Title

Objective

This tutorial details the procedure to launch a MATLAB script as an OAR job on IGRIDA.

Limitations

Beware of the number of licences for MATLAB as each OAR job will consume a token. Prefer compiling your script as in the following tutorial.

Logging In

Start by connecting the IGRIDA frontend using SSH:

```
mymachine%ssh mylogin@igrida-oar-frontend
```

Welcome to the IGRIDA frontend! It is prohibited to start processes at this level, only interactive or batch OAR jobs are allowed.

Preparing the Scripts

As a good practice, it is recommended that you create a log directory, located in `/temp_dd/igrida-fs-1/mylogin`, where the run will take place. In particular, your run outputs should never be written to your home directory, which would necessarily drive to an NFS crash at some time. For instance, type:

```
mkdir -p /temp_dd/igrida-fs1/mylogin/log/
```

Then, create an OAR launch script in your home directory `/udd/mylogin/matlab-igrida-demo/`. In this script, you start by loading the Matlab module. You can also specify the number of cores and the wall time. Do not forget to redirect the output and the error of the OAR job. The launch script ends by calling your Matlab m-file.

oar-launch-script.sh

```
#!/bin/bash
source /etc/profile.d/modules.sh
module load matlab

#OAR -l core=10,walltime=6:00:00
#OAR -O /temp_dd/igrida-fs1/mylogin/log/job.%jobid%.output
```

```
#OAR -E /temp_dd/igrida-fs1/mylogin/log/job.%jobid%.error
echo "My job was ran on these nodes:"
cat $OAR_NODEFILE

#Setup MCR cache directory locally
export MCR_CACHE_ROOT=/tmp/mcr_cache_${USER}_OAR_JOBID_${OAR_JOBID}
mkdir -p $MCR_CACHE_ROOT

#cd to your execution directory first
cd /udd/mylogin/matlab-igrida
matlab -r matlab-demo

#Remove temporary MCR cache directory
/bin/rm -rf $MCR_CACHE_ROOT
```

In your Matlab file `matlab-demo.m`, all the output must be redirected to the `/temp_dd/igrida-fs1/mylogin/` filesystem. Here is a simple example:

```
a=1;
b=a+1;
matFilename = sprintf('/temp_dd/igrida-fs1/mylogin/results.mat');
save(matFilename,a,b);
```

Prepare you OAR launch script to be executed:

```
igrida-oar-frontend%chmod +x oar-launch-script.sh
```

Launching the Job

You can now submit eh job using the command `oarsub` as in the following:

```
igrida-oar-frontend%oarsub -S  oar-launch-script.sh
[ADMISSION RULE] Modify resource description with type constraints
[ADMISSION RULE] Job walltime greater than 4 hours, adding property
duration_weight > 2 (large job).
Generate a job key...
OAR_JOB_ID=666666
```

Step back and watch your job being executed.

Verifying the Output

If you need to follow the execution of your job, you can check the status of all your jobs:

```
igrida-oar-frontend%oarstat -f | grep mylogin
```

and verify the output or the errors generated using a specific job ID:

```
igrida-oar-frontend%more /temp_dd/igrida-fs1/mylogin/log/job.jobID.output
igrida-oar-frontend%more /temp_dd/igrida-fs1/mylogin/log/job.jobID.error
```

From:
http://wiki.lahoud.fr/ - **wikiroute**

Permanent link:
http://wiki.lahoud.fr/doku.php?id=matlab_script_on_igrida&rev=1374312773

Last update: **2014/01/11 05:25**