) -

The WMS provides to the Grid users a set of tools to submit their jobs, have them executed on the distributed Computing Elements, get information about their status and retrieve their output. It also allows them to access Grid resources in an optimal way, optimising CPU usage, reducing file transfer time and cost, and balancing access to resources between users.

The WMS is currently composed of the following parts:

Data Management System (DMS) -

The DMS provides secure access to massive amounts of data in a universal global name space, moves and replicates data at high speed from one geographical site to another, and manages synchronisation of distributed replicas of files or databases. The emphasis is on automation, ease of use, scalability, uniformity, transparency and heterogeneity. The main components of the DMS, currently provided or in development, are:

- **Replica Manager:** manages the creation of file replicas by copying from one Storage Element to another and optimising the use of network bandwidth. It interfaces with the Replica Catalogue to allow tracking of the locations of files.





- *Replica Catalogue*: stores information about files held on any given Storage Element and provides a mapping of logical file names to one or more physical file names.
- *GDMP*: automatically mirrors file replicas from one Storage Element to a set of other subscribed sites.
- *Spitfire*: provides a Grid-enabled interface for access to relational databases.

Grid Monitoring and Information Systems -

The goal of this system is to monitor the performance of the Grid and provide easy access to current and archived information about: resources (Computing Elements, Storage Elements and the Network), job status (e.g. as implemented by the WMS Logging and Bookkeeping service), and user applications running on the Grid.

One of its main components is R-GMA, a relational GMA (Grid Monitoring Architecture) implementation based on that of the Global Grid Forum (www.ggf.org). The GMA consists of three components: consumers, producers and registry. Producers register themselves with the Registry and describe the type and structure of information they want to make available to the Grid. Consumers can query the registry to find out what type of information is available and locate producers that provide such information. R-GMA makes information from producers available to consumers as relations (tables).

EDG Fabric Installation and Job Management Tools -

Fabric Services are the low-level layer that the Grid middleware actually runs on. The services in this bottom fabric layer are not accessible by the Grid, but can be understood as delivering the building blocks for the upper level Grid layers. A typical computer fabric consists of clusters of computing nodes, where the user jobs are run, and a number of infrastructure elements. The EDG collaboration has developed a complete set of tools for the management of fabrics, in order to make the installation and configuration of the various nodes automatic and easier for the site managers, and for the control of jobs on the Worker Nodes.

The Storage Element -

The role of the SE software system is primarily to sit between the client and the Mass Storage System (MSS), providing the users three main functions:

- support existing protocols for data transfer and be extensible to new protocols that may appear;
- provide a range of functions for control, such as reservation, pinning, deletion, and transfer time estimation;
- act as information providers to the DataGrid Information Service, providing metadata about the SE, the underlying Mass Storage System, and the files therein.

Conclusions

The EDG project has now completed its second year, and very encouraging results have already been achieved in terms of its major goals.

A production quality testbed has been set up and implemented at a number of EDG sites, while a separate development testbed addresses the need for rapid testing and prototyping of the EDG middleware.

All three application groups (HEP, Bio-informatics, Earth Observation) have been given access to the testbed to test their applications and explore to what extent they can make use of the grid technologies. Their feedback has been used to improve the software and define the priorities for the development work.

A number of important services are still in development in the six functional areas of the middleware activity. While the WMS will implement billing and accounting, advance reservation of resources, job partitioning and checkpointing, the data management area will introduce APIs for Replica Selection through a Replica Optimiser and a new distributed and fault tolerant Replica Location Service jointly developed with Globus. More efficient systems and tools will be applied in the Information Systems and Fabric Management. Interfaces to Storage Elements will continue to be developed and network monitoring information will be used to influence the decisions of the Resource Broker. Due to the research nature of the applications, the requirements may vary over the lifetime of the





project. Consequently, the architecture is designed with flexibility in mind and will evolve to take into account further requirements and the experience gained from deploying implementations to support users applications.

References

[1] "The Anatomy of the Grid", I. Foster, C. Kesselman, et al.
Technical Report, Global Grid Forum, 2001
<http://www.globus.org/research/papers/anatomy.pdf>

[2] "The EDG Architecture"

<https://edms.cern.ch/document/333671/3.0>

[3] "European DataGrid Project: Experiences of deploying a large scale Testbed for e-Science applications"

<http://hep-proj-grid-tutorials.web.cern.ch/hep-proj-grid-tutorials/presentations/Perf2002Paper.doc>

Note: all official EDG documents are available on the web at: <http://www.edg.org>

GridLab: Grid Application Toolkit and Testbed

Summary

European and US research institutions, along with leading computer vendors Sun and Compaq, have joined together to develop a Grid Application Toolkit (GAT).

