

Introduction to GT3

- Background
 - The Grid Problem
 - The Globus Approach
 - OGSA & OGSF
 - Globus Toolkit
- GT3 Architecture and Functionality: The Latest Refinement of the Globus Toolkit
 - Core
 - Base Services
 - User-Defined Services
 - Future Directions
- **Installation and Administration**
 - Installation
 - Configuration
 - Debugging
 - Support
- Important Things to Remember

Overview

- Installing GT3
- Overview of installed services
- Running clients and services
- Configuring GT3
- Debugging
- Support

Overview of Installing GT3

- Prerequisites
 - JDK, ant
- Optional tools
 - Other hosting environments
- Installation
 - GPT installer or Ant-only
- Required post-installation setup
 - Acquiring certificates
 - Setting permissions

Installation Pre-requisites GT3.2

http://www-unix.globus.org/toolkit/docs/3.2/installation/install_support.html

- JDK 1.3.1+
 - Xindice + Sun JDK1.3.1 + Linux has errors for some versions of Linux
 - Pre-1.4.1 JDKs require JAAS as a separate download
- Ant 1.6.1
 - Required for source builds, recommended for binary
- Junit 3.8.1
 - Required for grid services
- C compiler – anything but gcc 3.2
- Gnu tar
- YACC
 - For cbindings and findServiceData from source
- JDBC compliant database
 - Only required for RFT, RLS

Optional Tools

- Alternate hosting environments
 - Jakarta Tomcat 4.1.24 (4.0.6 has also been tested to work)
 - > **Used for:** your web service container. This is optional because the GT3 installation provides a standalone web service container for testing purposes.
 - *JBOSS*
 - *Websphere*
 - *and more ...*
- Microsoft .NET Framework
- Junit for testing

Installing GT 3.2 Binary (Unix)

http://www-unix.globus.org/toolkit/docs/3.2/installation/install_installing.html

- Make sure pre-requisites are available
 - Set JAVA_HOME
 - Add \$ANT_HOME/bin to your PATH
- Download the appropriate binary bundle
 - `./install-gt3-bin /path/to/install`
- Or download the GPT source bundle
 - `./install-gt3 /path/to/install`
 - `./install-gt3-mmjfs /path/to/install`
 - > After you have certificates

Installing GT 3.0 (Windows)

- Make sure pre-requisites are available
 - Likely to use Cygwin to get the tools you need
- Unset CLASSPATH to avoid conflicting jars
- Install GT3 core by running “ant dist” and “ant setup” in ogsa/impl/java
- Install higher-level services using “ant deployGar”

Installing GT 3.0 (Binaries)

- GPT Binary bundles available for different UNIX platforms
 - [gt3.2.1-all-linux-glibc2.2-installer.tar.gz](#) (Linux/glibc2.2.5-44 binary installer, generated on Red Hat 7.3)
 - [gt3.2.1-all-linux-rh9-installer.tar.gz](#) (Linux/glibc2.3.2-27.9 binary installer, generated on Red Hat 9)
 - [gt3.2.1-all-linux-glibc2.3-installer.tar.gz](#) (Linux/glibc2.3.2-101.4 binary installer, generated on Fedora Core 1)
 - [gt3.2.1-all-solaris9-installer.tar.gz](#) (SunOS 5.9 Generic_112233-02 sun4u sparc SUNW,Sun-Fire-280R)
- Core and Higher Level Services binaries available for Windows

Post-installation setup

http://www-unix.globus.org/toolkit/docs/3.2/installation/install_config.html

- GSI uses X.509, so need to get certificates
 - Run setup-gsi as root
 - grid-cert-request for user and host
 - Can re-use GT2 certificates if you have them already
- Run setperms.sh (after install-gt3-mmjfs)
 - This is to make globus-grim setuid to the account which owns the hostcert, and to make the UHE launcher setuid so it can create jobs on behalf of users

