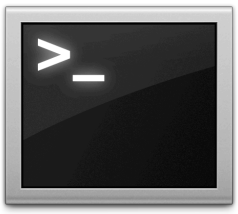
Running Jobs, Submission Scripts, Modules

Matt Gitzendanner: magitz@ufl.edu

5/28/13

UF Research Computing



How do I get my jobs started?

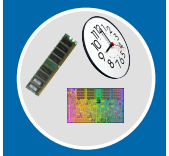
Cluster basics

User interaction

Galaxy

Login node (Head node)

Scheduler



Tell the scheduler what you want to do

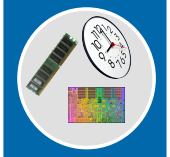
Compute resources

Your job runs on the cluster

Scheduling a job

- Need to tell scheduler what you want to do
  - **How many CPUs** you want and how you want them grouped
  - **How much RAM** your job will use
  - **How long** your job will run
  - The commands that will be run

Scheduler



Tell the scheduler what you want to do

UF Research Computing

- Ordinary Shell Script

```
#!/bin/bash

date
module load test_app
test_app –i file.txt
```

Read the manual for your application

Commands typed on the command line can be put in a script.

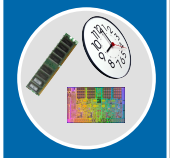
UF Research Computing

- Submission Script

```
#!/bin/bash
#
#PBS -N My_Job_Name
#PBS -M Joe_Shmoe@ufl.edu
#PBS -m abe
#PBS -o My_Job_Name.log
#PBS -j oe
#PBS -l nodes=1:ppn=1
#PBS -l walltime=00:05:00
#PBS –l pmem=900mb

cd $PBS_O_WORKDIR
date
module load test_app
test_app –i file.txt
```

Scheduler



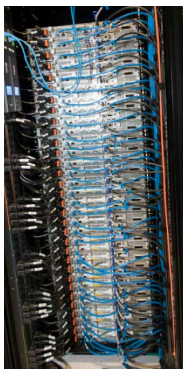
Tell the scheduler what you want to do

Nodes and processors

```
#PBS –l nodes=1:ppn=4
#PBS –l nodes=2:ppn=8
```

Heterogeneous cluster

- There is a wide mix of nodes on the cluster
  - From 4 cores per node
  - To many with 12-16 cores
  - *HiPerGator*-16K cores, **64** cores/ node
    - *Starting in June*
- The more ppn you ask for, the smaller the pool of nodes that can service your job



RAM

```
#PBS –l pmem=900mb
```

- Lots to consider, but do your best at estimating RAM needed for job
- Over about 3GB of RAM, "costs" toward CPU allocation

Wasted RAM leads to idle CPUs and low job throughput

Processor equivalents

- Accounts for large RAM requests
- Average ~3GB RAM/core

1 core, 10GB RAM: ~3 PEs
1 core, 60GB RAM: ~18 PEs

- Non-investors limit: 8 PEs
- Investor limits are based on PEs

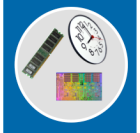
Walltime

```
#PBS –l walltime=00:50:00
```

- Fairly straight forward
- As with all resource requests, accuracy helps ensure ***your*** jobs and all other jobs will run sooner

	Maximum	Short	Long
Investor	31 days	<12 hrs	7 days
Other	7 days	<12 hrs	3 days

Scheduler



Tell the scheduler what you want to do

UF Research Computing

- Job Management
  - `qsub <file_name>`: job submission
  - `qstat –u <user>`: check queue status
  - `qdel <JOB_ID>`: job deletion
  - `checkjob –v <job number>` (shows PE value)

Lots of jobs

- You can script your job submission, ***BUT***:
  - How long will each job run?
    - Many short (<20 minutes) jobs are inefficient
      - Scheduling overhead

Pipettes only $1.99 each! *

* Plus $1.50 shipping per order

Would you order one at a time
or
place one order for 100?

Lots of jobs

```
#!/bin/bash
#PBS -l nodes=1:ppn=1
#PBS -l walltime=00:05:00
#PBS -l pmem=900mb

cd $PBS_O_WORKDIR
date
module load test_app
test_app –i file1.txt
test_app –i file2.txt
test_app –i file3.txt
…
test_app –i fileN.txt
```

You can very easily run multiple tasks in a single job script

Lots of jobs

- If you do submit lots of individual jobs:

  There is probably a better way…ask for help

  - **2,000-3,000** jobs maximum at a time
  - Add a ½ second pause between each job:

```
for i in $LIST
do
  qsub job_$i
  usleep 500
done
```

  - Consider how many jobs will run at once: what is your group's PE limit? Will they like you when you submit 2000 jobs?