The GridLab project is an international research project funded by the European Commission under the Fifth Framework Programme (IST-2001-32133). The main objective of GridLab is to bring both new and legacy applications to the emerging Grid world. It will provide new, innovative tools, services and methodologies

for efficient development of grid enabled applications. The GridLab software will enable researchers and developers worldwide to develop new generations of engineering and scientific applications that can harness the power of grid computing.



Key objectives

Design and develop a Grid Application Toolkit (GAT), to provide core, easy to use functionality through a carefully constructed set of generic APIs for *both simulation codes and Grid software*. The GAT will contain independent modules for handling many different aspects of Grid programming, including simulation, performance and grid monitoring, resource brokering and selecting, performance prediction, interaction with information servers, security, notification, collaboration, data handling, remote visualization, and remote application steering.

Simultaneously enhance real applications for the Grid, implementing new dynamic simulation scenarios using the GAT. Both *Cactus* and *Triana* will be extended to integrate and exploit GAT elements, making Grid Computing easily exploitable by a wide range of applications. Our simulation driven, compute intensive applications are fundamentally different from the highly data driven applications in many other Grid projects (e.g., DataGrid, GriPhyN, EuroGrid).

Develop and test Grid infrastructure/applications on real testbeds, constructed by linking heterogeneous collections of supercomputers and other resources spanning Europe and the USA, using and extending existing testbeds. Interoperability with different testbeds will be ensured by also using production testbeds in the USA, driving international high speed network connectivity. Testing will be carried out by the project and by several large, closely related user communities, including an EU Astrophysics Network, and various multidisciplinary US funded collaborations.

Partners

The GridLab consortium includes ten European project partners: Poznan Supercomputing and Networking Center (Poland) - Project Coordinator, Albert Einstein Institute (Germany), Konrad-Zuse Zentrum (Germany), University of Wales (UK), Vrije University (Netherlands), University of Lecce (Italy), Computer and Automation Research Institute - SZTAKI (Hungary), Masaryk University





(Czech Republic), National Technical University of Athens (Greece), two commercial partners: Sun Microsystems Gridware GmbH (Germany), Compaq Computer (France) and three partners from the US: Argonne National Laboratory, University of Southern California and University of Wisconsin.



Applications

CACTUS -

The primary application framework and environment for the development and testing of the GAT is the open source Cactus Computational Toolkit (Cactus). Cactus is a modular, collaborative framework for developing parallel scientific and engineering applications which are already widely used in Grid computing. Cactus was developed over many years to solve some of the largest scale problems in computational science, and to enable large scale collaborations of different communities to work together.

<http://www.cactuscode.org>

TRIANA -

The Work-Flow Application Toolkit, based on the Triana package developed originally for gravitational wave (GW) data analysis, will provide a second example application toolkit for GridLab. GW data analysis has a large community of users that will benefit significantly from the developments of GAT.

<http://www.triana.co.uk>

Other Applications -

GridLab develops a generic GAT which can be used by all the other scientific and engineering applications.

Using the GAT you can make your applications grid enabled in a very fast way.

Work Packages -

WP1 Grid Application Toolkit provides a link between Grid middleware and applications, usable by any conforming application or middleware component.

WP2 Cactus Grid Application Toolkit provides an extended GAT interface for Cactus.

WP3 Work-Flow Application Toolkit will develop Grid capabilities for a widely used data flow programming environment.

WP4 Grid Portals will be highly application driven, aimed at providing uniform, flexible and intuitive user access to Grid resources.

WP5 Testbed Management will administrate and maintain an active development testbed across roughly a dozen EU and US sites

WP6 Security will ensure the integration of all the technologies developed under other WPs, taking into account the various local security requirements and state of the art solutions.

WP7 Adaptive Application Components develops a set of components and APIs to be plugged into the Grid Application Toolkit.

WP8 Data handling and Visualization will provide Grid aware techniques for data management, analysis, and visualization, needed especially for dynamic applications.

WP9 Resource Management will develop resource management mechanisms such as estimators and resource brokers.

WP10 Information Services will extend existing Grid middleware toolkits with dynamic features needed by applications to select appropriate Grid resources.

WP11 Monitoring will develop new components that will tie in the general Grid monitoring architecture to support application steering, monitoring and automatic analysis.

WP12 Access for mobile users will develop and test Grid access technologies through a variety of mobile devices, from 2.5G and 3G wireless networks.





WP13 Information Dissemination and Exploitation will ensure the active dissemination of the project results through a variety of channels.

WP14 Project Management will ensure the project's professional management, including careful orchestration and monitoring of work across all groups.

Conclusions

GridLab will dramatically change the way we use computing today. GridLab will develop core capabilities for simulation and visualization codes to be self aware of the changing Grid environment, and to be able to fully exploit dynamic resources for fundamentally new and innovative application scenarios.

