Optimizing Batch Submission and Job Performance
Matt Gitzendanner: magitz@ufl.edu

2/25/15

HiPerGator
The University of Florida Supercomputer for Research
- 16,384 cores (total of ~20,000 today)
- Infiniband interconnect
- >3PB fast, high-availability, storage

Cluster basics
User interaction
Scheduler
Compute resources
Login server (Head node)
Tell the scheduler what you want to do
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

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.log
  #PBS -e My_Job.err
  #PBS -l nodes=1:ppn=1
  #PBS -l pmem=900mb
  #PBS -l walltime=00:05:00
  cd $PBS_O_WORKDIR
date
  module load test_app
test_app -i file.txt

UF Research Computing
Tell the scheduler what you want to do
How do you know???

- Experience
- Trial and Error
- Run on a development node:
  ```
  $ ssh dev1
  $ module load my_app
  $ my_app -i file1.txt &
  $ top
  ```

How do you know???

- Start general and refine
  ```
  #PBS -M magitz@ufl.edu
  #PBS -m abe
  #PBS -l nodes=1:ppn=1
  #PBS -l pmem=4gb
  #PBS -l walltime=24:00:00
  ```

- Look at the ending or abort email for time and ram usage and adjust

End-of-job emails:

```
PBS Job Id: 358634.moab.ufhpc
Job Name: NR.25.nex
Exec host: c7a-s1/60
Execution terminated
Exit_status=0
resources_used.cput=07:16:09
resources_used.mem=251348kb
resources_used.vmem=318916kb
resources_used.walltime=07:16:52
```

How do you know???

- Common misconceptions
  - More cores (processors) will make my application run faster
  - More RAM will make my application run faster

  - The University of Florida Supercomputer for Research will run my application faster than my laptop

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

Walltime

```
#PBS -l walltime=00:50:00
```

- Backfill- use ~ accurate walltimes
- MAXPS: 3days*10^9investment
  - Long running jobs won’t start in “burst” capacity
- Last 3 days of use factors into priority

<table>
<thead>
<tr>
<th></th>
<th>Maximum</th>
<th>Short</th>
<th>Long</th>
</tr>
</thead>
<tbody>
<tr>
<td>Investor</td>
<td>31 days</td>
<td>&lt;12hrs</td>
<td>2 days</td>
</tr>
<tr>
<td>Other</td>
<td>7 days</td>
<td>&lt;12hrs</td>
<td>3 days</td>
</tr>
</tbody>
</table>
RAM

#PBS -l pmem=900mb

- Lots to consider, but do your best at estimating RAM needed for job
- Over about 4GB of RAM, "costs" toward CPU allocation

Wasted RAM leads to idle CPUs and low job throughput

Nodes and processors

Single processor apps:
#PBS -l nodes=1:ppn=1

Threaded (& MPI) apps:
#PBS -l nodes=1:ppn=4

MPI apps:
#PBS -l nodes=2:ppn=32

Nodes — Processors — "Cores"

A compute node or server

Each node has 4 processors

Most HiPerGator nodes have 64 cores or ppn and 256GB of RAM

Each processor has 16 "cores"

The scheduler uses "processors" where most think of "cores".

The processor request (ppn) is what most people think of as cores.

Processor Requests

- Is your application parallel?
- Can it use CPUs on multiple nodes?
- How well does it scale?

Passing variables in qsub

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

cd $PBS_O_WORKDIR

date
echo Running with input file: $infile
module load test_app
test_app -i $infile

qsub test.pbs -v infile=myfile1.txt

-v x=3,y=5

64 cores per node
- If RAM allows, MPI jobs under ~32 cores, should use nodes=1:ppn=##
- Some older nodes have 4-16 cores
Task Arrays

- #PBS -t 1-1000
  Runs 1000 tasks all submitted at once
- #PBS -t 1-1000%20
  Will throttle to run 20 tasks at a time
- $PBS_ARRAYID
  - #PBS -t 1-100%10
  - cd $PBS_O_WORKDIR
  - module load my_app
  - file=`ls *.txt | head -n $PBS_ARRAYID | tail -n 1`
  - my_app -i $file

Checking resources

- How many jobs are running in my group?
  - showq -w group=<group_name>
- How many resources will this job take?
  - pbs_info -f script_file.pbs
- Why isn’t this job running?
  - checkjob -v <job_id>

  - NOTES: job violates constraints for partition base
    - Job 46226XX violates active HARD MAXPE limit of 310
      for group XXXX partition ALL (Req: 1  InUse: 310))

Some helpful environment variables

- $PBS_O_WORKDIR: the directory where you typed qsub
- $PBS_JOBID: the unique job id: e.g. 24461774_moab.ufhp
- $TMPDIR: temporary directory for each job on compute node’s local disk, good for jobs with lots of small I/O
- $PBS_NUM_PPN: 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, blastn -num_threads $PBS_NUM_PPN
- $PBS_JOBNAME: Name your gave your job with #PBS -N