SUN BLUEPRINTS APPROVED

Global Grid Connectivity Using Globus Toolkit With Solaris™ Operating System

Guide to Integrating Globus Toolkit With Sun N1™ Grid Engine on Solaris™ 9 x86 Operating System

Chong-Wee Simon See, Ph.D., Global Sales Organization, APAC

Gabriel Ghinita, National University of Singapore

Sun BluePrints™ OnLine—April 2004

`http://www.sun.com/blueprints`

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95045 U.S.A.
650 960-1300

Part No. 817-6228-10
Revision 1.0, 5/14/04
Edition: May 2004

Global Grid Connectivity Using Globus Toolkit With Solaris™ Operating System

This article describes how to integrate grid computing with Globus Toolkit software for a site using Sun N1™ Grid Engine software (formerly Sun Grid Engine) as a local resource manager. This article provides background information and step-by-step instructions for installing, configuring, integrating, and testing Globus Toolkit software with Sun N1 Grid Engine software on an x86 architecture using the Solaris™ 9 Operating System (Solaris 9 OS).

The article contains the following topics:

Introduction

This section provides background and introductory material for grid computing, the Globus Toolkit middleware, and Sun N1 Grid Engine software.

Grid Computing and Middleware

Grids are emerging as a new infrastructure for Internet-based parallel and distributed computing. They enable the sharing, exchange, discovery, and aggregation of resources distributed across multiple administrative domains, organizations, and enterprises. To accomplish this, grids need an infrastructure that supports services such as security, uniform access, resource management, scheduling, application composition, computational economy, and accounting.

The concept of grid computing is becoming popular with the emergence of the Internet as a ubiquitous media and the wide-spread availability of powerful computers and networks as low-cost commodity components. The local area network (LAN) connected clusters of computer platforms have been employed to solve computationally intensive problems, however they alone cannot offer the computational power demanded by applications. The geographically distributed resources need to be logically coupled together to make them work as a unified resource.

Grid middleware comes into play here. The most comprehensive grid middleware software currently available is the Globus Toolkit, version 3.0.2. The Globus Toolkit software offers resource management, data management, and information services, all layered on top of one security layer, the Grid Security Infrastructure (GSI).

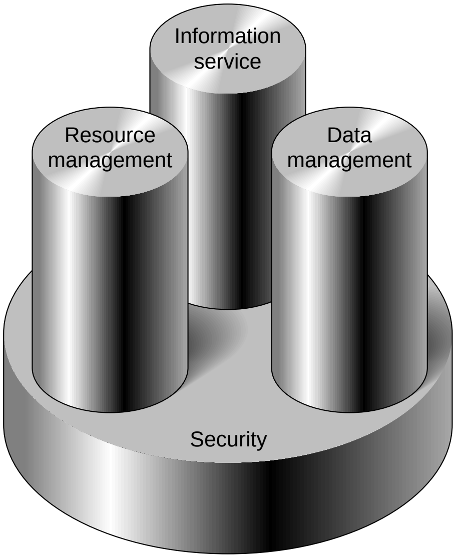

FIGURE 1 Three Key Pillars for Grid Computing on Top of the Security Infrastructure

The Globus Toolkit 3.0.2 architecture and infrastructure evolved radically from one version to another. Globus Toolkit version 3.0.2 implements the Open Grid Services Architecture (OGSA) and Open Grid Services Infrastructure (OGSI) specifications, leveraging grid computing to a new concept: grid services as a particular type of Web services. This approach creates a uniform interface to grid resources, beneficial for both grid application developers and grid users. The grid services are available through a grid services container, and the communication is based on Simple Object Access Protocol (SOAP) and HTTP protocols that are already standards of the World Wide Web. This approach allows for easy addition and integration of new services to the grid.

However, this innovative approach is still in its infancy and some existing grid applications, especially high performance computing (HPC) applications, do not have any short-term gain by moving to this new infrastructure. For this reason, the Globus Toolkit 3.0.2 distribution includes both Globus Toolkit version 2 and version 3 components. The Globus Toolkit version 2 components are not OGSA / OGSI compliant, but are easier to understand, manage, and install. This article addresses all components shipped with the Globus Toolkit 3.0.2 distribution. Installation, configuration, and testing are presented in parallel for both versions.

The three key pillars for grid computing presented in FIGURE 1 are implemented by different components in Globus Toolkit versions 2.4 and 3.*x*, but the Grid Security Infrastructure (GSI) remains mainly unchanged. GSI provides secure authentication and communication services on the grid. It is based upon Secure Socket Layer (SSL), public key infrastructure (PKI), and X.509 digital certificates. The main functions implemented by GSI are single/mutual authentication, confidential communication, authorization, and delegation.

Globus Toolkit Version 2.4

The main Globus Toolkit version 2.4 components are as follows:

- Grid Resource Allocation Manager (GRAM) is responsible for resource allocation, job submission and execution, and job status and progress management. GRAM makes use of Global Access to Secondary Service (GASS) to stage input/output (I/O) files and executables.
- Monitoring and Discovery Service (MDS), based on the Lightweight Directory Access Protocol (LDAP), provides support for collecting information about the grid and responding to queries from clients. The two MDS services are Grid Resource Information Service (GRIS) and Grid Index Information Service (GIIS). GRIS is responsible for collecting the information from information providers and registering information to GIIS. The GIIS enables the creation of hierarchical directory structures that can efficiently store and distribute information.
- GridFTP is a secure and high-performance data transfer tool; both partial and complete transfers are supported through the Globus Replica Catalog and Management features.