The applications themselves will possess the capability to migrate from site to site during the execution, both in whole or in part, to spawn related

tasks, and to acquire additional resources as needed, according to both the changing availabilities of various resources in the grid, and the needs of the applications themselves.

The GridLab project will develop an easy-to-use, flexible, generic and modular *Grid Application Toolkit* (GAT), enabling today applications to make innovative use of global computing resources. The project is grounded by two principles, the *co-development* of infrastructure with real applications and user communities, leading to *working* scenarios, and *dynamic* use of grids, with self-aware simulations adapting to their changing environment.

For more information on the GridLab project see the website:

www.gridlab.org

GridFTP as an Alternative File Transfer Mechanism within the UNICORE Environment

Dirk Breuer, Daniel Mallmann - Forschungszentrum Jülich GmbH, January 2003,
d.breuer@fz-juelich.de, d.mallmann@fz-juelich.de



David Snelling, Sven van den Berghe, Fujitsu European Centre for Information Technology Ltd., January 2003,
snelling@fle.fujitsu.com, svdb@fle.fujitsu.com

Summary

Grid applications which transfer large amounts of data need a high-performance, secure and robust data transfer mechanism. Within the EUROGRID project the UNICORE (Uniform Interface to Computer Resources) infrastructure is extended to use such a mechanism based on GridFTP. This protocol extends the popular file transfer protocol (FTP) with new features required for Data Grid applications, such as striping and partial file access.

Keywords: **AFT GridFTP UNICORE Globus EUROGRID**

Introduction

One task of the EUROGRID project is the development of an alternative file transfer mechanism (AFT), which does not transfer data as a java byte stream via the usual UNICORE Protocol Layer (UPL). Instead, this mechanism is based on GridFTP developed by Globus and is able to transfer

files directly from one Uspace to another within a UNICORE environment.

This article describes the Globus Toolkit security, features and installing instructions of GridFTP, the UNICORE architecture, and the prototype integration of GridFTP into the UNICORE software.





2. GLOBUS TOOLKIT: Security Infrastructure and GridFTP

This section describes the security infrastructure of the Globus Toolkit and gives an overview of GridFTP.

2.1 Security Infrastructure -

The Globus Toolkit uses the Grid Security Infrastructure (GSI) for enabling secure authentication and communication over an open network (<http://www.globus.org/security/>). GSI provides a number of useful services for Grids, including mutual authentication and single sign-on. GSI is based on public key encryption, X.509 certificates, and the Secure Socket Layer (SSL) communication protocol. Extensions to these standards have been added for single sign-on and delegation. The Globus Toolkit's implementation of the GSI adheres to the Generic Security Service API (GSS-API), which is a standard API for security systems promoted by the IETF.

Delegation and Single Sign-On -

The GSI provides a delegation capability, an extension of the standard SSL protocol which reduces the number of times the user must enter the pass phrase. If a Grid computation requires that several Grid resources are used (each requiring mutual authentication), or if there is a need to have agents (local or remote) requesting services on behalf of a user, the need to re-enter the user's pass phrase can be avoided by creating a proxy (<http://www.globus.org/security/proxy.html>)¹.

A proxy consists of a new certificate (with a new public key in it) and a new private key. The new certificate contains the owner's identity and additional information to indicate that it is a proxy. The new certificate is signed by the owner, rather than a CA. The certificate also includes a time stamp after which the proxy should no longer be accepted by others. Proxies have limited lifetimes.

Globus Specific Files and Directories -

Before using parts of the Globus Toolkit you have to store the following certificates, keys and files in the specified directories (http://www.globus.org/security/v1.1/etc_grid_security.html):

*/etc/grid-security/hostcert.pem, hostkey.pem, grid-mapfile
/etc/grid-security/certificates*

*42864e48.0, 42864e48.0.signing_policy
\$HOME/.globus
usercert.pem, userkey.pem*

To request Globus certificates you have to run `grid-cert-request` (http://www.globus.org/security/v1.1/host_cert.html).

The directory `/etc/grid-security` is a directory used to hold configuration information and private information used by the GSI libraries. It should be on local disk e.g. not NFS-mounted. It and its contents should not be writable by any untrusted users (generally this means only by root). Except for the private key of the host (`hostkey.pem`), all the contents may be world-readable.

