Grid Setup Guide

Niamh Fearon

October 22, 2019

Contents

1 Getting a Grid Certificate 1

2 Joining a Virtual Organisation 1

3 Preparing your Grid Certificate 2

4 DIRAC Client Installation 2

5 First Grid Jobs 2

5.1 Simple Job 2

5.2 Hello World Job 3

1 Getting a Grid Certificate

The first step needed to access the northgrid distributed computing systemis to apply for a grid certificate. To do this go to the website www.hep.manchester.ac.uk/ local and when prompted for a username and password enter:

- Username: hep
- Password: schuster

This will display the Manchester HEP Internal pages with various links to HEP Group information. To apply for a grid certificate click the link called 'Apply for an e-Science Certificate'. From here select the 'Request New User Certificate' option. You will need to fill out the form in this area, making note of the password you choose to use as you will need it later. The PIN we will use is 12345678910. After submitting the form a key and a Certificate Signing Request (CSR) a generated in the text box below the form. Copy and paste these keys to a blank text document in your local area. The bash command:

```
$ gedit GridCertKey.txt
```

creates and opens a blank text file called GridCertKey where your keys can be copied to and saved.
Once the request has sent through you will receive an email acknowledging the request. You will need to bring a scanned copy of photo ID (student card will do) and the PIN used for the online form to Sabah Salih's office on the 6th Floor. A confirmation email will be sent to your university email containing instructions that will allow you to download the grid certificate.
The grid certificate can be downloaded using the CAPortal link in the email (Option 1). You will need to enter the serial number in your email and the email address you used for the sign up. When asked for the private keys generated in the initial request click browse and find the GridCertKey.txt file that you created earlier. You will also need your password. Click save certificate and the *.p12* file will be downloaded in your local area.
Go into your browser and find the advanced preferences settings where the certificates are kept. Find the 'Your certificates' tab and import the *.p12* file. This will now allow you access to joining the local Virtual Organisation (VO).

2 Joining a Virtual Organisation

Once the *.p12* certificate is saved in your browser you can now request to become a member of the NorthGrid. Go to the website vo.northgrid.ac.uk and fill out the form to become a member and have access to the grid. An email will be sent which requires you to click on a link to confirm the membership request and another email is sent when the membership request is approved.

3 Preparing your Grid Certificate

The first step is to move the Grid certificate to your local directory. In your home HEP area create a directory to copy the certificate to.

```
$ cd ~
$ pwd
</afs/hep.man.ac.uk/u/username>
$ mkdir .globus
$ scp <path>/<grid cert name>.p12 ./.globus/
```

Now the grid certificate needs to be converted into separate certificate and key files. To do this:

```
$ cd ~/.globus
$ openssl pkcs12 -in <grid certificate name>.p12
-clcerts -nokeys -out usercert.pem
$ openssl pkcs12 -in <grid certificate name>.p12
-nocerts -out userkey.pem
```

When prompted for an import password use the password used for the initial registration. Use the same password for the PEM pass phrase. The permissions also need to be changed for the two newly made files.

```
$ chmod 400 userkey.pem
$ chmod 600 usercert.pem
```

Now you can use GridPP DIRAC.

4 DIRAC Client Installation

Create a directory within one of the PC server disks e.g. pc20xx-datax/ to install the UI, as the space in your local home area is not sufficient. The sequence of bash commands to install the UI are as follows:

```
$ mkdir dirac_ui
$ cd dirac_ui
$ wget -np -O dirac-install https://raw.githubusercontent.com
/DIRACGrid/DIRAC/integration/Core/scripts/dirac-install.py
$ chmod u+x dirac-install
$ ./dirac-install -r v6r20p5 -i 27 -g v14r1
$ . bashrc
$ dirac-proxy-init -x # (needs user cert password)
$ dirac-configure -F -S GridPP -C
dips://dirac01.grid.hep.ph.ic.ac.uk:9135/Configuration/Server -I
$ dirac-proxy-init -g vo.northgrid.ac.uk_user -M
```

If an error message prints at any point in this install then the most likely problem is a delay with registering with the VO and the DIRAC server picking this up. If after 24 hours the message persists then contact lcg-site-admin@imperial.ac.uk and they will have a look.

5 First Grid Jobs

5.1 Simple Job

In the *dirac_ui* directory create a simple *.jdl* file to submit your first DIRAC job.

```
[
JobName = "Simple_Job;
Executable = "/bin/ls";
Arguments = "-ltr";
StdOutput = "StdOut";
StdError = "StdErr";
OutputSandbox = {"StdOut","StdErr"};
]
```

Submit the job using the bash commands:

```
$ dirac-wms-job-submit -f logfile Simple.jdl
JobID=xxxxxx (useful to note down)
```

To check the job status:

```
$ dirac-wms-job-status -f logfile
JobID=xxxxxx Status=Waiting; MinorStatus=Pilot Agent Submission;
Site=ANY
```

Eventually the job will be finished.

```
$ dirac-wms-job-status -f logfile
JobID=xxxxxx Status=Done; MinorStatus=Execution Complete;
Site=LCG.Glasgow.uk; (where the job ran)
```

Once the job is done retrieve the output.

```
$ dirac-wms-job-get-output -f logfile
```

This creates a directory in the *dirac_ui* directory called the numbers of the JobID. Inside the job directory is the output of the job. In this case the output is called *StdOut* and contains the permissions, date and time of the job.

5.2 Hello World Job

The next step is running a Hello World job using the DIRAC API. The DIRAC API is encapsulated in Python classes designed to be used easily for job management. Using the API classes it is easy to write small scripts to run jobs, check job status and retrieve outputs.
Use the API to create a Hello World job, called *hello_world.py*.

```
# start DIRAC setup
from DIRAC.Core.Base import Script
Script.parseCommandLine( ignoreErrors = True )
# need these lines in every .py script

from DIRAC.Interfaces.API.Job import Job
from DIRAC.Interfaces.API.Dirac import Dirac

dirac = Dirac()
j = Job()

j.setCPUTime(500)
j.setExecutable('/bin/echo hello')
j.setExecutable('/bin/hostname')
j.setExecutable('/bin/echo hello again')
j.setName('API')

jobID = dirac.submit(j)
print 'Submission Result: ',jobID
```

In this example the job has 3 different steps, 2 echoes and one hostname. To submit the job:

```
$ python hello_world.py
```

This will output information to terminal, including the JobID. You can use the JobID to monitor the job status using a script, *job_monitoring.py*.

```
from DIRAC.Core.Base import Script
Script.parseCommandLine( ignoreErrors = True )

from DIRAC.Interfaces.API.Dirac import Dirac
from DIRAC.Interfaces.API.Job import Job
import sys
dirac = Dirac()
jobid = sys.argv[1]
print dirac.status(jobid)
```

To run this:

```
$ python job_monitoring.py <JobID>
```

Once the job is complete you can retrieve the outputs using another script *retrieve_hello_world.py*.

```
from DIRAC.Core.Base import Script
Script.parseCommandLine( ignoreErrors = True )

from DIRAC.Interfaces.API.Dirac import Dirac
from DIRAC.Interfaces.API.Job import Job
```

```
import sys
dirac = Dirac()
jobid = sys.argv[1]
print dirac.getOutputSandbox(jobid)
```

Execute the script by:

```
$ python retrieve_hello_world.py <JobID>
```

This will create a directory titled the JobID number and store the output files inside here. The outputs will included the 2 hello world statements and also print the server used to run the job.