Globus Toolkit Version 3.*x*

Compared to Globus Toolkit 2.4, the Globus Toolkit 3.*x* counterparts are as follows:

- Master Managed Job Factory Service (MMJFS) provides job submission, execution, and management services.
- Index Services provide a way to produce and query service data; they are mainly used in discovery operations.
- Reliable File Transfer (RFT) services, or multiRFT, are part of the Data Management implementation, with GridFTP and Replica Location Service (RLS), and provides the interface for reliable file transfers on grid servers.

Note – Globus Toolkit 3.*x* implements new OGSA/OGSI components to replace some components in Globus Toolkit 2.4. For example, Index Services in Globus Toolkit 3.*x* replace the GRIS in Globus Toolkit 2.4. The service data are saved in XML instead of LDIF. However, the data can be ported from GRIS to Globus Toolkit 3.*x*. In Globus Toolkit 2.4, the RFT is provided by the command `globus-url-copy`, which is implemented as a grid service in Globus Toolkit 3.*x*.

Sun N1 Grid Engine Software

Sun N1 Grid Engine software is a distributed management product that optimizes utilization of software and hardware resources. It can increase utilization of available resources to as much as 98 percent. Sun N1 Grid Engine software is both a job manager and a job scheduler for clusters of computers. The Sun N1 Grid Engine Enterprise Edition software can harness computing power across multiple clusters (campus grids).

Sun N1 Grid Engine software enabled hosts can be master hosts, execution hosts, submission hosts, and administration hosts. These roles are not mutually exclusive; it is possible for a host to perform all four functions. A typical cluster configuration is to have one master host, running the `sge_qmaster` (manager) and `sge_schedd` (scheduler) daemons and the other hosts running `sge_execd` (execution) daemons. All Sun N1 Grid Engine software hosts are communicating through TCP/IP; for this purpose, there is a special daemon, `sge_commd`, running on each host.

Computing resources are modeled by Sun N1 Grid Engine software as job execution queues. Each queue can have specific attributes and can support multiple parallel environments. The most frequent parallel environments used are Message Passing Interface (MPI) and parallel virtual machine (PVM).

The Globus Toolkit is a grid middleware technology that enables the usage of heterogeneous resources distributed across large geographical areas. It has to cope with stringent issues such as tight security and a complex infrastructure. It is difficult for a single software program to deal with the particularities of all involved systems. On the other extreme, you would not want all computing resources to be directly connected to the grid. This could result in management challenges and increased communication overhead.

To overcome these issues, it is better to take a hierarchical approach. Let every site or organization manage resources individually, using local policy, and allow access to these sites as a single entity. In other words, use tools such as Sun N1 Grid Engine software, Portable Batch System (PBS) or Load Sharing Facility (LSF) for local resource management, and grid middleware like Globus Toolkit for interconnecting sites.

This way, we only need one point of access (only one machine–the gatekeeper) for each site in a grid. This approach provides a performance gain, because local schedulers and job managers can use resources under their administration at close to 100 percent utilization rates, due to optimizations that are possible in homogeneous environments like clusters.

Prerequisites

This section defines the prerequisites for Globus Toolkit and Solaris 9 OS and describes the platforms tested.

The integration of Globus Toolkit (versions 2.4 and 3.0.2) and Sun N1 Grid Engine version 5.3 was successfully performed according to the information in this article on an Intel Pentium 4 machine with 256 Megabytes of RAM at the Computer Systems Laboratory, National University of Singapore.

During the tests, the Globus Toolkit 3.*x* enabled-rocks cluster in the Computer System Laboratory was involved for Globus Toolkit cross-site testing. Jobs were submitted from one site to another and executed successfully. During the tests, heterogeneous platforms (Solaris OS and Linux) were used, and providing proof of the Globus Toolkit's ability to interconnect distributed systems running on various platforms.

Solaris 9 OS

The Solaris 9 OS platform-specific prerequisites are as follows:

- GNU CC – Solaris 9 OS ships with a CC compiler, but Globus Toolkit 3.*x* requires a GNU compatible C compiler (GCC). Download and install the `gcc-3.2.2-sol9-intel-local` package from `http://www.sunfreeware.com`.
- GNU make – The make utility that ships with Solaris 9 OS is not compatible with the Globus Toolkit 3.*x* installation. Download and install the `make-3.80-sol9-intel-local` package from `http://www.sunfreeware.com`.
- `autoconf` – This package is needed for source installation. Download and install the `autoconf-2.57-sol9-intel-local` package from `http://www.sunfreeware.com`.
- `automake` – This package is needed for source installation. Download and install the `automake-1.7.2-sol9-intel-local` package from `http://www.sunfreeware.com`.
- Perl 5 – Solaris 9 OS ships with Perl 5.6.1 binary distribution, which is built with the CC compiler. You must install a Perl version built with GCC. Download the Perl source distribution from `www.perl.org`, and build it with the GCC compiler. For Globus Toolkit 3.*x* and Sun N1 Grid Engine software integration, we used Perl 5.8.0.