`hostcert.pem` contains the X.509 certificate for the host used for doing mutual authentication with connecting clients. `hostkey.pem` contains the RSA private key for the host. `hostcert.pem` however is public information (it is passed to every connecting client) and is typically world-readable. The `grid-mapfile` contains authorization information and information to map grid identities (e.g. X.509 subject names) to local system identities (e.g. user names). This file needs only be readable by processes that offer GSI-based services, but there is no known vulnerability caused by it being world-readable. The `/etc/grid-security/certificates` directory contains the X.509 certificates of all CAs trusted by this system. For a user to authenticate with GSI to any GSI-based service on this system, the certificate of the CA that signed the user's certificate must be in this directory. All the contents of this directory is used by both GSI-based clients and GSI-based services and should be world-readable. CA certificates are stored in files whose names are based on a hash of the CA identity for quick location. These filenames typically look like `42864e48.0`. Certificate Revocation Lists (CRLs) may also be stored in this directory. The CRLs follow the CA certificate hash naming convention, with an extra „r“, for example: `42864e48.r0`.

The `$HOME/.globus` directory contains the user certificate `usercert.pem` and the user key `userkey.pem`.

¹-To create a proxy with the default expiration of 12 hours, run the `grid-proxy-init` program without any parameters.





Managing the Grid-Mapfile -

Users can be authorized to use GSI-enabled services by entering them into the GSI access control list, the so-called grid-mapfile (<http://www.globus.org/security/v1.1/grid-mapfile.html>). The purpose of this file is to map a GSI Credential to a local user's login name. The GSI administrator at the site can map the holder of any GSI credential to any local user name. It is up to the GSI administrator to verify that the GSI identity² is owned by and matches the local username.

The grid-mapfile file is a plain text file, containing a quoted GSI Credential Name (the subject of an X509 certificate) and a unquoted local user name. Each subject name in the grid-mapfile must be listed only once. However, multiple identities may map to a shared local name. It is up to the GSI administrator to ensure that the grid-mapfile entries do not violate any site security policies.

An example of a grid-mapfile entry is:

```
„/C=DE/O=Forschungszentrum Juelich/O=ZAM/
CN=User Name“ username
```

2.2 GridFTP

GridFTP is a high-performance, secure, reliable data transfer protocol optimized for high-bandwidth wide-area networks (<http://www.globus.org/datagrid/gridftp.html>). The GridFTP protocol is based on FTP, the highly-popular Internet file transfer protocol. GLOBUS has selected a set of protocol features and extensions defined already in IETF RFCs and added a few additional features to meet requirements from current data grid projects.

Features of GridFTP

GridFTP provides the following protocol features³:

- GSI security on control and data channels
- Multiple data channels for parallel transfers
- Partial file transfers
- Third-party (direct server-to-server) transfers
- Restart (stream mode and extended block mode)
- Ability to manually set TCP buffer size

The C implementation of the GridFTP protocol takes the form of two APIs and corresponding libraries: `globus_ftp_control` and `globus_ftp_client`. These libraries use high-performance I/O and security

services provided by the Globus Toolkit. In addition to the software libraries, Globus has also implemented an API/library (`globus_gass_copy`) and a command-line tool (`globus-url-copy`) that integrates GridFTP, HTTP, and local file I/O to enable secure transfers using any combination of these protocols. Finally, they have adapted a popular FTP server package (Washington University's `wu-ftpd`) to support a majority of the GridFTP protocol features.

In contrast to the C implementation of GridFTP the current Java Commodity Grid Kit (CoG 0.9.13; <http://www.globus.org/cog/java/>) supports only basic GridFTP features (data channel authentication and protection). **GridFTP client library version 0.9.14** supports Extended Block Mode, parallel and striped connections, restart and performance markers, and data channel authentication. Both client-server and third party transfers are supported.

Setting up GridFTP -

Installation and set up of GridFTP is described in the Globus 2.2 Installation Instructions (<http://www.globus.org/gt2.2/install.html>).

To summarize, the installation will follow these steps:

1. Create a user named „globus“ (optional, but recommended).
2. Create a location to install GridFTP .
3. Download and install the Grid Packaging Technology (GPT).
4. Download source or binary data management client and server bundles.
5. Use GPT to install the bundles.
6. Configure the installation.
7. Set up GSI.

GridFTP Server -

A GridFTP server is based on the standard `wu-ftpd` with additional functionalities: parallel streams, striping, advanced restart, `gsi`, servers side processing, data channel caching, performance markers, structured file info. This server will normally run as `in.ftpd`. Other capabilities (that were already in `wu-ftp`) include: 3rd party transfers and manual control of TCP buffer size for performance reasons. After you have created a proxy, you can start a GridFTP server e.g. through:

² - Note that a real person is involved.

³ - Striping, auto negotiation of TCP buffer size and pipelining aren't currently implemented.





```
$GLOBUS_LOCATION/sbin/in.ftpd -s -p 5678 -d -l -L &
```

The `-s` tells `in.ftpd` to run in foreground in server mode as a standalone daemon, and the `-p` specifies a port. If that port is already in use, you have to try another. The `-d` (or `-v`) options tells the daemon to run in debug mode, sending log debug info to `/var/log/messages`. Because of specifying the `-l` and `-L` options, `in.ftpd` writes log session and log commands to `/var/log/messages`.

