



# GT4 installation & configuration handson

Gabor Kecskemeti

- MTA SZTAKI, Hungary
- Univ. Westminster, UK  
kecskemeti@sztaki.hu

# Prerequisites



- Basic Debian Sarge with these important packages installed:
  - gcc, make, zlib, java 1.5.0, ant 1.6.5 - GT4 wide dependency
  - sudo - WS-GRAM dependency
  - PostgreSQL - RFT dependency
  - Xinetd - GridFTP dependency
- A production machine has to be synced with a proper time server - use ntpdate!
- The root password on your machines is:  
master07Gc

# Prerequisites setup as **root**



- Installing the Sun JDK 1.5:
  - `apt-get install sun-java5-jdk`
  - Java home is in `/usr`
- Installing the current release of Apache Ant:
  - `wget`  
`http://neacm.fe.up.pt/pub/apache/ant/binaries/apache-ant-1.7.0-bin.tar.gz`
  - `tar -xzf apache-ant-1.7.0-bin.tar.gz`
  - `mv apache-ant-1.7.0 /usr/local/ant`

# User environment preparation as **root**



- Skeleton setup for all the users:
- Create the globus installation dir:
  - `mkdir /usr/local/globus`
- Extend the skeleton bashrc (edit `/etc/skel/.bashrc`)
  - `export JAVA_HOME=/usr`
  - `export ANT_HOME=/usr/local/ant`
  - `export GLOBUS_LOCATION=/usr/local/globus`
  - `export GPT_INSTALL_LOCATION=$GLOBUS_LOCATION`
  - `export PATH=$ANT_HOME/bin:$PATH`
  - `if [ -f $GLOBUS_LOCATION/etc/globus-user-env.sh ]`
  - `then`
  - `. $GLOBUS_LOCATION/etc/globus-user-env.sh`
  - `fi`

# Host setup as **root**



- Execute: “**hostname -f**”
  - Write down the results
- Execute: “**ifconfig**”
  - Write down the results - look for the inet addresses of the ethx (usually 0) interface - exclude the lo interface.
- Check the contents of the /etc/hosts
  - There should be an entry with the IP listed in the ifconfig’s output and an FQDN printed with the hostname command
  - If the entry is not present add this line: Ipaddress FQDN
  - Then execute **hostname FQDN** to update the hostname of the current session

# Example hosts file entries:



- 127.0.0.1 localhost B02AU0743
- 192.168.33.43 B02AU0743.fe.up.pt  
B02AU0743
- B02AU07XX.fe.up.pt is the scheme of your  
hostname

# GSI CA setup as **root**



- `mkdir -p /etc/grid-security/certificates`
- Install the ROOT CAs with yesterday's accounts on the SZTAKI site:
  - GEMMLCA CA - host certs:
    - `scp userXX@n34.hpcc.sztaki.hu:/etc/grid-security/certificates/*f7d25a56* /etc/grid-security/certificates`
  - UK Test CA - user certs:
    - `scp userXX@n34.hpcc.sztaki.hu:/etc/grid-security/certificates/*cb398b31* /etc/grid-security/certificates`
- For production use the each installed CA's CRL has to be updated regularly with a tool like `fetchcrl`.

# GSI host setup as **root**



- Download the porto-host-certificates.zip from the agenda page next to this ppt.
- `unzip porto-host-certificates.zip`
- `cd /etc/grid-security`
- `cp porto2/YOURFQDN/* .`
- `cp hostcert.pem containercert.pem`
- `cp hostkey.pem containerkey.pem`



# Basic container setup as **root**



- Setting up the container's location:
  - “`adduser globus`” - fill in the details - use globus as a pass
  - `chown globus:globus /usr/local/globus`
- Modify the access rights in `/etc/grid-security`:
  - `cd /etc/grid-security`
  - `chmod 400 *key.pem; chmod 644 *cert.pem`
  - `chown globus:globus container*.pem`
  - Without these certificates the globus will not install properly
  - If you acquire these certificates later please run the following script before starting the globus:
    - `/usr/local/globus/setup/globus/setup-globus-gram-job-manager`

