Tutorial 1: Introduction to Globus Toolkit

Prof. Richard Sinnott
Dr. John Watt
The Globus Project (1995)

- Consortium dedicated to collaborative design, development, testing and support of Globus
  - U.S. Argonne National Laboratory
  - University of Southern California/Information Sciences Institute
  - University of Chicago

- New members form an international consortium
  - Swedish Centre for Parallel Computers (PDC)
  - Edinburgh Parallel Computing Centre (EPCC)
- Includes Academic Affiliates program with participation from Asia-Pacific, Europe and US
- US Federal sponsorship
  - NASA, DoE, NSF, DARPA
- Industry sponsorship
  - IBM, Microsoft Research
The Globus Toolkit (1998-)

- An open-architecture, open-source set of software services and libraries that support computational grids.
- Components can be used independently or together to develop useful grid applications.

“the de facto standard for grid computing”

- New York Times
Accolades

- R&D Magazine R&D100 Award: "Most Promising New Technology”, 2002

Globus Toolkit timeline

- **GT1 (1998)**
  - GRAM and MDS

- **GT2 (2001)**
  - GridFTP, The Grid Packaging Toolkit (GPT)

- **GT3 (2002)**
  - Implementation of the Open Grid Services Architecture (OGSA)

- **GT4 (2005)**
  - Implementation of the Web Services Resource Framework (WSRF)
- We will work with GT3.3
  - Allows an introduction to Grid Service implementation with stable, production standard software
  - Is compatible with other Grid apps (Condor, PERMIS)
- Implements OGSA not WSRF
  - WSRF is the (current) goal for Web Services standardisation
  - Learning OGSA is a good introduction to WS
- Implementation of the two is very similar
OGSA: GT3 Overview

- GT3 Core
  - Container, APIs, Runtime env
- GT3 Security
  - SOAP, GSI
- GT3 Base
  - GRAM, MDS, RFT
- GT3 Data & Other
  - GridFTP, RLS...
Word to the wise!

- Globus is heavily dependent on the user environment
  - Every time you login or su as a different user you will have to make sure the environment is set correctly.
  - We have put the necessary declarations in your ~/.bash_profile file in your home directory
  - It's best to source this each time you login, even if you suspect it has picked it up automatically
    
    ```bash
    $ source ~/.bash_profile
    ```
  - If all your commands don’t work, it highly likely that the environment is wrong!
Environment check

- In your home directory for ‘you’ and ‘globus’, check that `.bash_profile` contains:
  
  ```bash
  source $GLOBUS_LOCATION/etc/globus-user-env.sh
  source $GLOBUS_LOCATION/etc/globus-devel-env.sh
  ```

- For account ‘you’ make sure
  
  ```bash
  export X509_USER_PROXY=/tmp/x509up_u502 (or 503)
  ```

- For account ‘globus’ make sure
  
  ```bash
  export X509_USER_PROXY=/tmp/x509cp_globus_grim
  ```
Let's get started...

- su to the user account “globus”, using the password provided and type
  
  $ cd $GLOBUS_LOCATION

- Type the following command:
  
  $ grid-proxy-init

- Type the following command:
  
  $ globus-start-container
What is going on?

- Globus runs in a user account “globus”
  - GT3 services accepting network connections run in a low-privilege local account.
  - Compromised network service can only return a ‘denial of service’
  - Two programs (grim and setuid_starter) run as root so GT3 services may access important local services
    - These programs are very tightly constrained in their capabilities
    - An attacker wouldn’t be able to do much with them!

This improves on the security model of previous Globus versions…
What is going on?

- grid-proxy-init creates a *proxy certificate*
  - Identity is established and propagated on the grid by X509 digital certificates
    - Certificate ties an identity (subject) to a public key
    - Issued and digitally signed by a central Certificate Authority
  - A proxy certificate is a short-lifetime, temporary certificate that is created on a grid resource
    - It is issued and signed by its base certificate (not a central authority)
    - It can create more proxy certificates if required
    - Short lifetime allows a compromised certificate to be time-limited in the damage it can do
What is going on?

- **globus-start-container**
  - This command initiates the Java Web Service container that Grid services deploy into
    - It runs as the low-privilege user 'globus'
    - Typically you will leave this window alone once active, and submit your jobs as a different user in a new shell.
  - **You will see a list of pre-deployed base services**
    - Some are small applications like Google search or Weather
    - Others are more fundamental to job submission
      - The Master Managed Job Factory Service (MMJFS)
Writing A Grid Service

You will have been introduced to Web Service concepts in the “Web Services” lecture

- A WSDL document describes the service interface
- Describes the operations a Web Service provides
- Is often known as the “portType”

Can be written in two ways

- Writing the WSDL directly (versatile but complex)
- Creating the WSDL from a Java interface
  
  ▶ NB: Actually, we need to create a GWSDL file for Grid Services, but we can use ANT to convert it for us! More later…
Writing A Grid Service

- The next step is to implement the service
  - Java class which implements the operations defined in the (G)WSDL interface (portType)
  - Must contain:
    - Globus package declarations
    - An extension from the ‘skeleton’ class “GridServiceImpl”
    - An implementation of the GWSDL portType (created on final compilation, and known as a ‘stub’)
  - Everything else in this file will be your coding of the implementation of the original portType.
Writing A Grid Service

- Finally, we need a file which describes how this Grid Service will appear to the outside world
- i.e. GSH (Grid Service Handle)
  - This is what appears in the list when you run the Globus container
- The file is written in WSDD format
  - Web Service Deployment Descriptor
Compiling Your Grid Service

- We need now to unify all these files and compile them into a final deployable ‘gar’ file.
- We need to
  - Convert our WSDL file into a GWSDL document
  - Create (and compile) the ‘stub’ classes (linking the interface and the implementation)
  - Compile the service implementation
- Help is at hand…
Ant

- Ant is a Java build tool (like ‘make’)
- Ant can combine all the previous documents using a ‘build.xml’ file to create our final ‘gar’
- A script will be provided which will write the build.xml file for you…
Help

- GT3 is very complicated!
  - OGSI specification draft ran to over 90 pages
  - Requires a stable, known platform with very specific supporting application versions
- User base was generally supported through the globus mailing list.
- Globus have now published the famous (in Grid terms at least!) Sotomayor GT3 Tutorial
  - THE world resource for Globus Toolkit education
  - [http://gdp.globus.org/gt3tutorial](http://gdp.globus.org/gt3tutorial)
Exercise

- Problem Set 3 involves implementing one of the tutorial examples.
- Provides a complete description and justification of each step in the build process
- Provides the Ant build script (tutorial_sh)

Questions:
- John Watt
- Email: jwatt@dcs.gla.ac.uk