To set up a Grid-FTP server, add an entry to `/etc/services` as root:

```
gsiftp 2811/tcp
```

Depending on whether your host is running `inetd` or `xinetd`, you will need to modify its configuration file.

GridFTP Client -

`globus-url-copy` is a GridFTP-client command-line tool that integrates GridFTP, HTTP, and local file I/O to enable secure transfers using any combination of these protocols. It copies the file specified by `sourceURL` to a location specified by `destURL`, using the GASS (Global Access to Secondary Storage) transfer API. All protocols supported by GASS (local file, `http`, `https`, ...) are provided. A restriction of this tool is, that it can only transfer one file. Transfers of directories and wildcards are not supported.

The basic syntax for `globus-url-copy` is:

```
globus-url-copy [options] sourceURL destURL
```

(The URL prefix for GridFTP is `gsiftp://`. It can also handle `file://`, `http://`, and `https://`.)

The following table shows some important options:

<code>-vb</code>	During the transfer, display the number of bytes transferred and the transfer rate per second (verbose)
<code>-b</code>	Binary. Do not apply any conversion to the files (default)
<code>-dbg</code>	Debug ftp connections. Prints control channel communication to <code>stderr</code>
<code>-s <subject></code>	Use this subject to match with both the source and dest servers
<code>-tcp-bs <size></code>	specify the size (in bytes) of the buffer to be used by the underlying ftp data channels
<code>-bs <block size></code>	specify the size (in bytes) of the buffer to be used by the underlying transfer methods
<code>-p <parallel></code>	specify the number of streams to be used in the ftp transfer
<code>-notpt</code>	turn third-party transfers off (on by default)
<code>-nodcau</code>	turn off data channel authentication for ftp transfers

Example of a parallel 3rd party transfer:

```
globus-url-copy -dbg -vb -p 32
```

```
gsiftp://zam288.zam.kfa-juelich.de:2811/tmp/file1
gsiftp://zam289.zam.kfa-juelich.de:2811/tmp/file1
```

