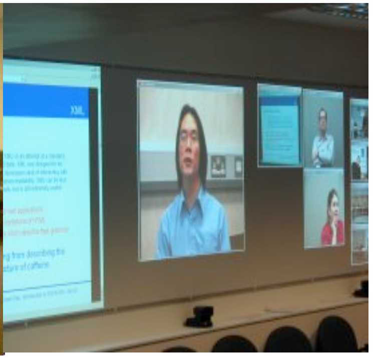
Tutorial 1:
Introduction to Globus Toolkit

John Watt, National e-Science Centre

National e-Science Centre

- **Kelvin Hub**
  - **Opened May 2003**
  - **Kelvin Building**
  - **Staff**
    - Technical Director
      - Prof. Richard Sinnott
    - 6 RAs
    - 2 Ph.D students
    - ScotGRID Support
      - David Martin
  - **Training Laboratory**
    - 20 Workstations
  - **Access Grid Node**



Training Lab

- Open 10am-4pm weekdays
  - NO out-of-hours access to room (Physics Dept. swipe-card needed)
  - If lab door is closed, knock on 246D or 246C…
  - NB. Machines will be accessible via SSH
- You will be assigned one terminal
  - MUST use only this terminal for the rest of the course
  - Two accounts: '*username*' and 'globus'
    - Password for low-permission account 'globus' is:
      - gr1dw0rk (3rd last char is a zero)
    - Password for '*username*' distributed privately

Tutorial Prerequisites

- Java
  - All our GT4 services are written in Java
  - Java proficiency is essential
- Web Services
  - Basic knowledge (Femi's lecture)
- UNIX
  - Experience with UNIX operating system required
  - File manipulation, compliers, environments etc..
- Everything else is assumed new…

Tutorials Timetable

Week	Day/Time	Topic	Staff
3	Fri 11am	Introduction to Globus	J.W.
4	Fri 11am	Globus Development	all
5	Fri 11am	Globus Development	all
6	Fri 11am	Condor	J.W.
7	Tue 12pm	SAML/PERMIS (L)	A.S.
7	Wed 12pm	Portals (L)	J.J.
7	Fri 11am	Q & A Session	all
8	Fri 11am	OGSA-DAI (L)	O.A.
10	Tue 12pm	Example Systems (L)	R.S.
10	Fri 11am	Assignment Demos	R.S.

Support

- Initial queries to John Watt (email only)
  - j.watt@nesc.gla.ac.uk
  - Subject: “GC5 Tutorial – <*Question*>”

- All NeSC staff will be available during office hours to help with any middleware issues

the globus alliance
www.globus.org

- **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

- The Globus Alliance (2003)
  - 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™
www.globustoolkit.org

- **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**
  - **MIT Technology Review: "Ten Technologies That Will Change the World" ,2003**

the globus toolkit™
www.globustoolkit.org

National e-Science Centre

- **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)

Before we start

- GT4 is 250MB of binaries and configuration files
  - Available free from http://www.globus.org
  - A fresh installation runs over 50 services
    - Typically takes around 5 hours on a Celeron!
    - Requires Java SDK, Apache Ant, gcc, make
      - Restrictive version requirements, a pain…
  - Some services work better than others
    - We will avoid 'bleeding edge' services like information/directory services and fancy service features like delegation
    - You will occasionally see errors from these pop up in your container
  - We will be looking at a small subset of services
    - Yet these allow us to deploy any service we can implement

Before we start…

- Globus is heavily dependent on the user environment/classpath etc.
  - 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
  - Its best to source this each time you login, even if you suspect it has picked it up automatically

```
$ source ~/.bash_profile
```

  - If all your commands don't work, it highly likely that the environment is wrong!

Accounts

- You require two user accounts to develop with Globus
  - Account ‘globus’
    - The account Globus was installed under
    - This account runs the Web Service container
    - Deploy your services with this account
  - Account ‘*username*’ (in this tutorial ‘testuser’)
    - The account you will run your GT4 clients under
    - Write your services with this account

  - Container must be running for Web Services to be invoked!

Container

- Login as ‘testuser’ (password ‘testuser’)
- Open a terminal window

```
$ su globus     (gr1dw0rk)
$ cd $GLOBUS_LOCATION
$ source ~/.bash_profile
```

- Type the following command:

```
$ globus-start-container –nosec
```

  - -nosec means we don’t want to use secure transport
    - Makes learning a little easier (Secure Message is still used)

Container

- You should see a list of deployed services:
  - [1]: http://localhost:8080/wsrf/services/TriggerFactoryService
  - [2]: http://localhost:8080/wsrf/services/DelegationTestService
  - [3]: http://localhost:8080/wsrf/services/SecureCounterService
  - …………
  - The URL displayed is the Web Service URI
    - This is the URI to call when you wish to invoke a Web Service
    - You can visually check any service you have deployed has been initialised successfully
- Do a `<Ctrl-C>` to stop the container
  - Ignore this error!
    - You will get this one every 60 seconds:

    -
      ```
      2007-01-25 10:13:20,808 ERROR impl.QueryAggregatorSource [Thread-
      12,pollGetMultiple:149]…
      ```

User

- Open another terminal (just as ‘testuser’)
- Type

```
$ source ~/.bash_profile
$ grid-proxy-init
```

- You should see the following output:

```
Your identity: /C=UK/O=Grid/O=Training/OU=GUGridComputingCourse/CN=User00
Creating proxy ............................................................ Done
Your proxy is valid until: Thu Jan 31 22:58:35 2007
```

User

-

- 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
    - grid-proxy-destroy removes your proxy

‘HelloWorld’

- We will now run through a ‘HelloWorld’ example
  - Create a ‘HelloWorld’ Grid Service
  - Deploy this service in GT4
  - Create a client to invoke the service
    - This exercise is to check your Globus environment is OK

- You will find prewritten files in
  - **`$TUT_DIR=/home/testuser/helloworld`**

Service portType

- First step is to define what your service will provide to the outside world
  - This is called the ‘service interface’ or ***portType***
  - Defines the *operations* available to service clients
  - Written in standard Web Services Description Language (WSDL)
- Our HelloWorld service WSDL file is:

```
$TUT_DIR/schema/HelloWorldService/HelloWorld.wsdl
```

Service Implementation

- **Next, we describe the inner workings of the service**
  - This is the implementation of the portType
  - Defines *how* the service performs the operations the portType says it provides
  - Written in Java

- **Our HelloWorld service implementation files:**

```
$TUT_DIR/org/globus/services/HelloWorld/impl/*
```

Service Deployment

- A client needs to know how to invoke your service
  - This is done by creating a ‘deployment descriptor’
  - Defines how the service is published to the outside world
  - Written in Web Service Deployment Descriptor (WSDD) format

- Our HelloWorld service deployment descriptor:

```
$TUT_DIR/org/globus/services/HelloWorld/*
```

Service Creation

- We have all the pieces for our service, but they are all separate (and not compiled!)
  - GT4 requires our service to be presented as a *grid archive* (.gar) - much like a .jar
  - Apache Ant can do all of this for us, provided we pass it the appropriate build files
  - You will be provided with generic files which can build almost any service with minimal changes
- Our HelloWorld build files:

```
$TUT_DIR/tutorial_build.sh, build.xml
```

Activate

- Build your service (in ‘testuser’ terminal)
  - Move to `/home/testuser/helloworld`
  - Run the following command:

```
$ ./tutorial_build.sh –d org/globus/services/HelloWorld
  –s schema/HelloWorldService/HelloWorld.wsdl
```

  - Locate the .gar file it created
  - Copy it to `$GLOBUS_LOCATION`
    - You have write permission in this directory

Activate

- Deploy the service (in ‘globus’ terminal)
  - Move to $GLOBUS_LOCATION
  - Run the following command:

```
$ globus-deploy-gar \
  org_globus_services_HelloWorld.gar
```

  - Start the GT4 container (no security)

```
$ globus-start-container -nosec
```

  - Find your service in the list
    - Should be near the top!

Activate

- Compile the client (back to ‘testuser’ terminal)
  - Move to /home/testuser/helloworld
  - Compile the client (and EPR – do this first)

```
$ javac –classpath \
  ./build/stubs/classes/:$CLASSPATH \
  org/globus/clients/HelloWorld/CreateEPR.java
$ javac –classpath \
  ./build/stubs/classes/:$CLASSPATH \
  org/globus/clients/HelloWorld/Client.java
```

  - Don’t worry about the EPR client, we will explore this next week…

Activate

- Run the client (as ‘testuser’):

```
$ java –classpath \
   ./build/stubs/classes/:$CLASSPATH \
   org.globus.clients.HelloWorld.CreateEPR \
   http://localhost:8080/wsrf/services/HelloWorldServ
   ice
```

```
$ java –classpath \
   ./build/stubs/classes/:$CLASSPATH \
   org.globus.clients.HelloWorld.Client \
   http://localhost:8080/wsrf/services/HelloWorldServ
   ice string
```

Success?

- You should get your string returned back to you (twice!)

- Stop the container (as globus) with a <Ctrl-C>
- You can remove your HelloWorld service with:

```
$ globus-undeploy-gar \
  org_globus_services_HelloWorld
```

- (no .gar at end)

Next Week

- A closer look at what we did today…
- From tonight the ‘testuser’ account will be disabled.
  - All work will be done with your ‘username’ account
  - Pick up your login details if you haven’t already done so.
  - The machine you have been sitting at today is now the machine you will work from
    - External access from sibu.dcs.gla.ac.uk

```
$ ssh username@labpc-X.nesc.gla.ac.uk
```