- GNU tar – Solaris 9 OS ships with a version of the tar utility that is not fully compatible with Globus Toolkit 3.*x*. The Globus Toolkit 3.*x* source bundles contain tar.gz archives that fail to uncompress during the installation. To overcome this, download and install the tar-1.13.19-sol8-intel-local package from http://www.sunfreeware.com.
- m4 – With the m4 version that ships with Solaris 9 OS, the bison tool fails to function properly. Download and install the m4-1.4-sol9-intel-local package from http://www.sunfreeware.com.

After installing the software, make sure that the newly installed utilities are used instead of the old ones. All the binary packages mentioned install in the /usr/local location, so you must make sure that the PATH environment variable contains /usr/local/bin *before* /usr/bin, /usr/ccs/bin, and /usr/ucb.

Also, make sure that the newly installed Perl interpreter is invoked.

After installing GCC, you must add the libgcc library to the list of system libraries. Use the crle utility to do this. For instance, if you install GCC to the /usr/local location, you would add the library as follows:

```
# crle –l /usr/lib:/usr/local/lib
```

You must have /usr/ccs/bin present in your PATH environment variable so that the installer can find the tar utility, and you must create a link named cc to the GCC compiler so that GCC is invoked instead of CC.

Sun N1 Grid Engine

No prerequisites exist for the Sun N1 Grid Engine software. Refer to the Sun N1 Grid Engine software documentation for more information.

Globus Toolkit

The Globus Toolkit is provided both as a source distribution and a binary distribution for a few platforms. However, it is recommended that you install Globus Toolkit from the source distribution.

The prerequisites for a source installation, as specified on the Globus official website, are as follows:

- JDK™ 1.4 or newer – Version 1.4 is recommended; Solaris 9 OS ships with Java 1.4 preinstalled. Note that JDK 1.3.1 is supported, but the JAAS library must be installed separately.
- Jakarta Ant 1.5 – Solaris 9 OS ships with Ant 1.4.1 preinstalled, but version 1.5 is recommended. Note that Ant 1.4 is supported, but the `crimson.jar` library must be replaced with the Globus Toolkit 3.*x* `xerces.jar` library.
- Junit 3.8.1 – The `junit.jar` library.
- YACC or Bison – For a source installation, a syntactical parser is needed. Solaris 9 OS does not come with a syntactical parser preinstalled. The `bison-1.75-sol9-intel-local` binary package is available at `http://www.sunfreeware.com`.

To install Ant software, download the latest version from `www.jakarta.org` and unzip it to your preferred location. Then, download the `junit.jar` library from `www.junit.org`, and copy it to the Ant `lib/` directory.

Globus Toolkit Installation

After you have met all of the prerequisites, proceed with the Globus Toolkit version 3.*x* installation. It is recommended that you create a special system account (for example, `globus`) to own the Globus Toolkit version 3.*x* files, then perform the installation as this user. The Globus Toolkit tries to minimize the usage of super-user privileges to avoid security problems.

This section documents all Globus Toolkit 3.*x* installation steps. The installation steps are not specific to the Solaris OS platform, but are common to all UNIX operating systems.

Phase 1: Installing Globus Toolkit 3.*x* Core

The following steps guide you through the process of installing the Globus Toolkit version 3.*x* core.

▼ To Install Globus Toolkit 3.*x* Core

1. **Set the** `JAVA_HOME` **environment variable to point to your JDK distribution; add** `$JAVA_HOME/bin` **to your** `PATH` **environment variable.**

2. **Set the** `ANT_HOME` **environment variable to point to your Ant distribution; add** `$ANT_HOME/bin` **to the** `PATH` **environment variable.**

3. **Download the Globus Toolkit version 3.*x* source installation bundle from** `http://www-unix.globus.org/ftppub/gt3/3.0/3.0.2/gt3.0.2-source-installer.tar.gz`**, and extract it to your preferred location, which is referred to as** <*`GLOBUS_SOURCE_INSTALLER`*> **from here on.**

4. **Choose an installation directory for Globus Toolkit version 3.*x*.**

   This directory is different than the source distribution directory and from here on is referred to as <*`GLOBUS_LOCATION`*>.

5. **Change directory to the** <*`GLOBUS_SOURCE_INSTALLER`*> **directory, and run the** `install-gt3` **installation script.**

   It is recommended that you edit the `install-gt3` script and append `\ -verbose` to the definitions of the `GPT_BUILD` and `GPT_INSTALL` variables. This increases verbosity. You should capture the installation log to a file as follows:

   ```
   $ ./install-gt3 <GLOBUS_LOCATION> 2>&1| tee install.log
   ```

6. **After the script exits, check the log for errors:**

   ```
   $ grep ERROR install.log
   ```

   If there are no errors, continue with the installation. Otherwise, check the log file to find the cause of the installation error.

7. **Change directory to the** <*`GLOBUS_LOCATION`*> **directory and run** `ant setup`**.**

8. **Log in as super-user, change directory to the** *<GLOBUS_LOCATION>* **directory, and run the following commands:**

```
# mkdir <JAVA_HOME>/jre/lib/endorsed

# cp endorsed/xalan.jar <JAVA_HOME>/jre/lib/endorse
```

where *<JAVA_HOME>* is the directory of your Java distribution.

