

Oxford NGS installation How-to

GridSAM Version 2.1.4

Document version: 0.3

Author: Matteo Turilli, matteo.turilli@oerc.ox.ac.uk

Document History

Version 0.1, 30 Jun 2009. Author Matteo Turilli

Version 0.2, 07 Jul 2009. Author Matteo Turilli

Version 0.3, 09 Jul 2009. Author Matteo Turilli

Conventions

- User dependent inputs are indicated between <> in `Courier New` font face
- Directories, commands, command outputs, user names, passwords and configuration parameters are formatted in `Courier New` font face
- Commands are preceded by a \$ or a # depending on who is issuing the command – a normal user in the former case, root in the latter. All commands are formatted in `Courier New` font face
- <OMIICLIENT> indicates the directory where you have installed the GridSAM client distribution
- <OMII> indicates the directory where you have install the Campus Grid Toolkit (CGT) and then the GridSAM server

What is GridSAM

GridSAM is an open-source job submission and monitoring web service. The aim of GridSAM is to provide a Web Service for submitting and monitoring jobs managed by a variety of Distributed Resource Managers (DRM). The modular design allows third-party to provide submission and file-transfer plug-ins to GridSAM. Moreover the job management API used by the GridSAM web service can be embedded into grid application that requires job submission and monitoring capabilities. GridSAM endorses the WS-I set of web service standards along with the Job Submission Description Language (JSDL) and Open Grid Services Architecture - Basic Execution Service (OGSA-BES) standards developed through the Open Grid Forum (formerly Global Grid Forum).

For more information see: <http://gridsam.sourceforge.net/>

How GridSAM works

GridSAM consists of several subsystems working together to support pluggable job persistence, queuing, job launching, file staging and failure recovery. Each subsystem has a set of well-defined interfaces. They are collectively known as the **Service Provider Interface** (SPI) contrary to the **Application Provider Interface** (API) used by client who demands job submission functionality.

In Figure 1 user Joe Blogg:

1. Creates and upload a user proxy certificate to NGS myProxy server. On a linux platform with the Globus Client commands installed, the typical command would be:

```
$ myproxy-init -s myproxy.grid-support.ac.uk -l your_login_name
```

2. Submits a job to the GridSAM instance. On a Linux machine with the gridsam client commands installed, the typical command would be:

```
$ gridsam-submit -s https://<your_domain>:18443/gridsam/services/gridsam  
-myproxy \  
-myproxyuser your_login_name \  
-myproxyhost myproxy.grid-support.ac.uk \  
-j your_JSDL_script  
$ password: <your_password>
```

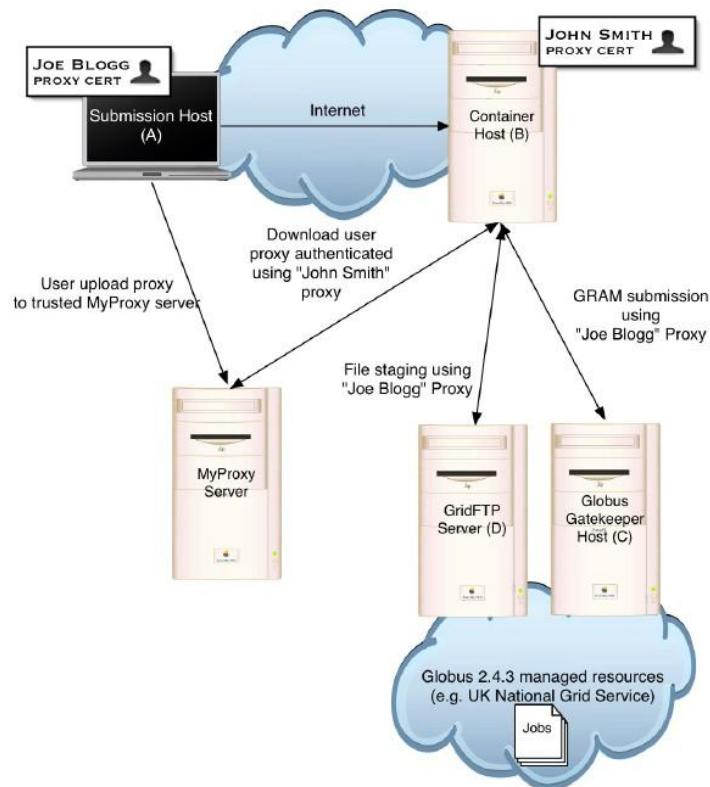


Figure 1 GridSAM architecture

The GridSAM instance (called John Smith in Figure 1) receiving the submission request would:

1. Download the user proxy certificate using:
 - a. Its own (John Smith) certificate to authenticate to the myProxy server.
 - b. The password submitted by the user in order to use her or his proxy certificate
2. Translate the user JSDL submission script into the language used by the chosen submission mechanism (Globus in case of Oxford Gridsam instance)
3. Submit the job to the NGS cluster chosen during the installation of the GridSAM instance
4. Stage-in files via GridFTP if required by the submission script
5. Answer to the status requests when the user issues the command:

```
$ gridsam-status -s  
https://gridsam.oerc.ox.ac.uk:18443/gridsam/services/gridsam -j  
urn:gridsam:<job_ID>
```

For more information see:

<http://gridsam.sourceforge.net/2.1.4/architectureguide.html>

What GridSAM does not do

A GridSAM instance does not:

- Act as a resource broker. Every single GridSAM instance submits jobs to a single NGS resource. So, for example, `gridsam.oerc.ox.ac.uk` submits its jobs only to the `ngs.oerc.ox.ac.uk` cluster.
- Host data for stage-in or out

Prerequisites

Server:

1. An UK e-Science host certificate for the GridSAM instance
2. Scientific Linux 4.7
3. Sun Java 1.6.0, perl, curl, unzip, bind-tuils shipped with SL4.7
4. ntp

Client:

1. An UK e-Science personal certificate
2. Scientific Linux 4.7
3. Sun Java 1.6

Preparation

The Oxford NGS team chose the Campus Grid Toolkit (CGT) container to install GridSAM. CGT is a focused distribution of the OMII Container, which will install GridSAM 2.1.4 (and optionally AHE), and assist with getting up and running your GridSAM instance quickly.

The installation process is divided in **six steps**:

1. Installing CGT with GridSAM server
2. Downloading and installing the GridSAM client.
3. Testing GridSAM server by submitting a job via the client. This job will run on the same machine where the GridSAM server has been installed. The fork jobmanager will be used.
4. Reconfiguring GridSAM server and client to use the UK e-Science certificates.
5. Configure the GridSAM server connector to use Globus in order to submit its jobs to a NGS resource.
6. Testing the final deployment of GridSAM submitting jobs with and without file staging to the chosen NGS resource.

Step 1: Installing CGT 1.1.3 and GridSAM server 2.1.4

For documentation and general installation instructions see:

<http://www.omii.ac.uk/docs/CGT/CampusGridToolkit.pdf>

NGS specific instructions:

1. Create an OMII account on the machine on which you will want to install CGT and GridSAM server.
In this guide the account is `omii`
2. As `omii`, edit `~/.bashrc` and add the line:

```
export JAVA_HOME=/usr/java/default
```

3. Download `campus-grid-toolkit-server-1.1.3.tar` from:
<http://www.omii.ac.uk/wiki/CGT>
4. Unpack the tarball and run the installer script:

```
CampusGridToolkitServerInstall.pl
```

5. The installer script will pose 3 questions. Question 1:

```
Please specify a directory where you would like the server installed
```

```
Enter the path where you want to install CGT. For example: /home/omii/CGT
```

6. Question 2:

```
Would you like your server to use Direct GridSAM or Managed Execution?
```

```
Choose option 1 for a "Direct GridSAM" installation
```

7. Do not worry if you see the message:

```
get-jvm:
  [java] Nothing known about Java version 1.6.0_13 from vendor Sun
  Microsystems Inc.
  [java] This software has not been tested with this version of Java.
  [java]
  [java] Do you want to proceed with the installation anyway? (y,n):
```

```
Just answer y. Our tests showed that java 1.6.0_13 works properly with GridSAM 2.1.4
```

