Troubleshooting Jobs on Odyssey

Bob Freeman, PhD
Research & Education Facilitator
XSEDE Campus Champion

with contributions from Paul Edmon, PhD
ITC Research Computing Associate
Goals

• Tackle PEND, FAIL, and slow performance issues
• Highlight different approaches to troubleshooting
• Arm you with useful SLURM and Unix commands

• Enable you to “Work smarter, better, faster”
PEND
PEND Problems

Why are you pending?

`squeue -t PD`

`squeue -t PD -o "%.18i %.9P %.8j %.8u %.2t %.10M %.6D %R %C"`
## PEND Problems

Why are you pending? Most common reasons are...

<table>
<thead>
<tr>
<th>Reason</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>Scheduler hasn’t gotten around to you yet. Hang tight...</td>
</tr>
<tr>
<td>Resources</td>
<td>The job is waiting for resources to become available</td>
</tr>
<tr>
<td>Priority</td>
<td>One or more higher priority jobs exist for this partition or reservation. Priority is influenced by age, job size, partition, QoS, and Fair-share.</td>
</tr>
<tr>
<td>Dependency</td>
<td>This job is waiting for a dependent job to complete (--dependency)</td>
</tr>
</tbody>
</table>
PEND Problems: Resources

- What did you request?
- What are the parameters of your SLURM submission script?

```
scontrol show jobid -dd JOBID
```

```
$ scontrol show jobid -dd 25889308
JobId=25889308 Name=p3_30_13
  
  SchedId=estrogen1.mq.nre.jrcnam06-00-01
  
  Priority=100000024 Nice=0 Account=hernquist_lab QOS=normal
  
  JobState=PENDING Reason=Resources Dependency=(null)
  
  Requeue=1 Restarts=0 BatchFlag=1 ExitCode=0:0
  
  DerivedExitCode=0:0
  
  RunTime=00:00:00 TimeLimit=7-00:00:00 TimeMin=N/A
  
  
  StartTime=2014-12-03T08:47:21 EndTime=Unknown
  
  PreemptTime=None SuspendTime=None SecsPreSuspend=0
  
  Partition=vogelsberger AllocNode:Sid=holy2a24208:48180
  
  ReqNodeList=(null) ExcNodeList=(null)
  
  NodeList=(null)
  
  NumNodes=1-1 NumCPUs=64 CPUs/Task=1
  
  Req:S:C:T=0:0:*:*
  
  Socks/Node="" MallocType Basic=0:0:*:* CoreSpec=0
  
  MinCPUsNode=1 MinMemoryNode=250000M MinTmpDiskNode=0
  
  Features=(null) Gres=(null) Reservation=(null)
  
  Shared=0 Contiguous=0 Licenses=(null) Network=(null)
  
  Command=/n/home01/ptorrey/Runc/NuclearDICE (GIZMO, PERN, PEBBLE)'
```
What is your priority in the scheduling queue?