Phase 2: Installing Certificate Authority (CA) and Issuing Certificates

Security is one of the main concerns of Globus Toolkit 3.*x*. The Globus Toolkit is built upon the Grid Security Infrastructure (GSI) layer. GSI provides authentication, access control, and confidentiality of data. It uses PKI and digital certificates to authenticate grid entities.

You need a host certificate for your host, an LDAP certificate for your directory service, and a user certificate for each grid user. A trusted Certificate Authority (CA) must sign all these certificates. You can use your own existing CA, you can use the Globus CA, or you can install your own simple CA with basic functionality for testing purposes.

The first option is not always available. The second one is quite impractical because there is a delay of approximately two days between the time you request a signed certificate, and the time it is sent to you by the Globus CA. Thus, for testing purposes, it is recommended that you install your own CA.

▼ To Install Your Own CA

1. **Download the Globus Toolkit CA bundle from** `ftp://ftp.globus.org/pub/gsi/simple_ca/globus_simple_ca_bundle-latest.tar.gz`**.**

2. **Copy the simple CA bundle to the** *<GLOBUS_SOURCE_INSTALLER>* **directory, then issue the following commands from the account you are using to install Globus Toolkit 3.*x*:**

```
$ export <GLOBUS_LOCATION>=/path/to/globus/installation

$ $<GLOBUS_LOCATION>/sbin/gpt-build  \

  globus_simple_ca_bundle-latest.tar.gz  <flavor>

$ $<GLOBUS_LOCATION>/sbin/gpt-postinstall
```

where *flavor* is the platform for which you want to install. For more information on flavors, consult the *Globus Toolkit 3 Administration Guide*. Examples of flavors are `gcc32`, `gcc32dbg`, and `gcc32pthr`.

Note – During the post-installation phase, you will be prompted to provide information about the newly created CA; make sure this information is valid.

In the `$HOME/.globus` directory of the user account you want to use to perform the installation, you can now find the simple CA directory where the CA has been installed. The `cacert.pem` file represents the CA public key. Also in this directory, you can find a tarball named `globus_simple_ca_<`*`CA-HASH`*`>_setup.tar.gz` where *CA-HASH* represents the hashcode for the CA certificate (`cacert.pem`).

3. **Compute the hashcode for your certificate using the following command:**

```
$ openssl x509 -in <certificate_file> -hash –noout
```

This tarball can then be distributed to other hosts that want to be clients for your CA, or trust your CA. They can install it using `gpt`. Your CA certificate file with other needed configuration files will be installed on the client machines.

4. **To setup the grid security for your installation, log in as `root`, and run the following command:**

```
# $<GLOBUS_LOCATION>/setup/globus_simple_ca_<hashcode>_setup/
setup-gsi
```

   This configures the security infrastructure to use the newly created CA and create the proper directory structure under the `/etc/grid-security` directory.

To Install a Globus Toolkit CA

- **If you choose to install and set up the Globus Toolkit CA instead of your own CA, run the `setup-gsi` script under the `$<`*`GLOBUS_LOCATION`*`>/setup` directory.**

  Under the `$<`*`GLOBUS_LOCATION`*`>/setup` directory is the `globus` directory that contains the data needed to install the Globus CA as your trusted CA.

To Issue Certificates

Note – After setting up grid security, you must create certificates for your grid host, for your directory service LDAP server, and for your users. The tools to create and sign the certificate are located in `$<`*`GLOBUS_LOCATION`*`>/bin`.

---

1. **Add the `$<`*`GLOBUS_LOCATION`*`>/bin` location to the `PATH` environment variable.**

2. **To create a certificate request for your host, use the following command syntax:**

```
$ grid-cert-request -service host -host <hostname>
```

   For example:

```
$ grid-cert-request -service host -host \
solaris.ddns.comp.nus.edu.sg
```

   This certificate is used by the GRAM and GridFTP services.

3. **To create a certificate for your LDAP server, use the following command syntax:**

```
$ grid-cert-request -service ldap –host <hostname>
```

For example:

```
$ grid-cert-request -service ldap –host \
solaris.ddns.comp.nus.edu.sg
```

4. **To create a certificate request for your user, use following command syntax:**

```
$ grid-cert-request
```

After you create your certificates, you must sign them using your CA. You can sign the certificates using the `grid-ca-sign` utility.

5. **Sign the certificates using the** `grid-ca-sign` **utility.**

```
$ grid-ca-sign –in <certificate_request.pem> -out
<certificate.pem>
```

6. **Obtain the signed certificates from the CA, and copy them to the proper locations:**

- `/etc/grid-security` for the host certificate
- `/etc/grid-security/ldap` for your LDAP certificate
- `$HOME/.globus` for the user certificate

Ensure that the certificates for the trusted CA are saved in the directory `/etc/grid-security/certificates/`. For example, the file `42864e48.0` is the trusted certificate for Globus Certification Authority.

After installing the host certificate, proceed with the next phase of installation, installing the Master Managed Job Factory Service (MMJFS), as described in the next section.

Phase 3: Installing Master Managed Job Factory Service

After you have the host certificate in place, install the Master Managed Job Factory Service (MMJFS), logged in as the user you performed the core installation as.

To Install MMJFS

1. **Change directory to the** *<GLOBUS_SOURCE_INSTALLER>* **directory, and run the** `install-gt3-mmjfs` **installation script.**

   It is recommended that you edit the `install-gt3-mmjfs` script and append `\ -verbose` to the definitions of the `GPT_BUILD` and `GPT_INSTALL` variables.