Some helpful environment variables

- `$PBS_O_WORKDIR` : the directory where you typed qsub
- `$PBS_JOBID` : the unique job id: e.g. 24461774.torx.ufhpc
- `$TMPDIR` : temporary directory for each job on compute node's local disk, good for jobs with lots of I/O
- `$PBS_NP` : Number of processors for single node job, use this when starting a threaded application to tell it how many processors to use. Prevents needing to change in multiple places. E.g. nodes=1:ppn=4 , `bastn -num_threads $PBS_NP`
- `$PBS_JOBNAME` : Name your gave your job with #PBS –N

So what is this "module" thing?

- **lmod**—Implementation of Environment Modules developed at TACC
- Allows easy management of user's environment

Lmod: Environmental Modules System

The standard way

```
PATH=$PATH:/some/long/path/to/application
export $PATH
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/long/path/to/
place/I/probably/cant/find
export $LD_LIBRARY_PATH
```

- Need to track down paths to applications, libraries, etc.
- Multiple compilers, and MPI implementations
- Manage dependencies
- Multiple versions of apps

The module way

- `module load trinity`
- Automatically:
  - Sets, `$HPC_TRINITY_DIR`
    - To run Inchworm, simply type
      `inchworm --reads reads.fa --run_inchworm [opts]`
  - Loads Bowtie and Allpaths, two Trinity dependencies
    - You don't need to hunt those down, or worry if they are in your path or not

Module discovery

- `module spider`
  - List everything
- `module spider cl`
  - List applications that have cl in name
- `module spider amber/12`
  - List details about this version of AMBER
- `module key molecular`
  - Keyword search for applications

Multiple versions

```
[magitz@submit1 ~]$ module spider gaussian
Rebuilding cache file, please wait ... done

----------------------------------------------------------------------------
gaussian:
----------------------------------------------------------------------------
  Description:
    A software for electronic structure modeling

  Versions:
     gaussian/e01
     gaussian/g03
     gaussian/g09

----------------------------------------------------------------------------
To find detailed information about gaussian please enter the full name.
For example:

   $ module spider gaussian/g09
----------------------------------------------------------------------------
```


Multiple variants of a version

```
[magitz@submit1 ~]$ module spider mrbayes/3.2.1
Rebuilding cache file, please wait ... Done
----------------------------------------------------------
mrbayes: mrbayes/3.2.1

----------------------------------------------------------
Description:
      Bayesian inference of phylogeny

    This module can be loaded directly: module load
mrbayes/3.2.1

    Additional variants of this module can also be loaded
after the loading the following modules:

       intel/2012, openmpi/1.6
```


Module loading

- `module load raxml`
- `module load intel raxml`
- `module load intel openmpi raxml`
- `module load intel/12 openmpi/1.6 raxml/3.2`
- `module unload raxml`

Module swapping

- `module load intel openmpi abyss`
- `module list`

```
Currently Loaded Modules:
  1) intel/2012  2) openmpi/1.6  3) abyss/default
```

- `module swap openmpi/1.6 openmpi/1.5.5`

```
Due to MODULEPATH changes the following modules
have been reloaded:
  1) abyss
```

Basic commands

- `module spider`
- `module spider gaussian`
- `module avail`
- `module list`
- `module load clustalw`
- `module load python/2.6.5`
- `module add intel openmpi`
- `module load intel/12 openmpi/1.6 mrbayes`
- `module del/rm/unload clustalw`

Let's look at some examples

- Examples of job scripts in:
  `/project/bio/training/2013-05-28/wordcloud/`
- Job scripts can have many commands
- qsub can pass variables into script with –v flag
  `qsub my_script.pbs -v FILE=f1.txt,OUT=outdir/out1.txt,SIZE=5`

UF Research Computing

- Help and Support
  - Help Request Tickets
    - https://support.hpc.ufl.edu
    - For any kind of question or help requests
    - Searchable database of solutions
  - We are here to help!
    - support@hpc.ufl.edu

UF Research Computing

- Help and Support (Continued)
  - http://wiki.hpc.ufl.edu
    - Documents on hardware and software resources
    - Various user guides
    - Many sample submission scripts
  - http://hpc.ufl.edu/support
    - Frequently Asked Questions
    - Account set up and maintenance