8. Question 3:

```
Which GridSAM connector would you like to use?
```

```
Choose option 1 for using the "Fork connector".
```

9. If the installation is successful you will see the message:

```
** server installation and configuration complete
** server available at https://<your_FQDN>:18443
```

Step 2: Downloading and Installing the Linux GridSAM client

For documentation and general installation instructions see:

<http://www.omii.ac.uk/docs/CGT/CampusGridToolkit.pdf>

NGS specific instructions:

1. From a machine with a Linux OS, point a web browser to `https://<your_domain>:18443` and download the OMII-UK Campus Grid Toolkit Client for UNIX tarball.

2. Untar the tarball and execute the client installation script:

```
CampusGridToolkitClientInstall.sh
```

3. Enter Installation Path (for example /home/<your_user>/bin/campus-grid-toolkit-client)
4. If you see a message similar to:

```
"Nothing known about Java version 1.6.0_13 from vendor Sun Microsystems Inc. This software has not been tested with this version of Java."
```

Answer y

The installation should proceed normally. If not, then you will have to upgrade/change your java installation.

5. If needed, add the information relative to the proxy you use to connect to the Internet
6. Enter the requested details about your machine name and yourself. This information will be used to create a temporary OMII certificate. This certificate will be replaced with your proper UK e-Science certificate in the following configuration steps.
7. If the installation is successful you should see the message:

```
"test-install:
...Done
Campus Grid client installation complete."
```

Step 3: Testing GridSAM server by submitting a job via the client

Test your client/server installation running from your client the script:

```
$ uname_webdav.pl
```

You should find this script in:

```
<OMIICLIENT>/campus-grid-toolkit-client/gridsam/uname
```

A successful run of that script should show something like:

```
$ ./uname_webdav.pl
Submitting the example job of running 'uname -a' 4 times.
Job 1 submitted (urn:gridsam:2381fcc721c582e90121f88d255e0012)
Job 2 submitted (urn:gridsam:2381fcc721c582e90121f88d39060015)
Job 3 submitted (urn:gridsam:2381fcc721c582e90121f88d57d40018)
Job 4 submitted (urn:gridsam:2381fcc721c582e90121f88d6b76001b)
Jobs outstanding: 4. Jobs completed: 0. Jobs failed: 0
Jobs outstanding: 4. Jobs completed: 0. Jobs failed: 0
Jobs outstanding: 4. Jobs completed: 0. Jobs failed: 0
Jobs outstanding: 4. Jobs completed: 0. Jobs failed: 0
Job 1 has completed successfully.
Job 2 has completed successfully.
Jobs outstanding: 2. Jobs completed: 2. Jobs failed: 0
Jobs outstanding: 2. Jobs completed: 2. Jobs failed: 0
Job 4 has completed successfully.
Job 3 has completed successfully.
Jobs outstanding: 0. Jobs completed: 4. Jobs failed: 0
Job 1 result : "Linux gridsam-test.oerc.ox.ac.uk 2.6.18-6-xen-vserver-amd64 #1
SMP Tue May 5 10:39:47 UTC 2009 x86_64 x86_64 x86_64 GNU/Linux"
```

Job 2 result : "Linux gridsam-test.oerc.ox.ac.uk 2.6.18-6-xen-vserver-amd64 #1 SMP Tue May 5 10:39:47 UTC 2009 x86_64 x86_64 x86_64 GNU/Linux"
Job 3 result : "Linux gridsam-test.oerc.ox.ac.uk 2.6.18-6-xen-vserver-amd64 #1 SMP Tue May 5 10:39:47 UTC 2009 x86_64 x86_64 x86_64 GNU/Linux"
Job 4 result : "Linux gridsam-test.oerc.ox.ac.uk 2.6.18-6-xen-vserver-amd64 #1 SMP Tue May 5 10:39:47 UTC 2009 x86_64 x86_64 x86_64 GNU/Linux"

Step 4: Reconfiguring GridSAM server and client to use the UK e-Science certificates

References

http://www.omii.ac.uk/docs/3.4.0/installation_guide/omii_3_installation_and_setup_guide.htm

Section: "Security"->"X.509 Certificates"->"Replacing the Temporary OMII Certificate".