   This recommendation increases verbosity. It is recommended that you capture the installation log to a file as follows:

   ```
   $ ./install-gt3-mmjfs <GLOBUS_LOCATION> 2>&1| tee install-
   mmjfs.log
   ```

2. **After the script exits, check the log for errors:**

   ```
   $ grep ERROR install-mmjfs.log
   ```

   If there are no errors, continue with the installation. Otherwise, check the log file to find the cause of the installation error.

3. **Log in as the super-user, and run the script** `setperms.sh`:

   ```
   # $<GLOBUS_LOCATION>/bin/setperms.sh
   ```

   Proceed to the next section for configuration and testing.

Globus Toolkit Configuration and Testing

This section describes how to configure and test the Globus Toolkit software.

Configuring Files

The Globus Toolkit 3.*x* services and service container have a large set of configuration files, most of them stored under the `$<`*`GLOBUS_LOCATION`*`>` and `$<`*`GLOBUS_LOCATION`*`>/etc` directories.

The following two configuration files need to be set up to achieve a minimal working configuration: `/etc/grid-security/grid-mapfile` and `/etc/grid-security/grim-port-type.xml`.

The `grid-mapfile` file contains the mappings between grid users and local user accounts. A grid user is identified by a digital certificate, signed by a trusted CA. The `grid-mapfile` contains mappings between X509 distinguished names (DNs), representing subjects of certificates, and local user account names. When a grid user requests to submit a job, it is first authenticated based on its certificate. After that, the job is started as the local user mapped to the grid user's x509 distinguished name (DN) in the `grid-mapfile`.

▼ To Configure Grid Services

1. **To be able to access grid services provided by your host, edit the** `grid-mapfile` **and add a line such as the following:**

```
"<DN certificate subject>" localuser
```

for each grid user, where the DN can be obtained by running the `grid-cert-info –subject –file <`*`certificate-file`*`>` command as a user that has obtained a certificate from the CA. An example entry is as follows:

```
"/O=Grid/OU=GlobusTest/CN=Gabriel Ghinita" globus
```

2. **To configure the MMJFS, edit the** `grim-port-type.xml` **file to tell it which local account the MMJFS should be used when it is invoked through the web-service container.**

   Its content should be similar to the following:

```
<authorized_port_types>
<port_type username="globus">http://www.globus.org/namespaces/
managed_job/
managed_job/ManagedJobPortType
</port_type>
</authorized_port_types>
```

Configuring Globus Toolkit Services

To complete the Globus Toolkit configuration, you need to configure Globus Toolkit services to start automatically at system startup.

- If using Globus Toolkit 2.4, you need to start GRAM and GridFTP through the `inetd` daemon and GRIS as a standalone daemon. Use the instructions that follow for configuring GRAM, MDs, and GridFTP.
- If using Globus Toolkit 3.*x*, you only need to start the grid services container.

▼ To Configure GRAM

1. **Add the following line to the** `/etc/services` **file on your system**:

```
gsigatekeeper      2119/tcp  # Globus Gatekeeper
```

2. **Add the following line to the** `/etc/inetd.conf` **file on your system**:

```
gsigatekeeper stream tcp nowait root
/usr/bin/env env LD_LIBRARY_PATH=<GLOBUS_LOCATION>/lib
<GLOBUS_LOCATION>/sbin/globus-gatekeeper
-conf <GLOBUS_LOCATION>/etc/globus-gatekeeper.conf
```

   where <*GLOBUS_LOCATION*> is the actual location of Globus Toolkit on your system.

3. **Restart the** `inetd` **daemon** `pkill –HUP inetd`**.**

To Configure MDS

1. **Create the `/etc/init.d/mds` script and edit it to read as follows:**

   ```
   <GLOBUS_LOCATION>/sbin/globus-mds start
   ```

   where <*GLOBUS_LOCATION*> is the actual location of Globus Toolkit on your system.

2. **Create a symbolic link to the `MDS` script from the location corresponding to your default run-level.**

   For example:

   ```
   # ln –s /etc/init.d/mds /etc/rc3.d/S90mds
   ```

   The GRIS daemon will start on the default port 2135.

To Configure GridFTP

1. **Add the following line to the `/etc/services` file on your system:**

   ```
   gsiftp  2811/tcp # Grid FTP service
   ```

2. **Add the following line to the `/etc/inetd.conf` file on your system:**

   ```
   gsiftp  stream  tcp     nowait  root
   /usr/bin/env env LD_LIBRARY_PATH=<GLOBUS_LOCATION>/lib
   <GLOBUS_LOCATION>/sbin/in.ftpd -l
   -Z <GLOBUS_LOCATION>/var/globus-gsiftp.log
   -a -G <GLOBUS_LOCATION>
   ```

   where <*GLOBUS_LOCATION*> is the actual location of Globus Toolkit on your system. The `–Z` flag turns on the logging facility.

3. **Restart the `inetd` daemon `pkill –HUP inetd`.**

To Configure the Grid Services Container

1. **To start the web services container as a service at system startup, create an** `/etc/init.d/globus-container` **script, then edit it to read as follows:**

   ```
   <GLOBUS_LOCATION>/bin/globus-start-container –p 8080
   >container.log 2>&1 &
   ```

   where `<GLOBUS_LOCATION>` is the actual location of Globus Toolkit on your system.