- showq-slurm -U
- showq-slurm -p PARTITION -o

<table>
<thead>
<tr>
<th>JOBID</th>
<th>JOBNAME</th>
<th>USERNAME</th>
<th>STATE</th>
<th>CORE</th>
<th>WCLIMIT</th>
<th>QUEUETIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>25912197</td>
<td>4002_b3lyp</td>
<td>hirzel</td>
<td>Waiting 8</td>
<td>75:00:00</td>
<td>Tue Dec 2 15:29:48</td>
<td></td>
</tr>
<tr>
<td>25911730</td>
<td>4027_b3lyp</td>
<td>rgbombarelli</td>
<td>Waiting 8</td>
<td>75:00:00</td>
<td>Tue Dec 2 15:26:41</td>
<td></td>
</tr>
<tr>
<td>25911735</td>
<td>4028_b3lyp</td>
<td>rgbombarelli</td>
<td>Waiting 8</td>
<td>75:00:00</td>
<td>Tue Dec 2 15:26:43</td>
<td></td>
</tr>
<tr>
<td>25911739</td>
<td>4029_b3lyp</td>
<td>rgbombarelli</td>
<td>Waiting 8</td>
<td>75:00:00</td>
<td>Tue Dec 2 15:26:44</td>
<td></td>
</tr>
<tr>
<td>25911743</td>
<td>4030_b3lyp</td>
<td>rgbombarelli</td>
<td>Waiting 8</td>
<td>75:00:00</td>
<td>Tue Dec 2 15:26:46</td>
<td></td>
</tr>
<tr>
<td>25911746</td>
<td>4031_b3lyp</td>
<td>rgbombarelli</td>
<td>Waiting 8</td>
<td>75:00:00</td>
<td>Tue Dec 2 15:26:48</td>
<td></td>
</tr>
<tr>
<td>25911817</td>
<td>4042_b3lyp</td>
<td>rgbombarelli</td>
<td>Waiting 8</td>
<td>75:00:00</td>
<td>Tue Dec 2 15:27:16</td>
<td></td>
</tr>
<tr>
<td>25911826</td>
<td>4043_b3lyp</td>
<td>rgbombarelli</td>
<td>Waiting 8</td>
<td>75:00:00</td>
<td>Tue Dec 2 15:27:18</td>
<td></td>
</tr>
<tr>
<td>25911828</td>
<td>4044_b3lyp</td>
<td>rgbombarelli</td>
<td>Waiting 8</td>
<td>75:00:00</td>
<td>Tue Dec 2 15:27:19</td>
<td></td>
</tr>
<tr>
<td>25911834</td>
<td>4045_b3lyp</td>
<td>rgbombarelli</td>
<td>Waiting 8</td>
<td>75:00:00</td>
<td>Tue Dec 2 15:27:22</td>
<td></td>
</tr>
<tr>
<td>25911838</td>
<td>4046_b3lyp</td>
<td>rgbombarelli</td>
<td>Waiting 8</td>
<td>75:00:00</td>
<td>Tue Dec 2 15:27:23</td>
<td></td>
</tr>
<tr>
<td>25912198</td>
<td>run_getSco</td>
<td>inun</td>
<td>Waiting 1</td>
<td>8:20:00</td>
<td>Tue Dec 2 15:29:48</td>
<td></td>
</tr>
<tr>
<td>25820732</td>
<td>CHEMBL1818</td>
<td>ncheron</td>
<td>Waiting 16</td>
<td>48:00:00</td>
<td>Tue Dec 2 05:15:18</td>
<td></td>
</tr>
<tr>
<td>25820733</td>
<td>CHEMBL1818</td>
<td>ncheron</td>
<td>Waiting 16</td>
<td>48:00:00</td>
<td>Tue Dec 2 05:18:08</td>
<td></td>
</tr>
<tr>
<td>25820736</td>
<td>CHEMBL1818</td>
<td>ncheron</td>
<td>Waiting 16</td>
<td>48:00:00</td>
<td>Tue Dec 2 05:15:21</td>
<td></td>
</tr>
<tr>
<td>25820737</td>
<td>CHEMBL1818</td>
<td>ncheron</td>
<td>Waiting 16</td>
<td>48:00:00</td>
<td>Tue Dec 2 05:17:49</td>
<td></td>
</tr>
<tr>
<td>25820738</td>
<td>CHEMBL1818</td>
<td>ncheron</td>
<td>Waiting 16</td>
<td>48:00:00</td>
<td>Tue Dec 2 05:17:48</td>
<td></td>
</tr>
<tr>
<td>25879921</td>
<td>4226_b3lyp</td>
<td>rgbombarelli</td>
<td>Waiting 8</td>
<td>75:00:00</td>
<td>Tue Dec 2 10:30:37</td>
<td></td>
</tr>
<tr>
<td>25879923</td>
<td>4227_b3lyp</td>
<td>rgbombarelli</td>
<td>Waiting 8</td>
<td>75:00:00</td>
<td>Tue Dec 2 10:30:40</td>
<td></td>
</tr>
<tr>
<td>25879925</td>
<td>4228_b3lyp</td>
<td>rgbombarelli</td>
<td>Waiting 8</td>
<td>75:00:00</td>
<td>Tue Dec 2 10:30:42</td>
<td></td>
</tr>
</tbody>
</table>
What is a Fair-share score?
- Score assigned to each lab & affects scheduling priority of jobs
- User’s current and past usage is considered when determining the scheduling of job execution

How is it calculated?
- Score is based on usage and shares:
  - \( 0 \leq \text{Usage} \leq 1 \), represents your proportional use of Odyssey
  - Shares \( \sim \) slices of a pie, or the part of Odyssey that is yours

Premise: Usage \( \equiv \) Shares, you’ve hit your Fair-share target

\[
\text{Fairshare Factor} = 2^{-\frac{\text{Usage}}{\text{Shares}}} \quad 0 < \text{FF} < 1
\]

When usage increases, FF decreases
When usage decreases, FF increases
What is my score?

```
sshare -U
sshare --acount=freeman_lab -a
```