Client

The UK e-Science Certificate is installed using a certificate management tool called `cmtool`, which is installed in the `<OMIICLIENT>/bin` subdirectory during installation of the GridSAM client. `cmtool` logs its output to a file called `CMTtool.log`. Currently, there is no way to specify another log file. The log file is overwritten (rather than being appended) during each run of `cmtool`.

1. Export your previously-obtained personal UK e-Science Certificate into a PKCS12 file. Please find more information about UK e-Science Certificates and NGS at:
<http://www.grid-support.ac.uk/content/view/355/244/>
2. Create a new directory called, for example, `certs` in the `<OMIICLIENT>` directory (for example `~/bin/campus-grid-toolkit-client`)
3. Save the PKCS12 file containing your host certificate in the `<OMIICLIENT>/certs` subdirectory
4. Download the following files, in the PEM format, from the National Grid Service website (<http://www.grid-support.ac.uk/content/view/182/98>):

- a. the UK e-Science CA Certificate (**new-escience-ca.pem**)

```
$ wget http://ca.grid-support.ac.uk/pub/certs/new-escience-ca.pem
```

- b. the UK e-Science CA CRL (**ca-crl.pem**)

```
$ wget http://ca.grid-support.ac.uk/pub/crl/ca-crl.pem
```

- c. the UK e-Science Root Certificate (**new-escience-root.pem**)

```
$ wget http://ca.grid-support.ac.uk/pub/certs/new-escience-root.pem
```

- d. the UK e-Science Root CRL (**root-crl.pem**)

```
$ wget http://ca.grid-support.ac.uk/pub/crl/root-crl.pem
```

5. Save **ca-crl.pem** in the `<OMIICLIENT>` directory
6. Save **new-escience-ca.pem**, **new-escience-root.pem**, **root-crl.pem** in `<OMIICLIENT>/certs`
7. Change directory to the `<OMIICLIENT>` directory
8. Import the PKCS12 file containing your personal UK e-Science certificate, the UK e-Science CA Certificate (**new-escience-ca.pem**) and the UK e-Science CRL (**ca-crl.pem**) into the OMII keystore (**omii.ks**):

```
$ bin/cmtool -S -d . -p <password> certs/<export-filename>.p12 -alias new-escience -ca certs/new-escience-ca.pem -crl ca-crl.pem
```

<password> and <export-filename> refer to the password and filename created when exporting the UK e-Science Certificate.

If successful you will see something like:

```
Importing trusted CA's certificate certs/new-escience-ca.pem into the new keystore ./omii.ks
Done.
```

9. Import the UK e-Science Root Certificate (**escience-root.pem**) into the OMII keystore (**omii.ks**):

```
$ bin/cmtool -I -d . -ca certs/new-escience-root.pem -alias escienceroot
```

10. Manually copy the UK e-Science CA CRL (**ca-crl.pem**) into the <OMII>/CRLs directory

11. Check that the above files have been imported into the keystore (using the password `tmpstore` when requested):

```
$ keytool -v -list -keystore omii.ks | less
```

12. disable the trust of the insecure OMII temporary Certification Authority

```
$ bin/cmtool -D -d . -alias omii_ca
Deleting trusted CA's certificate omii_ca from the keystore omii.ks
Done.
```

Server

1. Export your previously-obtained Host UK e-Science Certificate into a PKCS12 file. Please find more information about UK e-Science Certificates and NGS at:
<http://www.grid-support.ac.uk/content/view/355/244/>

2. Create a new directory called, for example, "certs" in the <OMII> directory (for example in `/home/omii/CGT`).

3. Save the PKCS12 file containing your host certificate in the <OMII>/certs subdirectory.

13. Download the following files, in the PEM format, from the National Grid Service website (<http://www.grid-support.ac.uk/content/view/182/98/>):