   You can change `container.log` to the path where you want to keep your container log file, and you can change the port number the container runs on (from the default of 8080).

2. **Create a** `symlink` **to the** `globus-container` **script from the location corresponding to your default run-level, for example:**

   ```
   # ln –s /etc/init.d/globus-container /etc/rc3.d/S92globus-
   container
   ```

Testing Globus Toolkit Services

This section provides instructions for testing the Globus Toolkit services.

To Create a Proxy Certificate

1. **Log in as a user who owns a certificate signed by a trusted CA.**

   Make sure that your `PATH` environment variable contains the `$<GLOBUS_LOCATION>/bin` directory and that the `LD_LIBRARY_PATH` environment variable contains the `$<GLOBUS_LOCATION>/lib` directory.

2. **Create a proxy certificate by entering the following command:**

   ```
   $ grid-proxy-init –debug –verify
   ```

   This proxy certificate will be used for the tests described in the remainder of this section.

3. **To check the validity of your proxy, use the following command:**

```
$ grid-proxy-info
```

Testing Globus Toolkit 2.4 Services

This section describes how to test Globus Toolkit version 2.4 services.

Make sure that you have a proxy certificate, created as described earlier in this section.

▼ To Test GRAM

1. **To test the GRAM service, run the following command:**

```
$ globus-job-run <hostname>:<port>/jobmanager /bin/date
```

where <*hostname*> and <*port*> identify the machine and port number the `globus-gatekeeper` daemon runs on (port number defaults to 2119). For example:

```
$ globus-job-run solaris.ddns.comp.nus.edu.sg/jobmanager /bin/
date
```

You should get the output of the `date` command. If you get an error, check the `$<`*`GLOBUS_LOCATION`*`>/var/globus-gatekeeper.log` file.

To Test GRIS and MDS

1. **To test the GRIS server running at your Globus Toolkit site, run the following command:**

```
$ grid-info-search -h <hostname> -p <port>  -anonymous –L
```

where <*hostname*> and <*port*> identify the machine and port number the GRIS daemon runs on (port number defaults to 2135). For example:

```
$ grid-info-search -h solaris.ddns.comp.nus.edu.sg  -anonymous –L
```

You should see an output of the directory service contents from the GRIS server.

To Test GridFTP

1. **To test the** `gsiftp` **server running at your Globus Toolkit site, run the following command:**

```
$ globus-url-copy -s "`grid-cert-info -subject`" \

  <source_url> <dest_url>
```

where the <*source_url*> and <*dest_url*> allow the `gsiftp` and `file` protocols. For example:

```
$ globus-url-copy -s "`grid-cert-info -subject`" \

  gsiftp://localhost:2811/tmp/src file:///tmp/dst
```

2. **Verify that the destination file has been created, and that it is identical to the source file.**

Testing Globus Toolkit 3.*x* Services

This section describes how to test Globus Toolkit version 3.*x* services.

Make sure that you have a proxy certificate, created as described earlier in this section.

To Test MMJFS

1. **From the grid user account, execute the following command:**

```
$ managed-job-globusrun -factory \ http://<hostname>:<port>/ogsa/
services/base/gram/MasterForkManagedJobFactoryService -file
$<GLOBUS_LOCATION>/etc/test.xml
```

For example:

```
$ managed-job-globusrun -factory \ http://
solaris.ddns.comp.nus.edu.sg:8080/ogsa/services/base/gram/
MasterForkManagedJobFactoryService -file $<GLOBUS_LOCATION>/etc/
test.xml
```

In the home directory of the local user account mapped to your grid user DN, in the `grid-mapfile` file, you should find two files, `stderr` and `stdout`, corresponding to the execution of the submitted job. You can submit your own job description file to start a different job. The file must be in the Resource Specification Language (RSL) format. The file used in the previous example ships with the Globus Toolkit 3.*x* distribution.

2. **Check the container output for errors.**

Note – It is possible for the container to dump a set of exceptions, but the submitted job still finishes correctly. This action is not abnormal, and it is caused by the fact that services are instantiated on-the-fly. Additionally, it is possible that a service request is issued before the service has completed initialization.

▼ To Test Index Services

1. **Use the `ogsi-find-service-data` client utility provided by the Globus Toolkit 3.*x* distribution to run the following command:**

```
$<GLOBUS_LOCATION>/bin/ogsi-find-service-data
```

where <*GLOBUS_LOCATION*> is the actual location of Globus Toolkit 3.*x* on your system; and you receive output from the index services for the `http://localhost:8080/ogsa/services/base/registry/ContainerRegistryService` service.

If you want to obtain information about other services or learn more about the `ogsi-find-service-data` utility, go to `http://www.globus.org/ogsa/releases/final/docs/infosvcs/querysvcdata.html`.

▼ To Test RFT Services

- **Use the `RFTClient` utility that ships with Globus Toolkit 3.*x*.**

`RFTClient` is a Java™ class that you can invoke from your Java Virtual Machine for the Java platform (JVM™). Documentation and usage examples for RFT are available at `http://www-unix.globus.org/toolkit/reliable_transfer.html`.

