THE UNIVERSITY OF ALABAMA IN HUNTSVILLE
Department of Electrical and Computer Engineering

CPE 412/512
Fall Semester 2015

Using the Batch Queuing System on the dmc.asc.edu system using *run_script* utility

Note using the Batch Queuing system is the only way to run jobs that execute for more than 10 minutes. Interactive jobs that are run from the command line that exceed this limit will be killed by the operating system (and will not get exclusive access to the machine while they are running!).

Step 1: Change to your appropriate working directory

Step 2: Create the source file and compile for mpi in the usual manner.

ex. mpic++ test.cpp -o test

Step 3: Using your favorite text editor (vi, nano, emacs, pico, etc.) create a bash script file that contains the module load command for mpi and also the mpirun command(s) with the appropriate parameters. For example if you want to have a two process MPI run for a file with an executable name *"test"* you may create the file *"r2.scr"* which contains the following lines of text.

```
#!/bin/bash
module load openmpi
mpirun -np 2 test
```

Step 4: Give your script file execute privileges by typing

```
chmod 744 r2.scr
```

Step 5: Then type the command run_script r2.scr. This is an interactive script and the following dialog information will appear.

```
Queue                  CPU    Mem # CPUs
-------------- ---------- ------ ------
small-serial     40:00:00    4gb      1
medium-serial    90:00:00   16gb      1
large-serial    240:00:00  120gb      1
small-parallel   48:00:00    8gb    2-8
medium-parallel 100:00:00   32gb   2-16
med-parallel    100:00:00   32gb   2-16
large-parallel  240:00:00  120gb   2-64
highly-parallel 240:00:00  256gb 64-128
class             2:00:00   64gb   1-64
daytime           4:00:00   16gb    1-4
express          01:00:00  500mb      1

Enter Queue Name (default <cr>: small-serial) class

Enter number of cpus (default <cr>: 1 ) 2

Enter Time Limit (default <cr>: 4:00:00 HH:MM:SS)

Enter memory limit (default <cr>: 2gb )

The job name will be r2scrSCRIPT (can be changed in .asc_queue file)

Should this job run on uv, or dmc (default: any)
dmc
============================================================
=====         Summary of your script     job           =====
============================================================
  The script file is:  r2.scr
  The time limit is  4:00:00  HH:MM:SS.
  The memory limit is:  2gb
  The job will start running after:  201309171133.23
  Look for:  r2scrSCRIPT  in queue:  class

Job number 982355.mds1.asc.edu
```

enter class to select "class" queue

enter 2 to reserve two processing cores

accept defaults press <enter>

accept defaults press <enter>

enter dmc

Batch Queue Job ID

Enter **class** for *Queue Name*, **2** (in this case) for number of cpus (must be equal to or less than mpirun -np x value entered in script file), and **dmc** for the machine it is to run on. Press the enter key to accept all other defaults for the other prompts. All screen output should go to the file r2scrSCRIPT.o982355 (i.e. the <filename>SCRIPT.o<job number> file in general).

You can view your status using the ***qstat*** command. For example type

qstat

to view the status of your job(s). If you are user *uahcls42* then the following output may appear.

```
                                                                  Req'd    Req'd       Elap
Job ID                  Username    Queue    Jobname          SessID  NDS   TSK   Memory   Time    S   Time
----------------------- ----------- -------- ---------------- ------ ----- ------ ------ --------- - ---------
982355.mds1.asc.edu     uahcls42    class    r2scrSCRIPT         --      1      2    2gb  04:00:00 Q       --
```

Unlike jobs run without this queue management in place, jobs run using the batch queue will be the only ones executed on the portion of the system you have reserved while your program is running. **Note that using this Batch Queue is the only way to run jobs that execute for more than 10 minutes. Jobs that are run from the command line that exceed this limit will be killed by the operating system (and will not get exclusive access to the machine while they are running!)**

You can also remove a job from the queue before it completes execution by using the ***qdel*** command with your job number as the command line argument. For example if you realize you made a mistake in your code and do not want job number 982355 run after it has been submitted to the queue, simply type

qdel 982355

to delete it.