- a. the UK e-Science CA Certificate (**new-escience-ca.pem**)

```
$ wget http://ca.grid-support.ac.uk/pub/certs/new-escience-ca.pem
```

- b. the UK e-Science CA CRL (**ca-crl.pem**)

```
$ wget http://ca.grid-support.ac.uk/pub/crl/ca-crl.pem
```

- c. the UK e-Science Root Certificate (**new-escience-root.pem**)

```
$ wget http://ca.grid-support.ac.uk/pub/certs/new-escience-root.pem
```

- d. the UK e-Science Root CRL (**root-crl.pem**)

```
$ wget http://ca.grid-support.ac.uk/pub/crl/root-crl.pem
```

4. Save **ca-crl.pem** in the <OMII> directory

5. Save **new-escience-ca.pem**, **new-escience-root.pem**, **root-crl.pem** and **ca-crl.pem** in `<OMII>/certs`
6. Change directory to the `<OMII>` directory
7. Import the PKCS12 file, the UK e-Science CA Certificate (**new-escience-ca.pem**) and the UK e-Science CRL (**ca-crl.pem**) into the OMII keystore (**omii.ks**):

```
$ bin/cmtool -S -d . -p <password> certs/<export-filename>.p12 -alias
new-escience -ca certs/new-escience-ca.pem -crl ca-crl.pem
```

`<password>` and `<export-filename>` refer to the password and filename created when exporting the UK e-Science Certificate.

If successful you will see something like:

```
Importing trusted CA's certificate certs/new-escience-ca.pem into the new
keystore ./omii.ks
Done.
```

8. Import the UK e-Science Root Certificate (**escience-root.pem**) into the OMII keystore (**omii.ks**):

```
$ bin/cmtool -I -d . -ca certs/new-escience-root.pem -alias escienceroor
```

9. Create the directory `<OMII>/CRLs`
10. Manually copy the UK e-Science CA and root CRLs (**ca-crl.pem**, **root-crl.pem**) into the `<OMII>/CRLs` directory
11. Check that the above files have been imported into the keystore (using the password 'tmpstore' when requested):

```
$ keytool -v -list -keystore omii.ks | less
```

12. The Server must be restarted so that the updated Certificate can take effect
 - a. Stop the Server (if it is running) by entering the following command

```
$ <OMII>/bin/stopomii.sh
Shutting down tomcat
Server running under PID of 2112 detected
Waiting.... HTTP server stopped.
Waiting Process stopped.
```

- b. Start/restart the Server by entering the following command

```
$ <OMII>/bin/startomii.sh
Starting up tomcat
Using JRE_HOME: /usr/java/default
Waiting for container....!... Container started.
```

13. disable the trust of the insecure OMII temporary Certification Authority

```
$ bin/cmtool -D -d . -alias omii_ca
Deleting trusted CA's certificate omii_ca from the keystore omii.ks
Done.
```

Repeat the instructions given in "Step 3: Testing GridSAM server by submitting a job via the client" in order to test the new client/server configuration.

Step 5: Configure the GridSAM server connector to use Globus

References

<http://gridsam.sourceforge.net/2.1.3/gridsam-service/drm.html>

<http://gridsam.sourceforge.net/2.1.3/gridsam-service/gt2.html>

Each GridSAM instance is configured with exactly one type of launching mechanism, here referred to as the DRMConnector implementation. GridSAM is configured with the Fork implementation by default therefore there are no other system dependencies. For more robust and efficient means to distribute the computation to Grid resources, GridSAM layers on systems such as SSH, Condor, Globus, Sun Grid Engine and PBS. The configuration involves modifying the *'jobmanager.xml'* file to import the modules that make up the DRMConnector and modify the contributions these modules require.