Sun N1 Grid Engine Software Installation

Sun N1 Grid Engine software is not bundled with the Solaris 9 OS. You can download the Sun N1 Grid Engine software binary from `http://wwws.sun.com/software/gridware/sge.html`, or the source distribution from `http://gridengine.sunsource.net`. The newest version is Sun N1 Grid Engine version 5.3p4. For this integration project and article, the binary packages for version 5.3p2 were used.

```
sge-5_3p2-common: contains Sun N1 Grid Engine software
architecture-independent files

sge-5_3p2-doc: contains Sun N1 Grid Engine software
documentation

sge-5_3p2-bin-sia: contains Sun N1 Grid Engine software binaries
for x86 architecture
```

The install directory for these packages is `/gridware/sge`.

▼ To Install Sun N1 Grid Engine Software

1. **Download the Sun N1 Grid Engine software binary from** `http://wwws.sun.com/software/gridware/sge.html`**, or the source distribution from** `http://gridengine.sunsource.net`**.**

2. **Install the packages on the machines on your cluster.**

3. **Run the** `./install_qmaster` **script on your Sun N1 Grid Engine master host and the** `./install_execd` **script on your execution hosts.**

   For more information on the Sun N1 Grid Engine software installation, configuration, creating and managing queues, and parallel environments, please consult the *Sun N1 Grid Engine Software Administration Guide*.

For this project, we assume a standard configuration containing:

- One master host running the sge_qmaster and sge_schedd daemons
- One or more execution hosts running the sge_execd daemons (the sge_commd communication daemon is running on both master and execution hosts).
- A MPI/MPICH distribution installed and MPI/MPICH parallel environment enabled for Sun N1 Grid Engine software.

Note – The Sun N1 Grid Engine software master host must be the same as the host running the globus-gatekeper daemon. This requirement is because the integration software between Globus Toolkit and Sun N1 Grid Engine software is invoking Sun N1 Grid Engine software submission and status commands directly on the local machine. As an alternative, it is possible to alter the code that invokes the Sun N1 Grid Engine software commands from Globus Toolkit to use, for example, ssh, and submit jobs to another machine, but this is beyond the scope of this article.

Integrating Sun N1 Grid Engine Software With Globus Toolkit

Currently, there is no utility software provided by Globus that provides integration between the Globus Toolkit and Sun N1 Grid Engine software (as there is for other cluster job managers such as PBS or LSF), but there are a few open source packages that enable Globus Toolkit to submit jobs to Sun N1 Grid Engine software job managers.

For this article, we used the integration software provided by the London e-Science Centre. For more information, refer to http://www.lesc.ic.ac.uk/projects/epic-gt-sge.html.

▼ To Integrate Sun N1 Grid Engine Software With Globus Toolkit

1. **Download the following three packages:**
   - globus_gram_job_manager_setup_sge-0.11.tar.gz, which contains Perl code to generate SGE scripts from RSL specification
   - mmjfs_sge_setup-0.0.tar.gz, which configures the MasterSGEManagedJobFactoryService as a Globus Toolkit 3.*x* service
   - mjs_sge_setup-0.0.tar.gz, which provides the job execution service used by MMJFS

2. **Copy the packages to your** $<*GLOBUS_SOURCE_INSTALLE*R> **directory, then install them.**

```
$ gpt-build globus_gram_job_manager_setup_sge-0.11.tar.gz
$ gpt-build mmjfs_sge_setup-0.0.tar.gz
$ gpt-build mjs_sge_setup-0.0.tar.gz
$ gpt-postinstall
```

   It is possible for your MPI distribution not to be properly detected during the installation.

3. **If the MPI distribution is not detected, edit the** <*GLOBUS_LOCATION*>/lib/perl/Globus/GRAM/JobManager/sge.pm **file, checking the line defining the** mpirun **variable.**

   This line should point to your mpirun executable.

   a. **If this line does not point to your** mpirun **executable, modify the value of the variable to do so.**

   b. **Check the** pe_mpi **variable to ensure that it has the value** mpi **or** mpich **corresponding to the MPI parallel environment configured for Sun N1 Grid Engine software.**

   There is one minor bug in the integration code, specifically in the section that translates RSL requests to Sun N1 Grid Engine software job scripts, that causes MPI jobs to fail.

4. **To fix the minor bug in the integration code, edit the** `sge.pm` **script at the section containing the lines** “`Where to write output and error?`” **by modifying the lines on the** `else` **branch:**

```
 $sge_job_script->print("#\$ -o " . $description->stdout() .
".real\n");

   $sge_job_script->print("#\$ -e " . $description->stderr() .
".real\n");
```

to read as follows:

```
$sge_job_script->print("#\$ -o " . $description->stdout() . "\n");

    $sge_job_script->print("#\$ -e " . $description->stderr() .
"\n");
```

The problem is that for some job runs, when the script tries to create the `/dev/null.real` files and fails, the job terminates in error. This fix solves the problem.

Integration Testing

To test that the integration of Sun N1 Grid Engine software with the Globus Toolkit was successful, submit some jobs to the Sun N1 Grid Engine software job manager of both Globus Toolkit 2.4 and Globus Toolkit 3.*x*.

Make sure that you have a proxy certificate, created as described in “Testing Globus Toolkit Services” on page 18.