<table>
<thead>
<tr>
<th>Account</th>
<th>User Raw Shares</th>
<th>Norm Shares</th>
<th>Raw Usage</th>
<th>Effectv Usage</th>
<th>FairShare</th>
</tr>
</thead>
<tbody>
<tr>
<td>root</td>
<td>1.000000</td>
<td>1.000000</td>
<td>7537869166</td>
<td>0.000000</td>
<td>0.500000</td>
</tr>
<tr>
<td>ac275</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0</td>
<td>0.000000</td>
<td>1.000000</td>
</tr>
<tr>
<td>acmg</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0</td>
<td>0.000000</td>
<td>1.000000</td>
</tr>
<tr>
<td>adams_lab_seas</td>
<td>0.000000</td>
<td>0.000000</td>
<td>2384</td>
<td>0.000000</td>
<td>0.999927</td>
</tr>
<tr>
<td>aidala_lab</td>
<td>0.000000</td>
<td>0.000000</td>
<td>9645</td>
<td>0.000001</td>
<td>0.999705</td>
</tr>
<tr>
<td>airoldi_lab</td>
<td>0.000000</td>
<td>0.000000</td>
<td>6173473</td>
<td>0.000820</td>
<td>0.828016</td>
</tr>
<tr>
<td>aizenberg_lab</td>
<td>0.000000</td>
<td>0.000000</td>
<td>12</td>
<td>0.000000</td>
<td>1.000000</td>
</tr>
<tr>
<td>amir_lab</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0</td>
<td>0.000000</td>
<td>1.000000</td>
</tr>
<tr>
<td>anderson_lab</td>
<td>0.000000</td>
<td>0.000000</td>
<td>188069</td>
<td>0.0000025</td>
<td>0.994267</td>
</tr>
<tr>
<td>anl</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0</td>
<td>0.000000</td>
<td>1.000000</td>
</tr>
<tr>
<td>arlotta_lab</td>
<td>0.000000</td>
<td>0.000000</td>
<td>478</td>
<td>0.000000</td>
<td>0.999985</td>
</tr>
<tr>
<td>aspuru-guzik_lab</td>
<td>0.000000</td>
<td>0.000000</td>
<td>1049842986</td>
<td>0.139265</td>
<td>0.040566</td>
</tr>
<tr>
<td>ay215</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0</td>
<td>0.000000</td>
<td>1.000000</td>
</tr>
<tr>
<td>aziz_lab</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0</td>
<td>0.000000</td>
<td>1.000000</td>
</tr>
<tr>
<td>baccarelli_lab</td>
<td>0.000000</td>
<td>0.000000</td>
<td>193902</td>
<td>0.0000026</td>
<td>0.994090</td>
</tr>
<tr>
<td>balskus_lab</td>
<td>0.000000</td>
<td>0.000000</td>
<td>6316</td>
<td>0.000001</td>
<td>0.999807</td>
</tr>
<tr>
<td>berger_lab</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0</td>
<td>0.000000</td>
<td>1.000000</td>
</tr>
<tr>
<td>bertoldi_lab</td>
<td>0.000000</td>
<td>0.000000</td>
<td>7748573</td>
<td>0.0001017</td>
<td>0.791765</td>
</tr>
</tbody>
</table>

```
[bfreeman@rclogin07 ~]$ sshare -u bfreeman | grep bfreeman
rc_admin  bfreeman  parent  0.003012  20442  0.000014  0.996721
```

March 8, 2016
FAIL
FAIL Problems

Determining the root cause is highly dependent on your ability to trace & document the problem

• Do you have log/error files?
  
  #SBATCH -o job_%j.stdout.txt
  #SBATCH -e job_%j.stderr.txt

• Did you use the -e parameter to get an error file?
  If omitted, STDERR is redirected to -o file or slurm-JOBID.out

• Is the path non-existent?
  The path to place the output files must exist! Otherwise, SLURM won’t know what to do and will do nothing but FAIL.

• What else is going on in your SLURM submission script?
FAIL Problems

• Look at the fail trail in any log and/or error files.
• *Nota bene!* The last error listed may not be the root cause!
FAIL Problems

Other things may be going on...

- Segmentation faults
- Shared/static library errors

- Are the correct software packages and versions loaded?
- Are there version conflicts?
- Are too many things loaded in your .bashrc?
SLOOOWWW...
Much harder to diagnose!

- Transient issues:
  - other jobs on the node
  - network
  - storage
  - bad node

- Software install issue
- Code compile/optimization problem
## Poor Performance

Slow run times? Most common reasons are...