1. On your server, change directory to:

```
<OMII>/apache-tomcat-5.5.23/webapps/gridsam/WEB-INF/classes
```

2. Edit `jobmanager.xml` to import the Globus DRM module connector, thus:

```
<?xml version="1.0" encoding="UTF-8"?>
<module id="jobmanager.default" version="1.0.0">
  <sub-module descriptor="jobmanager-globus.xml"/>
</module>
```

3. Edit `jobmanager-globus.xml` in order to connect to your globus resources:

```
<?xml version="1.0" encoding="UTF-8"?>
<module id="jobmanager.globus" version="1.0.0">

  <!-- dependent modules -->
  <sub-module descriptor="org/icenigrd/gridsam/resource/config/common.xml"/>
  <sub-module descriptor="org/icenigrd/gridsam/resource/config/embedded.xml"/>
  <sub-module descriptor="org/icenigrd/gridsam/resource/config/globus.xml"/>
  <sub-module descriptor="database.xml"/>
  <sub-module descriptor="authorisation.xml"/>

  <!-- override the factory defaults here -->
  <contribution configuration-id="hivemind.ApplicationDefaults">
    <!--
      The job manager location
    -->
    <default symbol="globus.JobManager" value="<RESOURCE_DN>/<RESOURCE_RB>"/>
    <!--
      The root filesystem for staging files
    -->
    <default symbol="globus.RootFileSystemURI" value="gsiftp://<RESOURCE_DN>"/>
    <!--
      How often the status of the job is polled
    -->
    <default symbol="globus.MonitoringPeriod" value="20000"/>
  </contribution>
</module>
```

The following values have been used to configure a GridSAM instance to connect to the Oxford NGS cluster Globus resources (job manager and gsiftp):

<RESOURCE_DN> = ngs.oerc.ox.ac.uk

<RESOURCE_RB> = jobmanager-pbs

4. The Server must be restarted so that the updated DRMConnector configuration can take effect:
 - a. Stop the Server (if it is running) by entering the following command

```
$ <OMII>/bin/stopomii.sh
Shutting down tomcat
Server running under PID of 2112 detected
Waiting.... HTTP server stopped.
Waiting Process stopped.
```

- b. Start/restart the Server by entering the following command

```
$ <OMII>/bin/startomii.sh
Starting up tomcat
Using JRE_HOME:          /usr/java/default
Waiting for container....!... Container started.
```

Step 6: Installing grid-proxy-init

References

http://vdt.cs.wisc.edu/releases/1.10.1/installation_pacman.html

<http://www.grid-support.ac.uk/content/view/378/276/>

This step requires a minimal installation of the Globus toolkit. In what follows the Virtual Data Toolkit (VDT) is used:

1. As root create the directory `/usr/local/vdt-1.10.1`. We will use this directory to install the package manager called `pacman` and then a minimal set of the globus toolkit functionalities
2. Change directory into `/usr/local/vdt-1.10.1`
3. Download Pacman:

```
$ wget http://vdt.cs.wisc.edu/software/pacman/3.26/pacman-3.26.tar.gz
```

4. Unpack the tarball:

```
$ tar zxvf pacman-3.26.tar.gz
```

5. Setup your environment to use `pacman`. This example assumes you're using the Bourne shell or Bash. If you're using the C shell you should source `setup.csh` instead.

```
$ cd pacman-3.26
$ source setup.sh
```

6. Change directory into `/usr/local/vdt-1.10.1` and run:

```
$ pacman -v cache -v download -v tar
```

7. Install the VDT Globus-Client package:

```
$ pacman -get http://vdt.cs.wisc.edu/vdt_1101_cache:Globus-Client
Do you want to add [http://vdt.cs.wisc.edu/vdt_1101_cache] to
[trusted.caches]? (y/n/yall):
```

Answer y

Do you agree to the licenses? [y/n]

Answer y

Where would you like to install CA files?

Answer l

Do you want to update the CA certification revocation lists (CRLs) automatically? [y/n]

Answer Y

Would you like to setup daily rotation of VDT log files?

