Globus Toolkit™

- Introduction to the GT2 by a Globus user

Yoshio Tanaka
yoshio.tanaka@aist.go.jp
Grid Tech. Research Center, AIST, Japan

Some of this material is:
Copyright (c) 2002 University of Chicago and The University of Southern California. All Rights Reserved. This presentation is licensed for use under the terms of the Globus Toolkit Public License. See http://www.globus.org/toolkit/download/license.html for the full text of this license.
Tutorial Goal

- Enable attendees to understand
  - Overview and architecture of the GT2
  - How each component of the GT2 works on the computational grids
  - How the GT2 is used

Introduce the GT2 from the point of view of a user
What is the Globus Toolkit?

- A Toolkit which makes it easier to develop computational Grids
- Developed by the Globus Project Developer Team (ANL, USC/ISI)
- Defacto standard as a low level Grid middleware
  - Most Grid testbeds are using the Globus Toolkit
- Latest version is 2.2.4
- Alpha version of the GT3 is available
  - Based on OGSA. Different architecture with GT2
Some notes on the Globus Toolkit (1/2)

- Globus Toolkit is not providing a framework for anonymous computing and mega-computing
  - Users are required
    - to have an account on servers to which the user would be mapped when accessing the servers
    - to have a user certificate issued by a trusted CA
    - to be allowed by the administrator of the server
  - Complete differences with mega-computing framework such as SETI@HOME
Some notes on the Globus Toolkit (2/2)

- Do not think that the Globus Toolkit solves all problems on the Grid.
  - The Globus Toolkit is a set of tools for the easy development of computational Grids and middleware
    - The Globus Toolkit includes low-level APIs and several UNIX commands
    - It is not easy to develop application programs using Globus APIs. High-level middleware helps application development.
  - Several necessary functions on the computational Grids are not supported by the Globus Toolkit.
    - Brokering, Co-scheduling, Fault Managements, etc.
  - Other supposed problems
    - using IP-unreachable resources (private IP addresses + MPICH-G2)
    - scalability (ldap, maintenance of grid-mapfiles, etc.)
What the Globus Toolkit provides?

- The Globus Toolkit provides **TOOLS** which make it **EASIER** to develop computational Grids.

Outline of the tutorial

- Introduce major components of the Globus Toolkit following a particular scenario
- Show “how the Globus Toolkit is used”
  - Testbed development using the Globus Toolkit (ApGrid Testbed is used as an example)
  - Middleware development using the Globus APIs (Ninf-G is used as a sample middleware)
User’s scenario

- Compute using the data stored in a data server at Osaka U.
- Either host A at TITECH or host B at AIST/Tsukuba is used for the computation.
- Result of the computation will be appeared on the client machine at AIST/Tokyo.
If the Globus Toolkit is not used...

1. Select a server
   A user has to check the status of each host individually

2. login to Host A using ssh

3. Get data from the data server using ftp

4. Do computation on Host A

5. Retrieve results from Host A using ftp

User @ AIST/Tokyo

Host A @ TITECH

Data Server @ Osaka U.

Host B @ AIST/Tsukuba
The Globus Toolkit makes it easier!

- Check the status of all hosts by a single operation
- Single Sign On
  - A user is required to enter passphrase only once!
- A process for the computation will be launched on Host A

User @ AIST/Tokyo
- The process can get data from the data server
- The results will be automatically back to the client user
How each component works?

- Single Sign On + delegation
  - GSI
- Information Retrieval
  - MDS
- Remote process invocation
  - GRAM
- Data Transfer
  - GridFTP, GASS
Security
GSI
GSI: Grid Security Infrastructure

- Authentication and authorization using standard protocols and their extensions.
  - Authentication: Identify the entity
  - Authorization: Establishing rights

- Standards
  - PKI, X.509, SSL, ...