<table>
<thead>
<tr>
<th>Reason</th>
<th>Diagnostics &amp; Potential solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SLURM submission errors</strong></td>
<td>Check <code>--mem/--mem-per-cpu</code></td>
</tr>
<tr>
<td></td>
<td>Check cores (-n) + nodes (-N)</td>
</tr>
<tr>
<td></td>
<td>Try running job interactively</td>
</tr>
<tr>
<td><strong>Overloaded Storage</strong></td>
<td>Is the mount point broken?</td>
</tr>
<tr>
<td></td>
<td>Is the storage being hammered?</td>
</tr>
<tr>
<td></td>
<td>Don’t run out of home/lab directories.</td>
</tr>
<tr>
<td></td>
<td>Is one file being accessed by hundreds of jobs? Make copies.</td>
</tr>
<tr>
<td></td>
<td>Too many files in one directory?</td>
</tr>
<tr>
<td></td>
<td>Check <a href="http://stats.rc.fas.harvard.edu">http://stats.rc.fas.harvard.edu</a></td>
</tr>
<tr>
<td><strong>Overloaded Node</strong></td>
<td>Too much Input/Output for the given network interface</td>
</tr>
<tr>
<td></td>
<td>Overuse of CPUs relative to SLURM request</td>
</tr>
<tr>
<td></td>
<td>Check <a href="http://stats.rc.fas.harvard.edu">http://stats.rc.fas.harvard.edu</a>; use <code>squeue</code> and <code>top</code></td>
</tr>
<tr>
<td><strong>Sick Node</strong></td>
<td>Memory errors</td>
</tr>
<tr>
<td></td>
<td><code>/scratch</code> filled up</td>
</tr>
<tr>
<td></td>
<td>Use <code>lsload</code> to look at node diagnostics</td>
</tr>
<tr>
<td><strong>Sick Code</strong></td>
<td>Monitor/trace program execution with <code>strace</code></td>
</tr>
<tr>
<td></td>
<td>Look at code performance with <code>perf</code> or <code>perftrace</code></td>
</tr>
</tbody>
</table>
Poor Performance: Cluster Status

Check the state of the cluster at [https://status.rc.fas.harvard.edu](https://status.rc.fas.harvard.edu)

- Broken into geographic regions
- Also lists past incidents
Poor Performance: Cluster Stats

Check the state of the systems at https://stats.rc.fas.harvard.edu

Heavy load, but normal

Heavy load, oversubscribed
Poor Performance: Nagios Storage

Includes storage like /n/regal
Poor Performance: Node health

Check the health of the compute node (abused by other jobs?)

• need to figure out which compute node your job is on
  
  `squeue -j JOBID`

• quick check using [https://stats.rc.fas.harvard.edu](https://stats.rc.fas.harvard.edu)

• look at compute node/cluster load using `lsload`

• look at jobs on node using `squeue`

• log in to node via ssh and profile
  
  o `top`
  
  o `strace`
  
  o `perftrace`
Poor Performance: Node health

Use `lsload` to examine the utilization of individual nodes

```
[bfreeman@rclogin05 ~]$ lsload

<table>
<thead>
<tr>
<th>Hostname</th>
<th>Cores</th>
<th>InUse</th>
<th>Ratio</th>
<th>Load</th>
<th>Mem</th>
<th>Alloc</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>aag01</td>
<td>64</td>
<td>64</td>
<td>100.0</td>
<td>63.68</td>
<td>258</td>
<td>208</td>
<td>ALLOCATED</td>
</tr>
<tr>
<td>aag02</td>
<td>64</td>
<td>64</td>
<td>100.0</td>
<td>64.20</td>
<td>258</td>
<td>184</td>
<td>ALLOCATED</td>
</tr>
<tr>
<td>aag03</td>
<td>64</td>
<td>56</td>
<td>87.5</td>
<td>58.89</td>
<td>258</td>
<td>120</td>
<td>MIXED+COMPLETING</td>
</tr>
<tr>
<td>aag04</td>
<td>64</td>
<td>56</td>
<td>87.5</td>
<td>62.11</td>
<td>258</td>
<td>172</td>
<td>MIXED+COMPLETING</td>
</tr>
<tr>
<td>holy2a01201</td>
<td>64</td>
<td>64</td>
<td>100.0</td>
<td>64.00</td>
<td>264</td>
<td>181</td>
<td>ALLOCATED</td>
</tr>
<tr>
<td>holy2a01202</td>
<td>64</td>
<td>64</td>
<td>100.0</td>
<td>54.34</td>
<td>264</td>
<td>250</td>
<td>ALLOCATED</td>
</tr>
<tr>
<td>holy2a01203</td>
<td>64</td>
<td>62</td>
<td>96.9</td>
<td>63.45</td>
<td>264</td>
<td>210</td>
<td>MIXED</td>
</tr>
<tr>
<td>holy2a01204</td>
<td>64</td>
<td>0</td>
<td>0.0</td>
<td>0.20</td>
<td>264</td>
<td>0</td>
<td>IDLE+DRAIN</td>
</tr>
</tbody>
</table>

014-11-23_mem_error exists [root@2014-12-02T20:24:01] ERROR Health check failed: Memory Check failed!
```

