Upcoming Biowulf Seminars
Bldg 50, Rm 1227

• November 30, 1 - 3 pm
  Python in HPC
  Overview of python tools used in high performance computing, and
  how to improve the performance of your python jobs on Biowulf

• Jan 16, 1 - 3 pm
  Relion tips and tricks, and Parallel jobs and benchmarking
  Mechanics and best practices for submitting RELION jobs to the batch
  system from both the command line and via the RELION GUI, as well
  as methods for monitoring and evaluating the results. Scaling of
  parallel jobs, how to benchmark to make effective use of your
  allocated resources
Making Effective Use of the Biowulf Batch System

Steven Fellini
staff@hpc.nih.gov
NIH HPC Staff, CIT
Oct 30, 2017
Effective Use == Effective Resource Allocation

• Specifying resources
  • Estimating required resources
  • Allocating resources with sbatch and swarm

• Monitoring resource allocation
• Scheduling and resource allocation
• Post-mortem analysis
Hardware Terminology Review

Socket
Receptacle on the motherboard for one physically packaged processor (each of which can contain one or more cores).

Core
A complete private set of registers, execution units, and retirement queues needed to execute programs.

COPUs
One or more hardware contexts within a single core. Each CPU has attributes of one core; managed & scheduled as a single logical processor by the OS.

Node

Hyper-threading
Estimating Resources

• CPU
  • Check documentation (https://hpc.nih.gov/apps/)
  • Objective -- match CPU:Threads 1:1
    (there are exceptions, e.g., MD jobs)

• Memory
  • Run a job or swarm with a large memory allocation
  • Check actual memory usage
  • Add 10% to actual memory usage

• Time
  • Run a job or swarm with a large time allocation
  • Check actual wall time
  • Add 10% to actual wall time
Allocating Resources with *sbatch* and *swarm*

- All jobs
  - `--mem (sbatch) or -g (swarm)`
  - `--time (sbatch and swarm)`
  - `-b to bundle command lines (swarm)`

- Single-threaded jobs
  - “-p 2” to load cores with 2 threads (swarm)

- Multi-threaded jobs
  - `--cpus-per-task (sbatch) or -t (swarm)`
  - Use `$SLURM_CPUS_PER_TASK` in batch script
  - `OMP_NUM_THREADS`

- Multi-node jobs
  - “Parallel Jobs and Benchmarking” Jan 16
Monitoring Resource Allocation

• CPU
  • jobload while the job is running
  • Dashboard during or after the job
  • (No easy way to monitor GPU utilization at the moment)

• Memory
  • jobload while the job is running
  • jobhist, Dashboard or sacct during or after the job has completed

• Walltime
  • jobhist, Dashboard or sacct during or after the job has completed
## jobload

% jobload -u someuser

<table>
<thead>
<tr>
<th>JOBID</th>
<th>TIME</th>
<th>NODES</th>
<th>CPUS</th>
<th>THREADS</th>
<th>LOAD</th>
<th>MEMORY</th>
</tr>
</thead>
<tbody>
<tr>
<td>51863534</td>
<td>2012-08-10 08:01</td>
<td>cn3095</td>
<td>4</td>
<td>4</td>
<td>100%</td>
<td>1.0 / 8.0 GB</td>
</tr>
<tr>
<td>51863535</td>
<td>2012-08-10 08:01</td>
<td>cn3256</td>
<td>4</td>
<td>5</td>
<td>125%</td>
<td>0.9 / 8.0 GB</td>
</tr>
<tr>
<td>51863536</td>
<td>2012-08-10 08:01</td>
<td>cn3348</td>
<td>4</td>
<td>1</td>
<td>25%</td>
<td>1.0 / 8.0 GB</td>
</tr>
<tr>
<td>51863537</td>
<td>2012-08-10 08:01</td>
<td>cn3401</td>
<td>4</td>
<td>3</td>
<td>75%</td>
<td>0.9 / 8.0 GB</td>
</tr>
<tr>
<td>51881591</td>
<td>2012-08-10 19:16</td>
<td>cn3097</td>
<td>4</td>
<td>4</td>
<td>100%</td>
<td>1.0 / 8.0 GB</td>
</tr>
</tbody>
</table>

