Documentation for Accessing MCD R&D Computing Resources.

Obtaining OSG Credentials for the MCDRD VO

The first step is joining the Muon Collider Detector Research and Development Virtual Organization. This will give you access to MCD grid computing resources. This section tells you how.

Note: As of March 23, 2013, DOEGrids CA will stop issuing new certificates as the transition to OSG completes. For more information see the Fermilab PKI Transition page.

Follow the instructions on the Fermilab documentation for obtaining a personal OSG certificate as a Fermilab staff or user. You will be joining the MCDRD VO, but when you fill out the OIM User Certificate Request Form choose Fermilab as the VO.

Upon submitting the OIM User Certificate Request form you should receive mail confirming the opening of your request ticket. Shortly thereafter it should be confirmed, allowing you to download the ticket from the OIM ticket page.

Import the ticket into your browser (In Chrome, go to Settings -> Advanced Settings... -> HTTP/SSL -> Manage Certificates... -> Import...). Now you should be able to access the MCDRD VO registration page. Fill in the form on this page and submit. The request will be submitted to the VO's administrator and you should get a confirmation email when the submission is made and when the request is approved.

Using the OSG

Setup:

(Subsections are modified from this page unless otherwise noted.)

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
```

Using Globus tools for submitting grid jobs from Linux/UNIX

This subsection modified from https://fermi.service-now.com/kb_view.do?sysparm_article=KB0010815

If you will be using Globus tools to run grid jobs from a Linux or other UNIX machines, you need to get a proxy certificate. To do so, your certificate and user key need to be in PEM format. To convert them from their original PKCS#12 format to PEM:

1. Export your Open Science Grid certificate from your browser.
2. Use the scp utility to copy the certificate to your detsim account, then ssh into detsim to perform the rest of these steps.

   ```
   $ scp /path/to/<YourCert>.p12 USERNAME@detsim.fnal.gov:~/
   $ ssh USERNAME@detsim.fnal.gov
   ```

3. Convert the certificate using the openssl command as shown (use your actual .p12 certificate filename with no angle brackets; use the output name usercert.pem as shown). (You may have to create the $HOME/.globus directory)

   ```
   $ openssl pkcs12 -in <YourCert>.p12 -clcerts -nokeys -out $HOME/.globus/usercert.pem
   ```

4. To get the encrypted private key (again use your actual .pl2 certificate filename; use the output name userkey.pem as shown):

   ```
   $ openssl pkcs12 -in <YourCert>.p12 -nocerts -out $HOME/.globus/userkey.pem
   ```

5. You must set the mode on your userkey.pem file to read/write only by the owner, otherwise grid-proxy-init will not use it

   ```
   $ chmod go-rw $HOME/.globus/userkey.pem
   ```

Session Certificate and quotas

Finally, obtain a session certificate .

```
voms-proxy-init -voms mcdrd:/mcdrd
```

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 mcdrd:/mcdrd
```

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
```

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
```