▼ To Test Integration With Globus Toolkit 2.4

- **To test the Globus Toolkit 2.4 GRAM service, run the following command:**

```
$ globus-job-run <hostname>:<port>/jobmanager-sge /bin/date
```

where <*hostname*> and <*port*> identify the machine and port number the `globus-gatekeeper` daemon runs on (port number default is 2119). For example:

```
$ globus-job-run solaris.ddns.comp.nus.edu.sg/jobmanager-sge /
bin/date
```

▼ To Test Integration With Globus Toolkit 3.*x*

- **To test the Globus Toolkit 3.x grid services container, run the following command:**

```
$ managed-job-globusrun -factory \ http://<hostname>:<port>/ogsa/
services/base/gram/MasterSGEManagedJobFactoryService -file
<GLOBUS_LOCATION>/etc/test.xml
```

where <*hostname*> and <*port*> identify the grid services container. For example:

```
$ managed-job-globusrun -factory http://
solaris.ddns.comp.nus.edu.sg:8080/ogsa/services/base/gram/
MasterSGEManagedJobFactoryService -file <GLOBUS_LOCATION>/etc/
test.xml
```

Troubleshooting

During the installation of Globus Toolkit 3.*x* on the Solaris 9 OS x86 platform, various problems and installation errors were encountered. Most of them were related to the incompatibility between the Globus Toolkit 3.*x* installation package and the software tools preinstalled with the OS. The Solaris OS prerequisites documented in “Prerequisites” on page 6 are meant to overcome these problems. If you meet the prerequisites described in this document, the installation should complete without any problems.

About the Authors

Dr. Chong-Wee Simon See, Ph.D.

Dr. Chong-Wee Simon See (Simon) is the Director of High Performance Computing Technology at Sun Microsystems Inc., Asia. Also, he is the Director for the Sun Asia Pacific Science and Technology Center. In addition to his employment at Sun, Simon is currently an Adjunct Associate Professor at Nanyang Technological University and National University of Singapore. His research interests are in the areas of high performance computing, computational science, applied mathematics, and simulation methodology. He has published over 40 papers in these areas and has won various awards.

Dr. See graduated from University of Salford (UK) with a Ph.D. in electrical engineering and numerical analysis in 1993. Prior to joining Sun, Dr. See worked for Silicon Graphics Inc. (SGI), DSO National Lab of Singapore, IBM, and International Simulation Ltd. (UK). He is a consultant with national research and supercomputing centers.

Gabriel Ghinita

Gabriel Ghinita is currently a Ph.D. candidate at the National University of Singapore (NSU). His research interest is in distributed computing, especially grid computing. He is a member of the Computer Systems Lab at NUS, under the supervision of adjunct professor Teo Yong Meng.

Gabriel graduated from Politehnica University of Bucharest in 2003. His honors project, “Object Communication Issues on the ALiCE Grid,” was completed during a six-month internship (February 2003 to August 2003) at NUS. He has been accepted to the graduate program at NUS as of January 2004.

Related Resources

Publications

- Ferreira, Luis, Berstis, Viktors, et al. *Introduction to Grid Computing with Globus*, IBM Redbooks Collection, September 2003.
- Jacob, Bart, Ferreira, Luis, et al. *Enabling Applications for Grid Computing with Globus*, IBM Redbooks Collection, June 2003.
- Ferreira, Luis, Jacob, Bart, et al. *Globus Toolkit 3.0 Quick Start*, IBM Redbooks Collection, September 2003.
- *Globus Toolkit 3 Administration Guide*, Globus Toolkit online documentation, http://www-unix.globus.org/toolkit/3.0/ogsa/docs/admin/.
- *Globus Simple CA Package*, Globus Toolkit online documentation, http://www.globus.org/security/simple-ca.html.
- *Sun ONE Grid Engine 5.3 Administration and User's Guide*, http://www.sun.com/products-n-solutions/hardware/docs/pdf/816-2077-12.pdf.
- *EPIC - Sun N1 Grid Engine Integration with the Globus Toolkit*, http://www.lesc.ic.ac.uk/projects/epic-gt-sge.html.
- *Solaris 9 System Administration Guide: Basic Administration*, http://docs-pdf.sun.com/806-4073/806-4073.pdf.
- Foster, I., et al. *The Anatomy of the Grid: Enabling Scalable Virtual Organizations*, International Supercomputer Applications 2001.
- Foster, I., et al. *The y of the Grid: An Open Grid Services Architecture for Distributed Systems Integration*, Mathematics and Computer Science Division, Argonne National Laboratory 2002.

Web Sites

Note – Sun is not responsible for the availability of third-party Web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Sun will not be responsible or liable for any actual or alleged damage or loss caused by or in connection with the use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

- Globus Toolkit: `http://globus.org`
- Solaris 9 Operating System: `http://wwws.sun.com/software/solaris/`
- Sun N1 Grid Engine 5.3: `http://wwws.sun.com/software/gridware/sge.html`
- Sun Grid Open Source Community: `http://gridengine.sunsource.net`

Ordering Sun Documents

The SunDocs^{SM} program provides more than 250 manuals from Sun Microsystems, Inc. If you live in the United States, Canada, Europe, or Japan, you can purchase documentation sets or individual manuals through this program.

Accessing Sun Documentation Online

The `docs.sun.com` web site enables you to access Sun technical documentation online. You can browse the `docs.sun.com` archive or search for a specific book title or subject. The URL is `http://docs.sun.com/`

To reference Sun BluePrints OnLine articles, visit the Sun BluePrints OnLine Web site at: `http://www.sun.com/blueprints/online.html`