- Extensions: Single sign on and delegation
  - Entering pass phrase is required only once
  - Implemented by proxy certificates
Review: PKI and X.509 certificate

- **Public Key Infrastructure (a pair of asymmetric keys)**
  - Private key is used for data encryption
  - Public key is used for data decryption
- **Every entity (users, computers, etc.) is required to obtain his/its certificate issued by a trusted Certificate Authority (CA)**
- **X.509 certificates contain**
  - Name of Subject
  - Public key of Subject
  - Name of Certificate Authority (CA) which has signed it, to match key and identity
  - Digital Signature of the signing CA
Review: PKI and X.509 certificate (cont’d)

- **X.509 certificates**
  - Similar to a driving license. Photo on the license corresponds to a public key.
  - Issued by a CA
  - Validity of the certificate depends on the opposite entity’s policy

<table>
<thead>
<tr>
<th>Name: Taro Sanso</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address: 1-1-1, Umezono, Tsukuba</td>
</tr>
<tr>
<td>Valid until Dec. 31, 2003</td>
</tr>
</tbody>
</table>

User Certificate

- **Subject DN**
- **Public Key**
- **Issuer (CA)**
- **Digital Signature**

Issued by a state/prefecture

Issued by a CA

Identify the entity

Private key (encrypted)
Certificate Authority

- CA includes two functionalities which can be separately supported.
  - Registration Authority (RA)
    - checks the validity of the entity
  - Certificate Authority (CA)
    - Issues certificates

Request to issue cert. → RA
Validity Check

User → CA
Issuing Cert.

Send a certificate
How a user is authenticated by a server

User Cert.
Subject DN
Public Key
Issuer (CA)
Digital Signature

Private key (encrypted)

QAZWSXEDC…

Send Cert.

Challenge string

Public Key of the CA

Public Key

Encrypted challenge string

QAZWSXEDC…

PL<OKNIJBN…
Requirements for security

- Single Sign on
- Client
  
- Server A
  
- Remote process creation requests*
  
- Communication*
  
- Remote file access requests*
  
- Server B
  
- Delegation

* with mutual authentication
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
User Proxies (cont’d)

Identity of the user

User Certificate
- Subject DN
- Public Key
- Issuer (CA)
- Digital Signature

Proxy Certificate
- Subject DN/Proxy
- (new) public key
- (new) private key (not encrypted)
- Issuer (user)
- Digital Signature (user)

grid-proxy-init

private key (encrypted)
sign
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
Requirements for users

- Obtain a certificate issued by a trusted CA
  - Globus CA can be used for tests
  - Run another CA for production run. The certificate and the signing policy file of the CA should be put on an appropriate directory (/etc/grid-security/certificates).

- Run `grid-proxy-init` command in advance
  - Will generate a proxy certificate. Enter PEM pass phrase for the decryption of a private key.
  - A proxy certificate will be generated /tmp directory
Summary of GSI

- Every entity has to obtain a certificate.
- Treat your private key carefully!!
  - Private key is stored only in well-guarded places, and only in encrypted form
- Create a user proxy in advance
  - Run grid-proxy-init command
  - virtual login to Grid environment
  - A proxy certificate will be generated on user's machine.
- Single sign on and delegation enable easy and secure access to remote resources.
Information Services
MDS
Resource-based Information

- **Specification of a compute resource**
  - IP address, name of the system administrator, OS version, current processor load, application software

- **Network characteristics**
  - Bandwidth and latency, protocols, logical topology

- **Globus infrastructure elements**
An extensible, reliable source of information about resources is critical efficient operation of the grid and to the construction of applications therein.

- Resource discovery
- Resource selection
- Application configuration and adaptation?

Client 1 and 2 request info directly from resources.

Client 3 uses GIIS for searching collective information.

GIIS requests information from GRIS services as needed.

