Running Jobs on Genepool

Douglas Jacobsen
NERSC Bioinformatics Computing Consultant

February 12, 2013
Structure of the Genepool System

User Access
- Command Line
- Scheduler
- Service

- ssh genepool.nersc.gov

- web services
- database services
- filesystems

- login nodes
- gpint nodes
- high priority & interactive nodes
- fpga

User Access
- Command Line
- Scheduler
- Service

- ssh genepool.nersc.gov

- web services
- database services
- filesystems

- login nodes
- gpint nodes
- high priority & interactive nodes
- fpga
Types of Jobs on genepool

• **Batch – Scheduled** (compute nodes, fpga)
  - 8,320 cores for 72,953,280 compute hours per year in genepool
  - use “qsub” to submit a job

• **Interactive – Scheduled** (compute nodes subset)
  - 80 cores presently, increasing size
  - use “qlogin” to submit a job

• **Interactive – Unscheduled** (login nodes, gpints)
  - 4 login nodes, 27 gpint nodes
  - ssh to the host, direct-use

• **Services – Unscheduled** (login nodes, gpints, gpweb, gpdb, gpodb)
  - Web services
  - Database services
  - Automated job submission / control
Basics of Batch Jobs

• Genepool is a shared resource
• Each calculation usually only takes a small portion of genepool
  – Every job is strictly limited on the consumption of genepool resources
  – The job description specifies the resource limits
• Univa GridEngine is used to schedule each calculation on genepool
  – The scheduler matches job resource limit requests with physical resources
Basics of GridEngine

- **GridEngine schedules “slots”**
  - Not memory, nor processors, nor nodes
- **A slot is a portion of a node**
  - For most nodes on genepool, a slot is defined as a single processor plus \( \frac{\text{ram.c}_{\text{nodeTotal}}}{\text{n.cores}} \) memory
  - Some nodes are *exclusively scheduled* – all slots on the node are bonded together as one schedulable unit
- **Jobs are placed in queues**
  - Queues manage the resources of disparate sets of nodes, and have distinct resource limits
    - normal.q has a 12 hour time limit
    - long.q has a 10 day time limit
- **Jobs are scheduled in order of a balance of:**
  - Resource availability
  - Job prioritization
## Compute Node Hardware

<table>
<thead>
<tr>
<th>Count</th>
<th>Cores</th>
<th>Slots</th>
<th>Scheduleable Memory</th>
<th>Memory/Slot</th>
<th>Interconnects</th>
</tr>
</thead>
<tbody>
<tr>
<td>515</td>
<td>8</td>
<td>8</td>
<td>42G</td>
<td>5.25G</td>
<td>1Gb Ethernet</td>
</tr>
<tr>
<td>220</td>
<td>16</td>
<td>16</td>
<td>120G</td>
<td>7.5G</td>
<td>14x FDR Infiniband</td>
</tr>
<tr>
<td>8</td>
<td>24</td>
<td>24</td>
<td>252G</td>
<td>10.5G</td>
<td>1G Ethernet</td>
</tr>
<tr>
<td>9</td>
<td>32</td>
<td>32</td>
<td>500G</td>
<td>15.625G</td>
<td>4 have 10G Eth</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5 have Infiniband</td>
</tr>
<tr>
<td>3</td>
<td>32</td>
<td>32</td>
<td>1000G</td>
<td>31.25G</td>
<td>1 has 10G Eth</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2 have Infiniband</td>
</tr>
<tr>
<td>1</td>
<td>80</td>
<td>64</td>
<td>2000G</td>
<td>31.25G</td>
<td>10G Ethernet</td>
</tr>
</tbody>
</table>

- The 42G nodes are scheduled “by slot”
  - Multiple jobs can run on the same node at the same time
- Higher memory nodes are exclusively scheduled
  - Only one job can run at a time on a node
Basics of Batch Job Submission

Example Batch Script

```bash
#!/bin/bash
module load blast+
input=$1
database=$2
blastn -query $input -db $database <more options>
```

Submitting the example

genepool$ qsub -cwd example.sh queries.fa myDB
Your job 347283 ("example.sh") has been submitted.