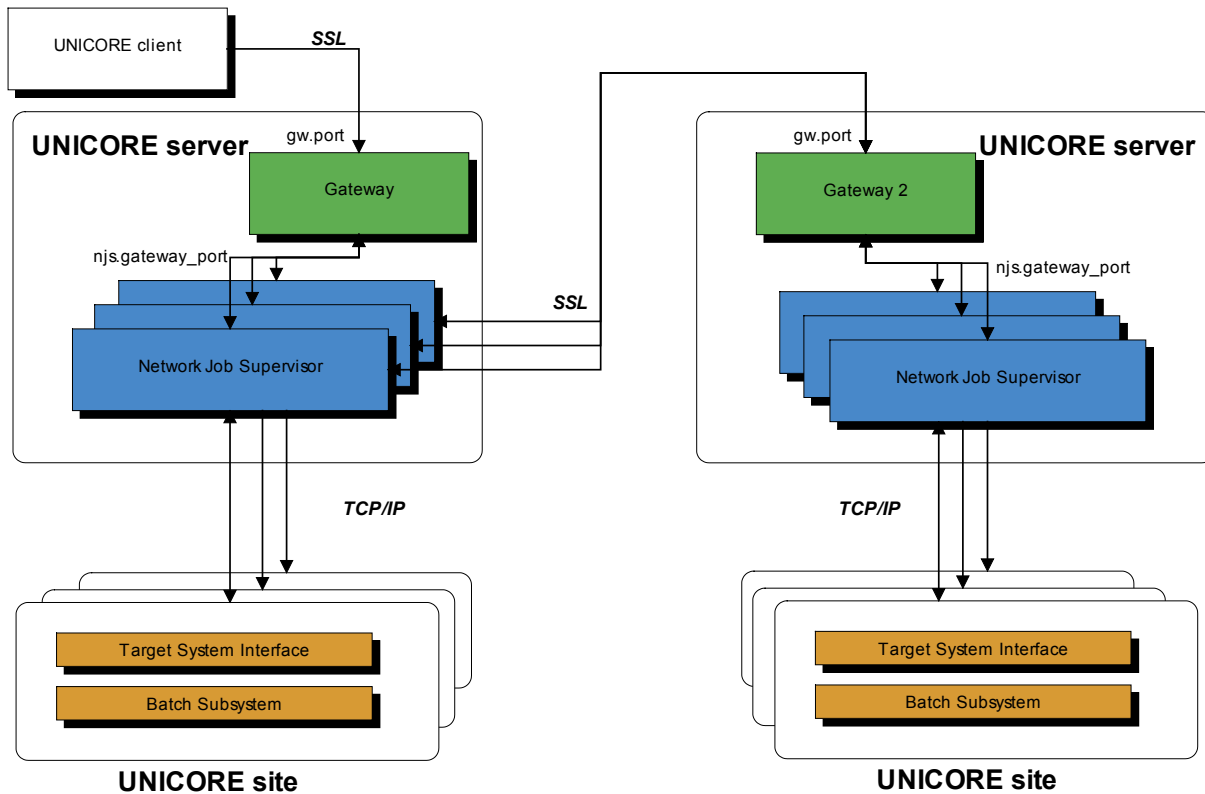
3 UNICORE Architecture -

The UNICORE architecture consists of three tiers⁴: user, server and target system tier (cp. the following figure). The user tier consists of the graphical user interface. It offers the functions to prepare and control UNICORE jobs and to set up and maintain the user's security environment. The security architecture is based on the Secure Socket Layer (SSL) protocol. The UNICORE client generates an Abstract Job Object (AJO) from the user input which is sent to the Gateway. The AJO is realized as a Java class library.

The Gateway as the first part of the UNICORE Server tier ensures the user authentication, secure communication between client and server, and provides the client with information about the resources available at the site. It also talks to the Network Job Supervisor (NJS) servers (second part of the UNICORE Server tier) at the site so that jobs and data, status requests and control commands for further processing are sent and data is received to make it available to the user. Each NJS controls one Virtual Site (Vsite) which is a single system or a cluster of systems sharing the same userids and file space. Each UNICORE job is assigned to a UNICORE Job directory (Uspace) which is the UNIX working space for the job. It is a temporary directory existing only during the lifetime of the job at the site.

4 - Mathilde Romberg, The UNICORE Grid infrastructure, Scientific Programming 10 (2002), 149-157





4 UNICORE GridFTP Integration

The following described AFT mechanism has been implemented and tested by Fujitsu Labs Europe and Research Centre Jülich. This mechanism uses an extended NJS to handle file transfers other than the byte stream that follows the consignment of an AJO. The current implementation is based on the following additional components:

1. Extended Gateway (gateway_3.6.2) that is able to handle AFT requests.
2. Extended NJS (NJS_3.6.4build3) that offers an AFT interface to allow the use of alternative (non-UPL) ways of transferring files between Vsites (provided by Fujitsu Labs Europe); this component contains a sample implementation AFTImpl.java using the remote copy command (provided by Fujitsu Labs Europe and tested by Research Centre Jülich).
3. Implementation based on 2. using the globus-url-copy command to invoke the GridFTP client (provided by Research Centre Jülich); the classes are put into AFTGridFTP.jar which extends the NJS classes.
4. GridFTP client and server

The NJS has been changed in the way that GridFTP

is used as an alternative file transfer after a job has been created, which contains a file transfer task in the UNICORE client.

Using this mechanism one can transfer a file directly from one Uspace to another one between different Vsites.

Prerequisites

For this implementation the following prerequisites must be satisfied:

- Setting up GridFTP clients and servers and GSI (as root) on the machines, where the TSIs are running;
- Globus user and host certificates, keys and grid-mapfiles are stored in the Globus-specific directories. This means, the GridFTP client needs usercert.pem and userkey.pem in the \$HOME/.globus directory. The GridFTP server needs hostcert.pem, hostkey.pem and grid-mapfile in the /etc/grid-security directory.

The use of EUROGRID certificates with GLOBUS requires two actions:

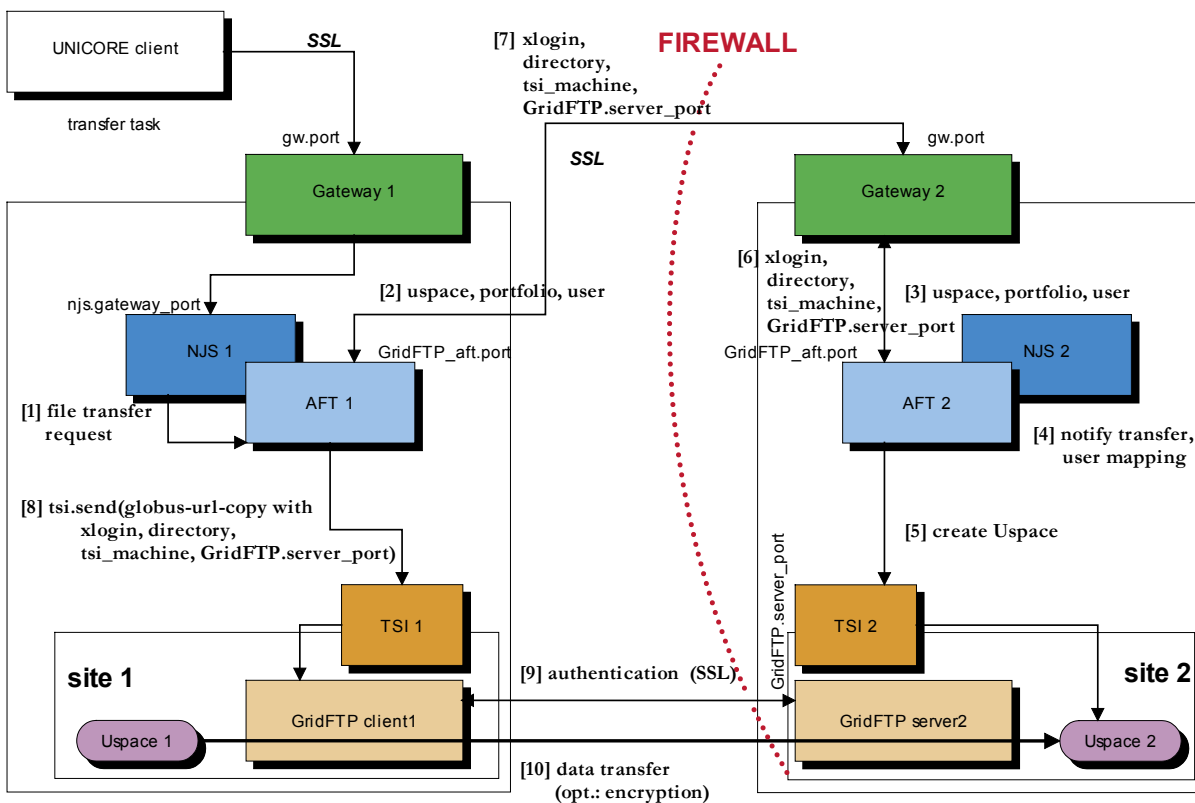


1. Setting up the GLOBUS security environment to accept EUROGRID certificates and
 2. Creation of a temporary GLOBUS proxy certificate from the EUROGRID certificate with the help of certain openssl commands.
- Before a file can be transferred via GridFTP, proxy certificates must be generated at the source Vsite by hand with grid-proxy-init on the basis of the user certificate and key.
 - GridFTP server must be started as in.ftpd.

Location where to find the Globus installation and client options to use

AFT Control and Data Flow -

The following figure describes the control and data flow between the involved components if a file transfer task is invoked in the UNICORE client. The AFT does not transfer the files via the usual UNICORE Protocol Layer (UPL). Instead, the alternative file transfer mechanism described in this article is used to transfer files directly from Uspace1



- An additional entry has to be specified in the connections file of the Gateway to notify transfer requests from other AFTs.
- The NJS configuration file njs.properties needs the following entries:
 - o Class name that implement the AFT interface of the NJS
 - o Port on which the AFT component of the NJS listens to for connections from other peers (via gateway)
 - o Name of the machine the local TSI is running on and the port on which the GridFTP server is listening on for client connections
 - o List of Vsites supporting this AFT based on GridFTP

to a remote Uspace2 in an UNICORE environment. The following main steps are performed:

1. NJS1 requests a file transfer from AFT1.
2. AFT1 sends information about uspace, portfolio and user to the remote Gateway2.
3. This information is forwarded to AFT2.
4. The transfer is notified and authorised for the user; user mapping.
5. AFT2 instructs TSI2 to create Uspace2.
6. AFT2 returns to Gateway2 the xlogin, Uspace2 directory name, the name of TSI machine and the port of the GridFTP server.
7. This information is forwarded to AFT1.
8. AFT1 then instructs TSI1 to invoke the GridFTP





- client with the required parameters.
9. The GridFTP client establishes a connection to the remote GridFTP server.
 10. If the user is authorised, the file transfer is performed.

Conclusion and Future Directions

The described AFT mechanism was implemented and tested by Fujitsu Labs Europe and Research Centre Jülich. With the present implementation the user can transfer a file from one Uspace to another one between different Vsites using GridFTP within a UNICORE environment.

The main item to be discussed is to integrate the security mechanism of GridFTP within UNICORE. This could be done by using the proxy plugin which is developed within the GRIP project. In the future GridFTP performance within a UNICORE environment will be measured. Thus you can compare the performance of the AFT based on GridFTP with the performance of UPL. Furthermore, the different security models will be combined, so that out of the user's view the UNICORE security infrastructure is used.