# Aquiring and installing the base toolkit as **globus**



- Acquire the current GT4 release for debian sarge:
  - `wget http://www-unix.globus.org/ftppub/gt4/4.0/4.0.3/installers/bin/gt4.0.3-x86_deb_3.1-installer.tar.gz`
- Unpack the gzipped tarball:
  - `tar -xzf gt4.0.3-x86_deb_3.1-installer.tar.gz`
- Execute the configure script for globus from the source tree:
  - `cd gt4.0.3-x86_deb_3.1-installer`
  - `./configure --prefix=$GLOBUS_LOCATION`
- Build and install the GT4 sit back and relax for a few minutes:
  - `make`
  - `make install`



# GT4 Configuration

# Configuring GridFTP - prerequisite of RFT - as **root**



- Port setup:
  - `echo gsiftp 2811/tcp >> /etc/services`
- Create the `/etc/xinetd.d/gridftp` config file:

```
service gsiftp
{
    instances      = 100
    id              = gridftp
    socket_type    = stream
    protocol       = tcp
    user           = root
    wait           = no
    env += GLOBUS_LOCATION=/usr/local/globus
    env += LD_LIBRARY_PATH=/usr/local/globus/lib
    env += GLOBUS_TCP_PORT_RANGE=20000,25000
    server = /usr/local/globus/sbin/globus-gridftp-server
    server_args = -l
    log_on_success += DURATION
    nice = 10
    disable = no
}
```

- Restart xinetd: `/etc/init.d/xinetd restart`



# Reliable File Transfer setup as **root**

- Edit this file next to the database -  
`/var/lib/postgresql/7.4/main/pg_hba.conf`:
  - Replace all “ident” entries to “trust”
  - Append the following line at the end:
    - `host[tab]rftDatabase[tab]globus[tab>YourIP[tab]255.255.255.255[tab]trust`
- Edit the configuration file next to the `pg_hba.conf` - called `postgresql.conf` - uncomment the line with `tcpip_socket` and enable it with changing the default value of “false” to “true”
- Restart the server: `/etc/init.d/postgresql-7.4 restart`

# Reliable File Transfer setup as postgres



- When installing PostgreSQL in debian the package automatically initializes the dbms. In the cases when it is missing use initdb.
- Add the globus database user:
- `createuser globus`

# Reliable File Transfer setup as **globus**



- Setting up the rftDatabase
  - `createdb rftDatabase`
  - `psql -d rftDatabase -f /usr/local/globus/share/globus_wsrf_rft/rft_schema.sql`
- Preparation of the RFT's JNDI config:
  - `/usr/local/globus/etc/globus_wsrf_rft/jndi-config.xml`
  - Erase the contents of this element:  
`/jndiConfig/service[@name="ReliableFileTransferService"]/resource[@name="dbConfiguration"]/parameter[contains(name/text(),"password")]/value`

# Creating Your first grid user as **griduser**



- `adduser griduser - as root - pass should be griduser`
- Create `.globus` folder and copy your user certs -  
usercert and userkey.pem:
  - `scp -r userXX@n34.hpcc.sztaki.hu:.globus ~`
  - `chmod 400 *key.pem; chmod 644 *.cert.pem`
- `ls -al ~/.globus`
  - `-r----- 1 sipos sipos 963 Dec 21 23:28 userkey.pem`
  - `-rw-r--r-- 1 sipos sipos 2718 Dec 21 23:31 usercert.pem`



# Checking your identity as **griduser**



- Your grid identity:
  - `grid-proxy-init`
    - Password - userXX
  - `grid-proxy-info`
    - Check the identity line:
    - identity : /O=Grid/OU=GlobusTest/OU=simpleCA-gt4.irt.vein.hu/OU=irt.vein.hu/CN=Sipos Gergely
- Your local identity :
  - `id`
    - uid=1002(**sipos**)...