Review of Public Key Cryptography

- Asymmetric keys
 - A **private** key is used to encrypt data.
 - A **public** key can decrypt data encrypted with the private key.
- An X.509 certificate includes...
 - Someone's subject name (user ID)
 - Their public key
 - A "signature" from a Certificate Authority (CA) that:
 - > Proves that the certificate came from the CA.
 - > Vouches for the subject name
 - > Vouches for the binding of the public key to the subject

Public Key Based Authentication

- User sends certificate over the wire.
- Other end sends user a challenge string.
- User encodes the challenge string with private key
 - Possession of private key means you can authenticate as subject in certificate
- Public key is used to decode the challenge.
 - If you can decode it, you know the subject
- Treat your private key carefully!!
 - Private key is stored only in well-guarded places, and only in encrypted form

X.509 Proxy Certificate

- Defines how a short term, restricted credential can be created from a normal, long-term X.509 credential
 - A "proxy certificate" is a special type of X.509 certificate that is signed by the normal end entity cert, or by another proxy
 - Supports single sign-on & delegation through "impersonation"

User Proxies

- Minimize exposure of user's private key
- A temporary, X.509 proxy credential for use by our computations
 - We call this a user proxy certificate
 - Allows process to act on behalf of user
 - User-signed user proxy cert stored in local file
 - Created via "grid-proxy-init" command
- Proxy's private key is not encrypted
 - Rely on file system security, proxy certificate file must be readable only by the owner

Delegation

- Remote creation of a user proxy
- Results in a new private key and X.509 proxy certificate, signed by the original key
- Allows remote process to act on behalf of the user
- Avoids sending passwords or private keys across the network

Overview of Installed Services

- What just installed, and how?
- Bundles on Unix and Windows:
 - GT3 core + higher-level services
- Bundles on Unix only:
 - GRAM bundle + GT2 dependencies
 - Cbindings bundle + client
 - Replica Location Service (RLS)
 - GT2 components

Other "Services" Bundled with GT3

- GridFTP
 - Used by RFT
- Replica Location Service (RLS)
 - Distributed registry service that records the locations of data copies and allows discovery of replicas
 - Designed and implemented in a collaboration between the Globus and DataGrid projects

The interfaces for these services are not yet OGSi-Compliant

Where did they install?

- /etc/grid-security
 - certificates/ subdirectory of trusted CAs
 - grid-mapfile
 - grim-port-type.xml
 - hostcert.pem, hostkey.pem
 - grid-security.conf
- \$GLOBUS_LOCATION
 - Everything else

Location of GARs

- *Grid Archive*, or *GAR file* is a single file which contains all the files and information the grid services container need to *deploy* a service
- Before the GARs are deployed, a copy is stored in gars/
- Contains the client and server Webservices Deployment Descriptor (WSDD), as well as the jar files
- To change the main server-config.wsdd, can edit the service's .wsdd file and re-deploy

GPT Wrappers

- GPT is a collection of packaging tools built around an XML based packaging data format.
- The GARs apply to both Windows and Unix
- GPT wraps the GAR with metadata, including dependency information and version number
- Allows for easier upgrades, and for other software to indicate dependencies
- <http://www.gridpackagingtools.com/>

GT3 Services

- Core
 - OGSF hosting environment
- MMJFS
 - The single point for submitting jobs
- MJS
 - Instances created per submitted job
- GRIM
 - Security tool for creating hostcert proxies

GT3 services (cont.)

- Index
 - A point to query for information
- Aggregator
 - SDE aggregation tool. Used by index
- mds_db
 - Used by index to track data
- Providers
 - Used to obtain host information

GT3 services (cont.)