March 8, 2016
What other jobs are running on my node?

`squeue -w NODENAME`

e.g. Ensure that the # of CPUs requested == # CPUs used
Poor Performance

Log in to node and top

```bash
ssh USERNAME@NODENAME
top
```
Poor Performance

Log in to node and monitor code execution

```
ssh RCUSERNAME@NODENAME
ps aux | grep RCUSERNAME
```

(obtain relevant process ID)

```
strace -p PROCESSID
```

![Code execution output](image)
Goals

Back to Basics
Basic Troubleshooting

Before seeking help, take some basic steps to ascertain what is going on with your job:

• Use `squeue` and `sacct` with `--format=` option to query details from SLURM
  • Are you having Fairshare issues (Priority)?
  • Is your job waiting for space (Resources)?
  • Will your job ever run (Dependency)?
  • Is there an error code or message

• Check your log files
  • You did specify both `-o` and `-e`, yes?
  • No log files? Does the path to your log files exist before the job start?
  • Message about Pre-emption, Timeout, Or Failure?
  • The last error in the log is usually not the problem. The first one is!

• Did you request e-mail messages for your jobs with `--mail-type=`?
• Is your SLURM script formatted properly?
• Are you loading legacy modules? Possible software/library conflicts?
• Make liberal use of echo statements in your SLURM submission script
• Use `set –x` after module loads to trace bash execution of your SLURM script
• Make use of advanced `sbatch` options (`--contiguous`, `--hint`, `--distribution`)

Check out Tips@12 presentation [http://fasrc.us/fasrcmaterials](http://fasrc.us/fasrcmaterials)
Problems, Pitfalls, and Prevention

This is a shared resource, so everyone has skin in the game. And you can help us and yourself...

• Node and cluster problems are not unusual, esp. as large as system as Odyssey: I/O errors, node failures, memory errors, etc. Let us know if you see these.

• Review our Usage & Responsibilities guidelines: http://fasrc.us/hpccustoms

• Review our Common Pitfalls, lest you fall victim: http://fasrc.us/hpcpitfalls

Don’t use multiple cores for R and Python scripts
  These interpreters/runtime environments are can one use 1 core. Don’t waste please.

PEND for >48 hrs
  Asking for very large resource requests (cores/memory);very low Fairshare score

Quick run and FAIL...Not including -t parameter
  no -t means shortest possible in all partitions == 10 min

Asking for multiple cores but forgetting to specify one node
  -n 4 -N 1 is very different from -n 4

Not specifying enough cores
  prog1 | prog2 | prog3 > outfile should run with 3 cores

Causing massive disk I/O on home folders/lab disk shares
  your work & others on the same filesystem slows to a crawl; simple commands like ls take forever
RC regularly reviews jobs based on their effective usage of their SLURM reservations (cores, memory, time, disk, ...) to promote maximum utilization of these resources.

- Over-requesting resources negatively affects the scheduling priority of your own jobs and blocks other users from these resources, which further lowers the overall research output for all HU users.
- Under-requesting resources negatively affects your job and those running on the same nodes; and potentially other jobs on the same filesystem.

You may be contacted if you are regularly are having issues with your job efficiency and we will work with you to improve your performance.

Can calculate the CPU efficiencies with the following formula:

```
sacct -u RCUSERNAME --format=user,state,jobid,alloccpus,elapsed,cputime
EffCPUs = CPUTime / Elapsed
%Eff = CPUTime / (AllocCPUs * Elapsed)
```
Please talk to your peers, and
We wish you success in your research!

http://rc.fas.harvard.edu
https://portal.rc.fas.harvard.edu
rchelp@fas.harvard.edu
@fasrc

Harvard Informatics
@harvardifx