- "qsub" submits the job for batch processing
- "-cwd" directs the job to work out of the present location in the filesystem
  - the current working directory
- Default resource limits will be applied, since none were specified
  - 1 slot
  - 5.25GB memory-slot
  - 12 hours
Resource Limits Request User Interface

• **Basic Resources:**
  
  – Cores/Processors
    
    • Default: 1 slot (unspecified pe)
    
    • `-pe pe_slots n` (n cores on a single machine, e.g. for threaded job)
    
    • `-pe pe_m n*m` (n nodes with m cores per node, e.g. for MPI job)
  
  – Memory – in units of memory/core
    
    • Default: 5.25G (if unspecified)
    
    • `-l ram.c=42G` (request 42 gigabytes / core)
    
    • `-l ram.c=500` (request 500 bytes / core)

    At present memory enforcement is on virtual memory; so the entire virtual memory requirements of your job must be considered!

  – Time – specified as HH:MM:SS or in seconds
    
    • Default: 12:00:00 (12 hours)
    
    • `-l h_rt=2:00:00` (request 2 hours)
    
    • `-l h_rt=300` (request 300s = 5 minutes)
• Additional Resources that can be specified:
  – -l exclusive.c request an exclusive node
  – -l infiniband.c future: specify a node with infiniband

• User-Requestable Queues
  – -l high.c OR -q high.q
    • put job in high-priority queue
## Queues on Genepool

<table>
<thead>
<tr>
<th>Queue Name</th>
<th>Walltime Limit</th>
<th>Nodes* (Slots)</th>
<th>Slot Limits</th>
<th>Memory/slot</th>
<th>Other Limits</th>
</tr>
</thead>
<tbody>
<tr>
<td>normal.q</td>
<td>12:00:00</td>
<td>443 (3544)</td>
<td>None</td>
<td>5.25G</td>
<td>N/A</td>
</tr>
<tr>
<td>long.q</td>
<td>240:00:00</td>
<td>70 (560)</td>
<td>320 per user</td>
<td>5.25G</td>
<td>N/A</td>
</tr>
<tr>
<td>normal_excl.q</td>
<td>12:00:00</td>
<td>170 (2720)</td>
<td>None</td>
<td>7.5G</td>
<td>Whole-node scheduling</td>
</tr>
<tr>
<td>long_excl.q</td>
<td>240:00:00</td>
<td>50 (750)</td>
<td>None</td>
<td>7.5G</td>
<td>Whole-node scheduling</td>
</tr>
<tr>
<td>high.q</td>
<td>240:00:00</td>
<td>10 (80)</td>
<td>8 per user</td>
<td>15G</td>
<td>N/A</td>
</tr>
</tbody>
</table>

* These numbers do not count the high-memory resources. All the high memory resources are in both normal_excl.q and long_excl.q
Submitting Jobs: Mapping Resources to Slots

• User interface is focused on machine resources required: cores, memory/core, time
• GridEngine is best able to schedule uniform-sized slots per machine-class
• NERSC automatically “re-shapes” your request to get optimally scheduled:

  ```
  qsub -l ram.c=40G myScript.sh
  
  qsub -l ram.c=5.25G -pe pe_slots 8 myScript.sh
  ```

  Total memory automatically inflated to 42G
Job Prioritization: Fair Share

• Genepool was originally created by merging together a variety of legacy systems
• Each group was assigned a “share” proportional to its contribution to genepool
• GridEngine tries to ensure that each group on average uses just that share
  – When the system is idle, any group can use the whole cluster
• If you belong to multiple projects, make sure you attribute the job to the correct project with:
  – qsub -P <project>.p ...

Job Submission Recommendations

• If at all possible use 12 hours or less!
  – The long queue has few nodes, and usage is constrained

• Requesting more than 42G results in getting an exclusive node
  – Unless you need the new nodes, this can significantly drain your project’s share

• Do specify –cwd or –wd <directory> with qsub
  – Writing output to your home directory (the default) from the cluster en masse can slow everybody down

• Specify a meaningful name for your job
  – qsub -N eColi_BlastSeg11 will make things easier on you later as you try to monitor your jobs or pick up the pieces after a crash
Resource Limits Request User Interface

• Examples:
  – Number of “slots” – effectively processors for most of genepool
    • Request 8 processors on one node
      – genepool$ qsub -pe pe_slots 8 ...
      – genepool$ qsub -pe pe_8 8 ...
    • Request two 16-processor nodes
      – genepool$ qsub -pe pe_16 32 ...
  – Memory per slot
    – genepool$ qsub -l ram.c=8G ...
  – “walltime” limit, total execution time limit
    – genepool$ qsub -l h_rt=5:00:00 ...