Answer y

Do you want to automatically update your CA Certificates? [y/n]

Answer y

8. Read the file `post-install/README` and follow the instructions to install and enable the automatic update of the certificate authority certificates

- a. Export the following variables and source the configuration files provided by VDT:

```
$ export GLOBUS_LOCATION=/usr/local/vdt-1.10.1/globus
$ export GLOBUS_ERROR_VERBOSE=true
$ . /usr/local/vdt-1.10.1/vdt/etc/vdt-man-setup.sh
$ . /usr/local/vdt-1.10.1/vdt/etc/vdt-local-setup.sh
$ . /usr/local/vdt-1.10.1/perl/perl-setup.sh
```

- b. Edit the value of `cacerts_url` in the configuration file at:

```
$VDT_LOCATION/vdt/etc/vdt-update-certs.conf
```

And uncomment:

```
cacerts_url = http://vdt.cs.wisc.edu/software/certificates/vdt-igtf-
ca-certs-version
```

- c. Change directory into `$VDT_LOCATION/vdt/lib`

- d. Run the following command:

```
$ . $VDT_LOCATION/vdt-questions.sh; $VDT_LOCATION/vdt/sbin/vdt-
setup-ca-certificates
```

- e. Make sure `vdt-update-certs` is enabled, so that future certificate updates are fetched automatically:

```
$ vdt-control --enable vdt-update-certs
$ vdt-control --on vdt-update-certs
```

9. As the `omii` user edit `~/bashrc` and add:

```
export VDT_LOCATION=/usr/local/vdt-1.10.1
export PATH=/usr/local/vdt-1.10.1/vdt/bin:$PATH
```

```

export PATH=/usr/local/vdt-1.10.1/vdt/sbin:$PATH
export MANPATH=$MANPATH:/usr/local/vdt-1.10.1/vdt/man
. /usr/local/vdt-1.10.1/vdt/etc/vdt-man-setup.sh
. /usr/local/vdt-1.10.1/vdt/etc/vdt-local-setup.sh
export GLOBUS_LOCATION=/usr/local/vdt-1.10.1/globus
export GLOBUS_ERROR_VERBOSE=true
. /usr/local/vdt-1.10.1/globus/etc/globus-user-env.sh
. /usr/local/vdt-1.10.1/perl/perl-setup.sh
export MANPATH=$MANPATH:/usr/local/vdt-1.10.1/perl/man
export LD_LIBRARY_PATH=/usr/local/vdt-1.10.1/expat/lib:$LD_LIBRARY_PATH
export MANPATH=$MANPATH:/usr/local/vdt-1.10.1/expat/man
export LD_LIBRARY_PATH=/usr/local/vdt-1.10.1/berkeley-
db/lib:$LD_LIBRARY_PATH
export GPT_LOCATION=/usr/local/vdt-1.10.1/gpt
export PATH=/usr/local/vdt-1.10.1/gpt/sbin:$PATH
export PERL5LIB=/usr/local/vdt-1.10.1/vdt/lib:$PERL5LIB
export PATH=/usr/local/vdt-1.10.1/logrotate/sbin:$PATH
export MANPATH=$MANPATH:/usr/local/vdt-1.10.1/logrotate/man

```

10. Create the directory `~/ .globus`

11. Convert your host certificate in `userhost.pem` and `userkey.pem` (this is necessary in order to use the command `grid-proxy-init`):

```

$ openssl pkcs12 -in <certificate>.p12 -clcerts -nokeys -out
~/ .globus/usercert.pem

```

When asked, enter the password set when exporting the certificate

```

$ openssl pkcs12 -in <certificate>.p12 -nocerts -nodes -out
~/ .globus/userkey.pem

```

When asked, enter the password set when exporting the certificate

```

$ chmod 400 ~/ .globus/userkey.pem

```

12. Create a proxy certificate:

```

$ grid-proxy-init
Your identity: /C=UK/O=eScience/OU=Oxford/L=OeSC/CN=gridsam-
test.oerc.ox.ac.uk/emailAddress=ngs-help@oerc.ox.ac.uk
Creating proxy ..... Done
Your proxy is valid until: Sun Jun 28 06:11:06 2009

```

Note that you are using a host certificate as a user certificate. Unfortunately, `grid-proxy-init` will not support the naming schema `hostcert.pem` and `hostkey.pem` in `~/ .globus`. This is why we call them `usercert.pem` and `userkey.pem` even if strictly speaking, it is not entirely correct.

