Running Abaqus on the pangea cluster at Marquette University

pangea login

Log onto pangea in Linux via

```
ssh -Y -l yourUserNameGoesHere pangea.marqnet.mu.edu
```

Abaqus CAE setup

add ($$ means command line, don’t type the $$’s) the following:

```
$$ module load abaqus
```

to your .bashrc (which is located in ~/.bashrc , if it doesn’t exists you can create it, just remember to start the file name with a period “.”), then type

```
$$ source .bashrc
```

Once that is added to the .bashrc file

You can open abaqus cae using the command

```
$$ abaqus_cae
```

Running Abaqus

to run Abaqus on pangea, use the following commands

The abaqus executable is in, if you ever need to track it down

```
/cm/shared/apps/abaqus/DassaultSystemes/SIMULIA/Commands/abaqus
```

If Abaqus is setup correctly

```
$$ which abaqus
```

Should give you this path.

To run CAE use

```
$$ abaqus_cae
```

To run a background job called myInput.inp

```
$$ abaqus job=myInput
```

This will create a .que file that submits the job. It will also create a .qlog file so you can monitor your job. Since .qlog does not give a lot of info, you can also track the .dat and .sta file which are located in /cm/shared/scratch/*yourUserName*_myInput_0_exec/ (where *yourUserName* is your specific user name)

I like to use tail -f /cm/shared/scratch/*yourUserName*_myInput_0_exec/myInput.dat to track the .dat file

When the run in complete the file will be moved back to your working directory. If you kill the job, or it hits the time limit, it will not be moved back to your directory and you should delete the temporary directory.

To run a background job called myInput.in on 12 processors

```
$$ abaqus job=myInput cpus=12
```

To run a background job called myInput.in on 36 processors

```
$$ abaqus job=myInput cpus=36
```

To run a background job called myInput.in on 12 processors with a timelimit of 300 minutes

```
$$ abaqus job=myInput cpus=12 -queue queabaqus:wall300
```

A good rule-of-thumb (heuristic) is that your wall time should never exceed 24 hours, if it does you should write a restart file, take a look at the results, and restart it for another 24 hours

You can also change the partition (i.e., what type of nodes are used and what type of limits are set on the job), the available partitions are

- moorecores (newer nodes 6-9, 2.7GHz, 36 cores per node, Limited to 48 hours)
- roycores (older nodes 1-5, 3.0 GHz, 24 cores per node, Limited to 48 hours)
- debug (all nodes, Limited to 120 minutes, Default time allocated is 30 minutes)
- defq (all nodes, Limited to 24 hours)
- longq (don't use this unless its absolutely necessary, it does not have a time limit, it is rare that you should be running a job for more that 24 hours without restarts files, and without checking on it, it's a good neighbor policy not to use this partition without checking with other folks using pangea and see if you job may be conflicting with theirs over its long run time)

for example

```
$$ abaqus job=myInput cpus=12 -queue quemoorecores:wall300
```

would run on nodes 6-9.

To run a background job called myInput.in with a user defined material called myUmat.f

```
$$ abaqus job=myInput user=myUmat.f
```

Its good to run user materials using double precision so the following is recommended when using user defined inputs

```
$$ abaqus job=myInput user=myUmat.f double
```

When running on partial nodes (especially, nodes 6-9 that have many cores) it is good to use

```
$$ abaqus job=s4b cpus=40 -queue quemoorecores:wall100 -mp_host_split 2
```

Right now (12/19/19) there are some issues with using a full node plus 1 core on a second node. For now, you should use multiple cores on additional nodes and the -mp_host_split 2 command

Other parameters that can be set through the command line are available in
/cm/shared/apps/abaqus/common/SLURMQueue.env

To run a python file called post_tensor.py use the following command

```
$$ Abaqus python post_tensor.py
```

If you have a python file that plots the results in MatPlotLib called plotResults.py, that can be run using regular python

```
$$ python plotResults.py
```

Other pangea is for is at
https://marq.sharepoint.com/teams/pangea/Wiki/Home.aspx