GLOBAL TOOLKIT 3.0 DELIVERS GRID STANDARDS

Alpha version of popular open source Grid software points the way to a new generation of Grid services and applications

SAN DIEGO, January 13, 2003 Grid computing takes a major step forward today with the first implementation of emerging standards known as the Open Grid Services Architecture (OGSA).

The Globus Project™ issued its alpha release of the Globus Toolkit™ 3.0 (GT3), a set of open-source software and services whose earlier versions have transformed the way on-line resources are shared across organizations. GT3 alpha is available at <http://www.globus.org/ogsa/releases/alpha/>.



GT3's release, which coincides with the first GlobusWorld conference this week in San Diego (<http://www.globusworld.org>), is the result of the past year's effort toward defining specifications for Grid services that extend standard Web services. The OGSA-based alpha version builds on prior releases of the Globus Toolkit, which is central to hundreds of science and engineering projects on the Grid.

The Globus Project also announced that other leading Grid participants are committing to use of GT3 and OGSA. Companies include Avaki, Cray, Entropia, Hewlett-Packard, IBM, Oracle, Platform Computing, Silicon Graphics, Inc., Sun Microsystems, and Veridian. Research projects include FusionGrid, TeraGrid, the Department of Energy Science Grid, the Grid Physics Network (GriPhyN), the Network for Earthquake Engineering Simulation, the International Virtual Data Grid Laboratory, and the National Science Foundation

Middleware Initiative. A collection of quotes about GT3 by these partners is at <http://www.globus.org/about/news/prGT3quotes03-01-12.html>.

"We're enthused about this latest Globus Toolkit release," said Ian Foster, associate division director for mathematics and computer science at Argonne National Laboratory and professor of computer science at the University of Chicago. "The Grid's promise of seamlessly sharing resources across distributed organizations takes another major step towards realization with GT3 and its implementation of the OGSA standards. The array of partners that we have assembled demonstrates the power of combining open source and open standards with industrial investment." Foster is co-leader of the Globus Project with colleagues Carl Kesselman (professor of computer science at the University of Southern California and director of the USC Information Sciences Institute's Center for Grid





Technologies) and Steve Tuecke (lead architect of the Argonne Distributed Systems Laboratory). GT3 will benefit from an expanding community of developers who are closely involved in helping to develop Grid standards through the Global Grid Forum (GGF), a community-based organization with public- and private-sector contributors. For example, the UK e-Science program is leading the GGF's OGSA Database Access and Integration (DAIS) working group to build database capabilities into OGSA and GT3. Lawrence Berkeley National Laboratory is also contributing directly to the GT3 code base.

GT3 provides a major step forward in the functionality provided by the Globus Toolkit," said Kesselman. "However, of equal importance is that GT3 builds on OGSA, which in turn builds on Web services. By leveraging widely supported commodity technologies, we can lower the barrier of entry to the deployment of Grids and the development of Grid technologies. As a consequence, we expect to see the base of GT3 deployment to extend into new and important user communities."

The GT3 beta release will be in Spring 2003, with official release in Summer 2003, Tuecke emphasized. "The term 'alpha' means code that works to the best of its developers' knowledge, but is not final or bug-free," he said. "Support for the alpha release will be on a best-effort basis, because the Globus Project development team will be focused largely on improving the implementation for future releases."

Development of GT3 is sponsored primarily by the U.S. Department of Energy through its Office of Science's Mathematical, Information and Computational Sciences Division, as well as by industry partners IBM and Microsoft Research. "Grid technologies are essential to the scientific mission of the U.S. Department of Energy (DOE)," said Ed Oliver, Associate Director for the DOE Advanced Scientific Computing Research Office (ASCR). "ASCR has long supported this type of fundamental R&D both to further the study

of computer science, and to add important new capabilities to energy-related research. We are also gratified by the Grid's broad impact in commercial computing, which is a secondary but important benefit."

About the Globus Project

The Globus Project conducts research and development to create the fundamental technologies behind the "Grid," which lets people share computing power, databases, and other tools securely online across corporate, institutional, and geographic boundaries without sacrificing local autonomy. The project's open source Globus Toolkit™ includes software services and libraries for resource monitoring, discovery, and management, plus security and file management.

The toolkit is central to science and engineering projects that total nearly a half-billion dollars internationally, and it is the substrate on which leading IT companies are building significant commercial Grid products. The Globus Toolkit 2.0 received a 2002 R&D 100 Award from R&D Magazine, which further honored the toolkit as 2002's "Most Promising New Technology." The Globus Project is based at Argonne National Laboratory and the University of Southern California's Information Sciences Institute.

For more information, see <http://www.globus.org>

For more information, see <http://www.globus.org/toolkit/>

Media queries:
Tom Garritano,
E: garritano@mcs.anl.gov,
T: 630-667-4434.