Cache contains info from A and B.
- Use LDAP as Inquiry
- Access information in a distributed directory
  - Directory represented by collection of LDAP servers
  - Each server optimized for particular function
- Directory can be updated by:
  - Information providers and tools
  - Applications (i.e., users)
  - Backend tools which generate info on demand
- Information dynamically available to tools and applications
MDS Elements

- **Grid Information Service (GRIS)**
  - Provides access the attributes of a resource
  - Can be queried directly

- **Grid Index Information Service (GIIS)**
  - Provides aggregate directory for GRIS to report attributes
  - Hierarchical groups of resources

- **Lightweight Directory Access Protocol (LDAP)**
Example: retrieve load of the server

- retrieve load average of the resource
  
  ```
  % grid-info-search "(ObjectClass=MdsCpuFree)" dn Mds-Cpu-Free-1minX100
  # processors, koume.hpcc.jp, local, grid
  dn: Mds-Device-Group-name=processors, Mds-Host-hn=koume.hpcc.jp,Mds-Vo-name=local, o=grid
  Mds-Cpu-Free-1minX100: 078
  ```
Resource Management
GRAM
Process invocation on remote resources

- GRAM (Grid Resource Allocation Manager)
  - Secure invocation of processes on remote resources
    - Authentication based on GSI and authorization based on grid-mapfile
      - single sign on
      - authorization based on Unix accounts
  - File transfer and redirection of stdin/stdout using GASS
    - Automatic file staging at run time
Main components of GRAM

- **Gatekeeper**
  - A server process listening requests from clients
  - authenticate clients

- **Job manager**
  - interface to local queuing system
  - manage launched jobs

- **Queuing system**
  - Launch job processes
  - fork is default
Gatekeeper

- **Accept requests from clients**
  - Authenticate the client based on GSI
  - Authorization using grid-mapfile
- **Invoke a job manager under the user’s permission**
  - The job manager will launch job processes
- **Usually, the gatekeeper is invoked through inetd/xinetd.**
**Mapping from Global ID to a local account**

```
"/C=US/O=Globus/O=Electrotechnical Laboratory/CN=Yoshio Tanaka" yoshio
"/C=JP/O=AIST GTRC/CN=Sugree Phatanapherom/Email=sugree@ku.ac.th" sugree
"/C=JP/O=AIST GTRC/CN=Hidemoto Nakada/Email=hide@aist.go.jp" nakada
"/O=Grid/O=Globus/OU=apgrid.org/CN=Osamu Tatebe" tatebe
"/O=Grid/O=Globus/OU=a02.aist.go.jp/CN=Ninf Demonstrator" ninfdemo
......
```

**The entry of the user have to be appeared on grid-mapfile for launching jobs.**
Job Manager

- **Launch job processes**
- **monitors the processes**
  - Reports job status to the client
    - Callback function is automatically called
    - QUEUED, ACTIVE, FAILED, DONE
  - Control job processes
    - Canceling jobs
- **Abstract interface to queuing systems**
  - Provides same interface to different queuing systems.
  - Fork, PBS, Condor, LSF, SGE...
GRAM: Summary

Job request

Job control, monitor

Error notification

gatekeeper

Job-manager

Queuing system

process

process
Difference with SSH

- **Single sign on by GSI**
  - Enables delegation
  - Users may not care local user name of each host
- **Enable utilization of local schedulers at the server**
- **Flexible remote job control**
  - Monitoring, cancellation
- **Stream connection between a client and a server is not established.**
  - Job manager connects to a client using Globus-IO
Data Management
GridFTP, GASS
1. Data Transport and Access
   - **Common protocol**
     - Secure, efficient, flexible, extensible data movement
   - **Many tools already support this protocol**

2. Replica Management Architecture
   - **Simple scheme for managing:**
     - multiple copies of files
     - collections of files
The GridFTP Tools

- **Patches to existing FTP code**
  - GS!-enabled versions of existing FTP client and server, for high-quality production code

- **Custom-developed libraries**
  - Implement full GridFTP protocol, targeting custom use, high-performance

- **Custom-developed tools**
  - Servers and clients with specialized functionality and performance
Globus Access to Secondary Storage (GASS)

- Services for file and executable staging and I/O redirection that work well with GRAM.
- GASS uses GSI-enabled HTTP as the protocol for data transfer, and a caching algorithm for copying data when necessary.
- The `globus_gass`, `globus_gass_transfer`, and `globus_gass_cache` APIs provide access to these capabilities.
main( ) {
    fd = globus_gass_open(...)
    ... 
    read(fd,...)
    ... 
    globus_gass_close(fd)
}