- RIPS
 - Queue information for jobs
- Filestreaming
 - Used to move stdin/stdout for jobs
- RFT
 - Reliable File Transfer
- JMS
 - JMS notification source for J2SE/J2EE
- Servicegroup
 - OGSi service groups

Dependencies

- Some backend tools see re-use from previous releases for use in resource management
- For instance, jobmanager scripts for interfacing to local scheduling systems are the same
- Besides GT2, Java CoG supplies integrated security

Overview of Running Clients and Services

- Set your environment
- Create a proxy (single sign-on)
- Available clients
 - GRAM client
 - Index clients
 - RFT client

Environment Setup

- Set your environment:
 - export GLOBUS_LOCATION
 - source etc/globus-user-env.sh
- grid-proxy-init
- This environment setup is assumed for all later slides

Starting the container

- Services run in the container
- bin/globus-start-container -p <port>
- The container will print a list of Grid Service Handles (GSH) that can be used by clients
 - <http://127.0.0.1:8080/ogsa/services/base/gram/MasterForkManagedJobFactoryService> for instance
- User's Guide has more details

Running the GRAM Client

- To submit a job:
 - bin/managed-job-globusrun -factory host{:port/service} -file etc/test.xml
- Prerequisites:
 - Authorized to use the service by grid-mapfile and grim-port-type.xml
 - Have a proxy
 - Setuid GRIM and User Hosting Environment (UHE) launcher

Running the GRAM Client (Cont.)

- etc/test.xml contains the new format for the Resource Specification Language (RSL)
- By default, outputs to /tmp/stdout /tmp/stderr
 - Probably good to customize it for yourself to avoid permissions errors
- See Resource Management links under <http://www-unix.globus.org/developer>

RSL Syntax

- Elementary form: parenthesis clauses
 - (attribute op value [value ...])
 - Now a XML schema
- Operators Supported:
 - <, <=, =, >=, >, !=
- Some supported attributes:
 - executable, arguments, environment, stdin, stdout, stderr, resourceManagerContact, resourceManagerName
- Unknown attributes are passed through
 - May be handled by subsequent tools

Reading From the Index Service

- bin/globus-service-browser
- bin/ogsi-find-service-data
 - requires C bindings
- Example:
 - ogsi-find-service-data -service
http://128.9.64.178:9009/ogsa/services/base/gram/MasterForkManagedJobFactoryService -sde Cluster
- See Information Services for more details

Using RFT

- First, start a GridFTP server
- Setup a postgres database for RFT to use
- Enter DB values into server-config.wsdd
- java org.globus.ogsa.gui.RFTClient <RFT factory> <path to transfers>
- Store transfers in a file, one URL per line
- See Data Management for more details

Using RLS

- globus-rls-server
 - New version uses PostgreSQL and psqLODBC
- bin/globus-rls-admin -p rls://serverhost
 - ping test of server
- http://www.globus.org/rls/

Configuring GT3

- Add backend schedulers to MMJFS
 - PBS, LSF, Condor, ...
 - These will be GPT setup packages. Run gpt-install and gpt-postinstall
- Configure a hierarchy of index services
- Adding new services
 - ant deployGar -Dgar.name=/path/to/gar
- Check GTR for new services, and consider publishing your own (gtr.globus.org)

Debugging

- Always go to the most basic levels first!
- If “connection refused” try telnet
- If “DB connection refused” try your native DB client first
- If trouble with your proxy, try
 - grid-proxy-init -verify -debug
- If trouble with “policy” or GRIM, try
 - bin/globus-grim -out /tmp/grim_test

Debugging (Cont.)

- Logfiles
 - Make sure to redirect container logs to a file
 - ~user/.globus/uhe-<host>/log contains the logs for the UHE running on <host>
 - GridFTP servers use syslog for keeping logs
- Increase debugging level in ogsilogging.properties file

Support

- See
 - <http://www.globus.org/toolkit/support.html>
- It gives an overview of documentation, mailing lists, and bugzilla
- <http://www-unix.globus.org/toolkit/docs/3.2/index.htm>
|
- <http://www.casa-sotomayor.net/gt3-tutorial/>