How do I use the OSG Grid

Overview

This example shows how to run jobs and especially SLIC, the Simulator for the Linear Collider, on the **FermiGrid** which is part of the **Open Science Grid**. **SLIC** is a Geant4-based simulations package that uses an XML geometry input format called LCDD to describe geometry, sensitive detectors and readout geometry.

> The example scripts should be pasted directly into the terminal on detsim. Do *not* use an editor, as the escape characters will not be interpreted correctly.

Prerequisites

1. Obtain a DOE grid certificate from http://security.fnal.gov/pki/Get-Personal-DOEGrids-Cert.html, which also explains how to export the certificate from the browser, dealing with directory permissions, etc.
2. Register with the ILC VO (Virtual organization) at http://cd-amr.fnal.gov/ilc/ilcsim/ilcvo-registration.shtml, which will guide you to: https://voms.fnal.gov:8443/vomrs/ilc/vomrs
3. Get an account on **detsim**, using the following form http://cd-amr.fnal.gov/ilc/ilcsim/ilcsim.shtml. This machines serve as a portal to the grid.

> Setting up your own gateway to the grid is beyond the scope of this write-up. It involves installing and configuring the Virtual Data Toolkit (VDT) , installing a **host certificate** for the gateway machine, etc. For an administrative guide see the Fermi Grid web page.

Setup and Configuration

Kerberos

Fermilab uses Kerberos for external authentication. *This section assumes that you have a Fermilab Kerberos principal.* Follow these instructions if you need an account at Fermilab and are authorized to obtain one.

Assuming that your machine has recent versions of SSH and Kerberos and you will *not* be using a Cryptocard, download Fermilab's official Kerberos configuration file.

Download the file.

```
wget http://security.fnal.gov/krb5.conf
```

Set the environment variable KRB5_CONFIG to point to the Fermilab configuration file.

```
export KRB5_CONFIG=`pwd`/krb5.conf
```

This variable can be added to your shell profile or setup in a script and the configuration file it points to will override the one in */etc*.

Connecting to detsim

Initialize the Kerberos session.

```
kinit -f USERNAME@FNAL.GOV
```

Connect to detsim using *ssh*

.

```
ssh USERNAME@detsim.fnal.gov
```

You may need to use ssh_config for the SSH configuration file.

```
ssh -F ssh_config USERNAME@detsim.fnal.gov
```

Session Certificate and quotas

Finally, obtain a session certificate .

```
voms-proxy-init -voms ilc:/ilc/sid
```

By default the proxy is valid for 12 hours, which is probably too short for your job. To obtain a proxy that is valid for 72 hours, issue the command:

```
voms-proxy-init -valid 72:00  -voms ilc:/ilc/sid
```

The following groups should have a higher limit on the number of current jobs than the generic */ilc/detector* so feel free to substitute them in the *voms-proxy-init* command. The following webpage list the quotas on the general FERMIGRID: http://fermigrid.fnal.gov/grid_users/fermigrid_gpgrid.pdf

- **/ilc/sid** - SiD
- **/ilc/ilddet** - ILC Large Detector

To check the status of the proxy:

```
voms-proxy-info -all
```

To check quotas and to check how many slots are already taken:

```
condor_config_val GROUP_QUOTA_group_siddet -name fnpc5x1.fnal.gov -pool fnpccm1.fnal.gov
condor_userprio -all -pool fnpccm1.fnal.gov
```

Running from an External Site

If you want to submit jobs from a node other than detsim, the ilc VOMS server information needs to be explicitly provided.

The following should be put into a file, *ilc-fermilab-voms*.

```
"ilc" "voms.fnal.gov" "15023" "/DC=org/DC=doegrids/OU=Services/CN=http/voms.fnal.gov" "ilc"
```

Now a valid grid certificate can be obtained by referencing this configuration file.

```
voms-proxy-init ilc:/ilc -userconf ./ilc-fermilab-voms
```

The above command will fail if *ilc-fermilab-voms* is not owned by your account.

Provided that your local node is configured correctly, this should allow you to start a grid session on an external node outside Fermilab.

Simple commands such as *globus-job-run* should work "out of the box" from an external site. In order to actually submit jobs to the Fermilab batch system, you will need to have a Condor job scheduler running. Talk to your site administrator about setting up this software, which can be configured as part of the VDT.

Example Grid Jobs

Submitting the First Example Jobs

Now you should be all setup to submit a test job to make sure that everything is working. Cut and paste the following lines into your terminal window. This will submit a grid job which starts 5 separate processes. The processes will just execute sleep for 10 seconds before terminating. Since no output is created the sleep_grid.out.$(Cluster).$(Process) and sleep_grid.err.$(Cluster).$(Process) files should be empty.