# Creating Your first grid user as **root**



- Add your identity to the gridmapfile:
  - export GLOBUS\_LOCATION=/usr/local/globus
  - . \$GLOBUS\_LOCATION/etc/globus-user-env.sh
  - grid-mapfile-add-entry -dn  
/O=Grid/OU=GlobusTest/OU=simpleCA-  
gt4.irt.vein.hu/OU=irt.vein.hu/CN=Sipos Gergely  
-ln sipos

# One time WS-GRAM/sudoers setup as **root**



- Add the following to the end of the /etc/sudoers:
  - Runas\_Alias GLOBUSUSERS = **ALL, !root**
  - globus ALL=(GLOBUSUSERS) NOPASSWD:  
/usr/local/globus/libexec/globus-gridmap-and-  
execute -g /etc/grid-security/grid-mapfile  
/usr/local/globus/libexec/globus-job-manager-  
script.pl \*
  - globus ALL=(GLOBUSUSERS) NOPASSWD:  
/usr/local/globus/libexec/globus-gridmap-and-  
execute -g /etc/grid-security/grid-mapfile  
/usr/local/globus/libexec/globus-gram-local-proxy-  
tool \*

# Start the grid service container as **globus**



- globus-start-container-detached

# Notes for production use



- PostgreSQL config should be tuned for safety
- Host, and Container Certs should be issued by a real CA
- Each GT4 service can be installed on a dedicated machine
- When the WS-GRAM sits on a different host than the GridFTP do not forget to update this file:
- `/usr/local/globus/etc/gram-service/globus_gram_fs_map_config.xml`
- Sudoers and gridmap files might be maintained automatically

# Notes for production use - MDS4 edition



- MDS can be configured to publish its ServiceGroup to other MDS4 services or collect some other MDS4 sources by upstream and downstream urls.
- Examples (from `$GLOBUS_LOCATION/etc/globus_wsrf_mds_index/hierarchy.xml`):
  - `<upstream>https://n34.hpcc.sztaki.hu:8443/wsrf/services/DefaultIndexService</upstream>`
  - `<downstream>https://grid-compute-ws.cpc.wmin.ac.uk:8443/wsrf/services/DefaultIndexService</downstream>`



Enjoy!

Some example usage of our local  
installation



# GridFTP copy examples

- Command line tool:
- `globus-url-copy`
  - Parameters:
    - `-tcp-bs` buffersize
    - `-p` parallelism
    - `Sourceurl desturl` - supported protocols: `https,http,gsiftp,ftp,file`
- Examples:
  - `globus-url-copy gsiftp://localhost/etc/xinetd.d/gridftp  
file:///tmp/gridftp_copy`
  - `globus-url-copy file:///tmp/gridftp_copy  
gsiftp://localhost//tmp/gridftp_copy_2`



# RFT and GridFTP cooperation



- `cp /usr/local/globus/share/globus_wsrf_rft_client/transfer.xfr /tmp/mytransfer.xfr`
- Modify the `mytransfer.xfr`:
  - Replace the subject names of the gridftp servers to ours - the lines containing “Ravi Madduri” - use the one provided by this command:
    - `grid-cert-info -file /etc/grid-security/containercert.pem -subject`
  - replace localhost to your FQDN (drop the 5678)
- `echo hello world > /tmp/rftTest.tmp`
- `rft -h <FQDN> -f /tmp/mytransfer.xfr`

# WS-GRAM job submit



- Interactive:
  - `globusrun-ws -submit -F JobFactoryURL -Ft FactoryType -s -c command ...`
- Batch:
  - `globusrun-ws -submit -batch -F JobFactoryURL -Ft FactoryType -o EPRFile -c command`
  - `globusrun-ws -status -job-epr-file EPRFile`
- Test:
  - `globusrun-ws -submit -s -c /bin/hostname`

# Simple MDS4 query



- `wsrp-query -z none -s https://<FQDN>:8443/wsrp/services/Default IndexService "//*[local-name()='GLUECE']"`