(a) GASS file access API

(b) RSL extensions

(c) Remote cache management

% globus-gass-cache

(d) Low-level APIs for customizing cache & GASS server
How the Globus Toolkit is used
Two examples

- **ApGrid Testbed**
  - GT2 is used as a common software infrastructure
    - Resource Management
    - Security Infrastructure
    - Information Service
  - mds.apgrid.org

- **GridRPC System Ninf-G**
  - Developed using Globus C and Java APIs
  - GSI, MDS, GRAM, GASS, Globus-IO
ApGrid Testbed

- Experimental Testbed
  - Have experience and knowledge on running VO
  - Evaluation of Grid systems
  - Running Applications
- Donation (Contribution) based
- Approximately 300 CPUs from 10 organizations (as of Feb. 2003)
- Software Infrastructure
  - GSI based security infra.
  - MDS is used for information service
- GT2 is used as a common software
How the GT2 is used in the ApGrid Testbed?

- **Security is based on GSI**
  - every resource (users, computers, ...) must be issued her/his/its certificate by a trusted CA.
  - Job processes should be launched through GRAM
  - User have to generate a proxy certificate prior to submit jobs.

- **The ApGrid Testbed provides hierarchical MDS tree**

- **GT2 must be installed on both server and client machines**
Several institutions have been running their own CAs.
- AIST GTRC, KISTI, Osaka U., TITECH, HKU, ...

The ApGrid Testbed allows multiple root CAs.
- Trust each other’s CA. Trusted CAs should be selected by each institution.
- Trusted CA’s certificate and signing policy should be put in /etc/grid-security/certificates/ directory.

Certificates, signing policy files and optional tools of the ApGrid CAs is put on the ApGrid home page and can be downloaded via https access (signed by VeriSign).

Most institutions trust Globus CA. But it should be invalidated in the near future.
The ApGrid Testbed provides hierarchical MDS tree
Requirements for Users
- how to use the ApGrid Testbed -

● Obtain a certificate issued by one or more of ApGrid trusted CAs.
  - e.g. AIST GTRC CA provides tools (set of scripts) and related documents for generating a CSR. Web interface for this procedure is implemented using JavaWebStart and available on the web page.

● Ask the technical contact of each institution to make an account and put an entry to grid-mapfile on each target machine. The required procedures for making accounts depend on institutions.
  - e.g. AIST GTRC requests a user to provide his certificate and ssh public key for giving an account for him.
1. **Install GT2 on appropriate resources.**
   - Installation from source bundles are recommended.
   - *Setting up and configuration of the server*
     - gsi-setup
     - hostcert
     - gatekeeper/jobmanager
   - *ApGrid Software Package is going to be ready*
     - GPT 2.2.5, GT 2.2.4, MPICH1.2.5.1, Ninf-G 1.1, SCMSWeb 2.1, Iperf 1.6.5

2. **Put trusted CA’s certificates and signing policy files in /etc/grid-security/certificates/directory.**
   - They can be downloaded from ApGrid web site.
3. **Configure GRIS/GIIS to register information to mds.apgrid.org.**

- **Add the following contents into**
  
  ```
  # for default MDS 2.1 install
  # register this server KOUME to ApGrid
  dn: Mds-Vo-Op-name=register, Mds-Vo-name=ApGrid, o=Grid
  regtype: mdsreg2
  reghn: mds.apgrid.org
  regport: 2135
  regperiod: 600
  type: ldap
  hn: koume.hpcc.jp
  port: 2135
  rootdn: Mds-Vo-name=KOUME, o=Grid
  ...
  ```
(cont'd)

- how to contribute to the ApGrid Testbed

4. Usual administrative work
   - Make accounts for users
   - Add entries of the users to grid-mapfile
   - Check logs and monitor resource utilization
   - ...

What is Ninf-G?

- A software package which supports programming and execution of Grid applications using GridRPC.