(**Note!**: $(Cluster) represents the job number and $(Process) represents the (5) process numbers)
The condor log files are: sleep_grid.log.\$(Cluster).\$(Process)

```
cat > sleep_grid << +EOF
universe = grid
GridResource = gt2 fnpcosg1.fnal.gov/jobmanager-condor
executable = /bin/sleep
transfer_output = true
transfer_error = true
transfer_executable = true
log = sleep_grid.log.\$(Cluster).\$(Process)
notification = NEVER
output = sleep_grid.out.\$(Cluster).\$(Process)
error = sleep_grid.err.\$(Cluster).\$(Process)
stream_output = false
stream_error = false
ShouldTransferFiles = YES
WhenToTransferOutput = ON_EXIT
globusrsl = (jobtype=single)(maxwalltime=999)
Arguments = 10
queue 5
+EOF


condor_submit sleep_grid
```

The second example is an exploration job where the job reports the run time environment it encounters and the file systems that are mounted. This is very often useful to find out what is available on the worker nodes 🙂. So have a look at env_grid.out.$(Cluster).$(Process).

Note!: The grid job doesn't inherit the run time environment from your interactive session!

```
rm -f env_grid.sh
cat > env_grid.sh << +EOF
#!/bin/sh -f
printenv
pwd
cd \${_CONDOR_SCRATCH_DIR}
pwd
#
# This sets up the environment for osg in case we want to
# use grid services like srmcp
#
. $OSG_GRID/setup.sh
source \${VDT_LOCATION}/setup.sh
printenv
/bin/df
+EOF
chmod +x env_grid.sh

rm -f env_grid.run
cat > env_grid.run << +EOF
universe = grid
GridResource = gt2 fnpcosg1.fnal.gov/jobmanager-condor
executable = ./env_grid.sh
transfer_output = true
transfer_error = true
transfer_executable = true
log = env_grid.log.\$(Cluster).\$(Process)
notification = NEVER
output = env_grid.out.\$(Cluster).\$(Process)
error = env_grid.err.\$(Cluster).\$(Process)
stream_output = false
stream_error = false
ShouldTransferFiles = YES
WhenToTransferOutput = ON_EXIT
globusrsl = (jobtype=single)(maxwalltime=999)
queue
+EOF

condor_submit env_grid.run
```

Submitting a Job running SLIC

Now finally let's run SLIC 🙂 . We will use the SLIC installation and a data set that are available on the GRID worker nodes. As in the previous examples cut and paste the contends below:

```
rm -f slic_grid.csh
cat > slic_grid.csh << +EOF
#!/bin/csh
echo start
/bin/date
cd \${_CONDOR_SCRATCH_DIR}
setenv LABELRUN slic_grid-\${ClusterProcess}
setenv TARFILE \${LABELRUN}-results.tar
echo \${TARFILE}
echo start
/bin/date
mkdir results
/grid/app/ilc/sid/SimDist/v2r4p2/SimDist/scripts/slic.sh -r 5   \
-g /grid/app/ilc/detector/SimDist/detectors/sid01/sid01.lcdd            \
-i /grid/data/ilc/detector/LDC/stdhep/ZZ_run10.stdhep -o ./results/ZZ_run10\${LABELRUN} >& \
./results/ZZ_run10\${LABELRUN}.lis
ls -lh results
/bin/date
echo "build output tarball: " \${TARFILE}
tar -cf \${TARFILE} results
echo done
+EOF
chmod +x slic_grid.csh

rm -f slic_grid.run
cat > slic_grid.run << +EOF
universe = grid
GridResource = gt2 fnpcosg1.fnal.gov/jobmanager-condor
executable = ./slic_grid.csh
transfer_output = true
transfer_error = true
transfer_executable = true
environment = "ClusterProcess=\$(Cluster)-\$(Process)"
transfer_output_files = slic_grid-\$(Cluster)-\$(Process)-results.tar
log = slic_grid.log.\$(Cluster).\$(Process)
notification = NEVER
output = slic_grid.out.\$(Cluster).\$(Process)
error = slic_grid.err.\$(Cluster).\$(Process)
stream_output = false
stream_error = false
ShouldTransferFiles = YES
WhenToTransferOutput = ON_EXIT
globusrsl = (jobtype=single)(maxwalltime=999)
queue
+EOF

condor_submit slic_grid.run
```

Running Commands directly on the Head Node

To run some commands directly on the grid head nodes use a syntax like this:

```
globus-job-run fngp-osg.fnal.gov/jobmanager-condor /bin/ls /grid/app
globus-job-run fngp-osg.fnal.gov/jobmanager-condor /usr/bin/printenv
globus-job-run fngp-osg.fnal.gov/jobmanager-condor /bin/df
```

The examples above show how to check what grid applications are installed, the runtime environment of a job and what file systems are mounted. To check for available SLIC/SimDist distributions type:

```
globus-job-run fngp-osg.fnal.gov/jobmanager-condor /bin/ls /grid/app/ilc/detector/SimDist/
```

Checking and Killing your Jobs, releasing held jobs

You can see the status of all jobs using the following command:

```
condor_q
```

or

```
condor_q -globus
```

Or to check the jobs submitted by user <username>:

```
condor_q  -submitter <username>
```

You can view information about all requests with the following command:

```
condor_status  -submitters
```

To cancel a job type condor_rm followed by the job number:

```
condor_rm <job number>
```

Condor can put a job into held state when e.g. the proxy expires while the job is running. In that case the job still might be running fine on the worker node but even after successsful completion there will not be any log files etc. copied back. To remedy that situation renew the proxy and then release the jobs.

```
voms-proxy-init -valid 72:00  -voms ilc:/ilc/sid
condor_release -all
```