% jobload -j 51874438_233

<table>
<thead>
<tr>
<th>JOBID</th>
<th>TIME</th>
<th>NODES</th>
<th>CPUS</th>
<th>THREADS</th>
<th>LOAD</th>
<th>MEMORY</th>
</tr>
</thead>
<tbody>
<tr>
<td>51874438_233</td>
<td>2012-08-10 19:16</td>
<td>cn3105</td>
<td>2</td>
<td>1</td>
<td>50%</td>
<td>0.5 / 1.5 GB</td>
</tr>
<tr>
<td>Jobid</td>
<td>Partition</td>
<td>State</td>
<td>Nodes</td>
<td>CPUs</td>
<td>Walltime</td>
<td>Runtime</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------</td>
<td>---------</td>
<td>-------</td>
<td>------</td>
<td>----------</td>
<td>---------</td>
</tr>
<tr>
<td>52102264_67</td>
<td>norm</td>
<td>COMPLETED</td>
<td>1</td>
<td>2</td>
<td>02:00:00</td>
<td>00:05:29</td>
</tr>
</tbody>
</table>
```
% sacct --format=Jobname,AllocCPUS,AllocNodes,ReqMem,MaxRSS,Elapsed -j 52102332
JobName    AllocCPUS AllocNodes  ReqMem  MaxRSS  Elapsed
----------  --------  -----------  -------  --------  --------
tbss_2_reg  2        1            4Gn     00:05:29
batch      2        1            4Gn  815152K  00:05:29
```
Using Your Dashboard to Monitor Jobs

https://hpc.nih.gov

https://hpc.nih.gov/dashboard/
Scheduling and Resource Allocation

• Scheduling is determined by job priority
• Priority is determined by Fairshare value of user
• Fairshare is determined by recent cpu and memory allocations of running jobs

• Unnecessarily long time allocation will prevent jobs from being backfilled
Why are my jobs pending?

- ‘freen’ shows free CPUs but not free memory or disk
- Other jobs have higher priority (sprio)
- Nodes are reserved for higher-priority jobs
Consequences of...

<table>
<thead>
<tr>
<th></th>
<th>Specifying more resources than needed</th>
<th>Specifying fewer resources than needed</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CPU</strong></td>
<td>Wasted CPU resources, possibly unnecessary scheduling delays</td>
<td>Job runs a little/a lot slower</td>
</tr>
<tr>
<td><strong>Memory</strong></td>
<td>Wasted memory resources, possibly unnecessary scheduling delays</td>
<td>Job is “Killed” by the kernel</td>
</tr>
<tr>
<td><strong>Time</strong></td>
<td>Possibly unnecessary scheduling delays</td>
<td>Job is killed by the batch system</td>
</tr>
</tbody>
</table>
Post-mortem of jobs using user Dashboard

Or

The Good, the Bad, and the Ugly...
Comment: job is running with default allocations for CPU and memory
Recommendation: if a subjob of a large swarm, try “-p 2”
Another A+
Recommendation: reduce CPU allocation
Comment: perfect CPU utilization; underutilized memory but entire node is allocated due to cpu allocation
Comment: good overall utilization; possibly split into two jobs with a dependency, and with differing resource allocations
Comment: 8 CPUs too little/too much, 2 would do
Recommendation: could run in half the memory
The memory use likely exceeded memory allocation.
Comment: good memory utilization
Recommendation: increasing CPU allocation probably won’t help
Comment: CPUs badly overloaded
Recommendation: could run in less than half the memory
Comment: CPUs overloaded 200%
Recommendation: 256 GB memory allocated, MBs used
Comment: good memory utilization
Recommendation: might run faster with 32 CPUs?
Recommendation: increasing CPU count might improve?
Recommendation: allocating 56 CPUs would likely help
staff@hpc.nih.gov