- Ninf-G includes
  - C/C++, Java APIs, libraries for software development
  - IDL compiler for stub generation
  - Shell scripts to
    - compile client program
    - build and publish remote libraries
  - sample programs
  - manual documents
A simple RPC-based programming model for the Grid

- Utilize remote resources
- Clients make calls with data to be computed
- Task Parallelism (synchronous and asynchronous calls)

Key property: EASE OF USE
• **Some characteristics**
  - Very simple RPC API
  - Existing libs and apps into Grid components
  - IDL embodying call info, minimal client-side management

```c
double A[n][n], B[n][n], C[n][n]; /* Data Decl.*/
dmmul(n, A, B, C); /* Call local function*/
grpc_function_handle_default(&hdl, "dmmul");
grpc_call(hdl, n, A, B, C); /* Call server side routine*/
```
Sample Architecture and Protocol of GridRPC System - Ninf -

Client side

- Call remote library
  - Retrieve interface information
  - Invoke Remote Library Executable
  - It Calls back to the client

Server side

- Server side setup
  - Build Remote Library Executable
  - Register it to the Ninf Server

IDL file
Numerical Library

IDL Compiler
Generate

Remote Library Executable

Client

Interface Request

Interface Reply

Ninf Server

Connect back

Invoke Executable

fork
Architecture of Ninf-G

Client side

- Interface Request
- Interface Reply
- Invoke Executable
- Connect back

Server side

- IDL file
- Numerical Library
- IDL Compiler
- Generate
  - Remote Library Executable
  - Interface Information
    - LDIF File

- GRAM
- MDS
- Fork
- Retrieve
Ninf-G: how the Globus Toolkit is used?

- Ninf-G is implemented using Globus APIs (C and Java)
- Stub information is retrieved from MDS
- Ninf-G executable will be invoked through Globus gatekeeper
  - GSI based authentication/authorization between clients and servers
- Callback from Ninf-G executables to clients uses Globus-IO.
- GASS is used for file transfer.
- Ninf-G installation requires to add resource provider to MDS (shell script is provided).
Demonstration

Ninf-G Applications on the ApGrid Testbed
Goal
- Long term, global weather prediction
  - Winding of Jet-Stream
  - Blocking phenomenon of high atmospheric pressure

Barotropic S-Model
- Weather forecasting model proposed by Prof. Tanaka
- Simple and precise
  - Modeling complicated 3D turbulence as a horizontal one
    - 200 sec for 100-days prediction/ 1 simulation (Pentium III machine)
- Keep high precision over long periods
  - Taking a statistical ensemble mean
    - ~50 simulations
  - Introducing perturbation at every time step
  - Typical parameter survey

Gridifying the program enables quick response
Ninfty the Program

- Dividing a program into two parts as a client-server system
  - **Client:**
    - Pre-processing: reading input data
    - Post-processing: averaging results of ensembles, visualizing results
  - **Server**
    - Weather forecasting simulation

- Integrating the system by using grid-middle ware
  - Ninfg library: communication between C/S
  - Grid lib portal: easy and secure access from a user’s machine
APGrid

Weather forecast
Server program
64cpu Cluster
AIST, Japan

Weather forecast
Server program
64cpu Cluster
KISTI, Korea

Weather forecast
Server program
22cpu Cluster
KU, Thailand

Call Remote
server program
using Ninf-G

client auth.

signOn/signOff
Job Control
submission/query
/cancel

HTTP server
+ Servlet
(Apache +
Tomcat)

JDBC Interface
(TCP/IP)

Accounting
DB (Postgress)

Job Queuing
Manager &
Signing Server

SignOn/SignOff
Accounting
information

Weather forecast
client program

user

GO TO DEMO!
Summary

- **Globus Toolkit™**: APIs, SDKs, and tools which implement Grid protocols & services
  - Provides basic software infrastructure for suite of tools addressing the programming problem
- **GT2 is defacto standard as Grid tools.**
  - Build Grid testbed
  - Develop Grid middleware
Acknowledgement

- **Globus Team**
  - Steve Tuecke, Liming Lee, ...

- **ApGrid participants, especially for**
  - KISTI (Sangwan Kim, ...)
  - KU (Sugree Phatanapherom, ...)
  - SDSC (Mason Katz, ...)

- **Ninf users**