The proxy certificate has to be updated every 12 hours. It is a good idea to execute a script via `omii crontab` every 10 hours or so in order to update automatically the proxy certificate:

1. Save the following script where you prefer (for example `/home/omii/bin/update_proxy.sh`):

```

#!/bin/sh
. /home/omii/.bashrc
echo <PASSWORD> | /usr/local/vdt-1.10.1/globus/bin/grid-proxy-init
-pwstdin

```

2. Add the following line to your crontab:

```
$ crontab -e
0 0,6,12,18 * * * /home/omii/bin/update_proxy.sh > /home/omii/logs/grid-
proxy-init.log
```

Step 7: Testing the final deployment of GridSAM submitting jobs with and without file staging to the chosen NGS resource

1. Download the NGS GridSAM test suite from <http://>
2. Save `ngs_gridsam_test.tar.bz2` in `<OMIICLIENT>/gridsam`
3. Untar `ngs_gridsam_test.tar.bz2` and change directory into `ngs_tests`
4. Create a proxy certificate:

```
$ myproxy-init -s myproxy.grid-support.ac.uk -l <your_name>
Your identity: /C=UK/O=eScience/OU=...
Creating proxy ..... Done
Proxy Verify OK
Your proxy is valid until: <date>
Enter MyProxy pass phrase: <enter_your_phrase>
Verifying - Enter My Proxy pass phrase: <re-enter_your_phrase>
A proxy valid for 168 hours (7.0 days) for user <your_name> now exists on
myproxy.grid-support.ac.uk
```

5. Run your tests with the following command:

```
$ Gridsam-submit -s https://<your_domain>:18443/gridsam/services/gridsam
-myproxy -myproxyuser <your_name> -myproxyhost myproxy.grid-support.ac.uk
-j <test_name>
```

The output will be:

```
MyProxy password: <enter_your_myproxy_pass_phrase>
Urn:gridsam:<submission_ID> (i.e. 2381fcc72231bab701225577d488000b)
```

6. Check the status of your submission with:

```
$ gridsam-status -s https://<your_domain>:18443/gridsam/services/gridsam
-j Urn:gridsam:<submission_ID>
```

Once your job has been completed, the output will be something like:

```
Job Progress 'Urn:gridsam:<submission_ID>': pending -> staging-in ->
staged-in -> active-queued -> active -> executed -> staging-out -> staged-
out -> done
```

```
--- pending - <date> <time> ---
job is being scheduled
--- staging-in - <date> <time> ---
staging files...
--- staged-in - <date> <time> ---
0 files staged in
--- active-queued - <date> <time> ---
globus job is pending
--- active - <date> <time> ---
```

```

globus job is active
--- executed - <date> <time> ---
globus job completed
--- staging-out - <date> <time> ---
staging files out...
--- staged-out - <date> <time> ---
2 files staged out
--- done - <date> <time> ---
Job completed

-----
Job Properties
-----
urn:gridsam:globus:id=https://<NGS_resource>:<number>/<number>/<number>/
urn:gridsam:principal=CN=...
urn:gridsam:globus:rsl=&
(executable=<location_name>)
  (arguments=-a )
  (stdout=<location_name>)

  (stderr=<location_name>)

(jobType=<type>)
-----

```

Oxford Use Case

Oxford GridSAM instance:

- GridSAM Version: 2.1.4
- Platform: SL4.7 Xen Virtual machine with 1GB ram
- Java version: Java(TM) SE Runtime Environment (build 1.6.0_13-b03)
- Activity: 1 Apr 2009 - 1 Jul 2009, 663 jobs submitted

After an initial period of debugging that has led to the release of GridSAM 2.1.4, the GridSAM instance has never crashed during an uptime of more than 2 months. Its footprint is stable at around 300MB (VmSize).