How do I use a cluster using slurm?

Background

There are some limited access clusters managed by Luddy IT Staff that are using the slurm job scheduler. These systems have a head node that you can log into and, from there, you use slurm commands to allocate and run jobs on the compute nodes. This page provides a very quick introduction to using slurm on these Luddy-managed clusters. Please see the SLURM Homepage for more detailed information about using SLURM and see the Storage System Notes below for cluster-specific storage information.

Head Nodes and Compute Nodes

The Luddy clusters have what is called a head node that you can just log into and compute nodes that you can use by allocating them via SLURM.

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Head Node</th>
<th>Compute Nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bio SGX Cluster</td>
<td>bio-sgx.soic.indiana.edu</td>
<td>bio-sgx01.soic.indiana.edu through bio-sgx12.cs.indiana.edu</td>
</tr>
<tr>
<td>Dagger Cluster</td>
<td>dagger.sice.indiana.edu</td>
<td>dagger01.sice.indiana.edu through dagger16.sice.indiana.edu</td>
</tr>
<tr>
<td>Hive Cluster</td>
<td>tank.sice.indiana.edu</td>
<td>hive.soic.indiana.edu</td>
</tr>
<tr>
<td>Tatooine Cluster</td>
<td>tatooine.sice.indiana.edu</td>
<td>tatooine1.sice.indiana.edu through tatooine8.sice.indiana.edu</td>
</tr>
</tbody>
</table>

From the head node, you can then run your jobs on the compute nodes. You should NOT do your compute processing on the head node. Rather, you will need to use SLURM from the head node to allocate compute nodes and run your jobs there.

Interactive Logins

In some cases, you will just want to allocate a compute node (or nodes) so you can ssh login and use the system interactively. Note that you are not allowed to just ssh login to a node without first allocating the resource. If you just want an interactive shell on one note you can easily do that as follows using srun:

```
srun -N 1 -n 1 --pty bash -i
```

You can also allocate a single node or multiple notes for ssh logins using the salloc command and then see which node(s) you were allocated using the squeue command. For example, you can ssh into the head node and allocate a node in the cluster as follows:

```
[sip] $ salloc -N 1 bash
salloc: Granted job allocation 109512
[sip] $ squeue
 JOBID PARTITION  NAME   USER ST       TIME  NODES NODELIST(REASON)
109512 batch bash robh R       0:12      1 odin006
[sip] $ ssh odin006
[odin006] ... run whatever you want here ...
[odin006] exit
Connection to odin006 closed.
[sip] $ exit
salloc: Relinquishing job allocation 109512
[sip] $
```

In this example (and those that follow) the command prompt is displayed as the host name in brackets followed by a dollar sign (eg. "[sip]$") to indicate which system you are logged into.

Be sure to exit the shell created by the salloc to relinquish your allocation, thereby making the nodes available to others. If you need to allocate multiple nodes for interactive ssh logins, you can just give the desired number of nodes using the -N argument to salloc.

There may be a limit on the time you can allocate a node and you will loose your allocation and be logged out of the nodes if you hit this limit.

Running Jobs Interactively

If you have a program that you just want to run interactively on a number of compute nodes, one way to do this is using the SLURM srun command. For example, let’s create a simple executable script called hostname.sh that just prints the hostname:
Then, we can run this script on 4 compute nodes as follows:

```bash
[odin]$ srun -N 4 hostname.sh
odin007.cs.indiana.edu
odin008.cs.indiana.edu
odin006.cs.indiana.edu
odin009.cs.indiana.edu
[odin]$
```

In this example you can see that we were allocated 4 different nodes and the output of running the test.sh script on each of them is displayed. This was run in parallel so the ordering of the output is indeterminate and may well vary each time you run this.

**Running Batch Jobs**

In many cases your job will have to run for a long time, you will have multiple jobs to run, and/or the resources needed to run your job will not be immediately available. In such cases, rather than using srun interactively and waiting around for the output you will want to use batch mode. This is specified using sbatch and, when your job completes, the output is then written to a file rather to the terminal. For example:

```bash
[odin]$ sbatch -N 4 hostname.sh
sbatch: Submitted batch job 109518
[odin]$ cat slurm-109518.out
odin006.cs.indiana.edu
[odin]$
```

At this point you are probably asking yourself why the output didn't show the hostname of 4 systems since we allocated 4 nodes? It is important to note that sbatch allocates 4 nodes but then only runs your script on the first node in the allocation (odin006 in the above example). Typically, your program will be taking care of managing the nodes that are allocated so sbatch doesn't run the same program on all 4 nodes.

Here is an example script called batchtest.sh that will run our simple hostname.sh script on all allocated nodes:

```bash
#!/bin/sh
srun hostname.sh
```

We can then run batchtest.sh via sbatch to run hostname.sh on all allocated nodes:

```bash
[odin]$ sbatch -N 4 batchtest.sh
sbatch: Submitted batch job 109519
[odin]$ cat slurm-109519.out
odin006.cs.indiana.edu
odin009.cs.indiana.edu
odin008.cs.indiana.edu
odin007.cs.indiana.edu
[odin]$
```

Our simple batchtest.sh script doesn't have to tell srun how many nodes to use. The SLURM system sets up environment variables defining which nodes we have allocated and srun then uses all allocated nodes.

**SLURM Commands**

The above examples provide a very simple introduction to SLURM. You should see the slurm man pages and on-line documentation for further information. The SLURM commands you are likely to be interested in include srun, sbatch, sinfo, squeue, scancel, and scontrol.

**Storage System Notes**

Each of the cluster are configured differently regarding data storage space. This table gives a summary of the storage space available on each cluster.
<table>
<thead>
<tr>
<th>Cluster</th>
<th>Filesystem</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bio SGX</td>
<td>$HOME</td>
<td>• Local on each node in 900GB root filesystem on single drive</td>
<td>No backups No disk redundancy</td>
</tr>
<tr>
<td>dagger</td>
<td>/HOME</td>
<td>• Local on each daggerNN compute node in 600GB software RAID1 filesystem</td>
<td>No backups</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Local on swarm in 500GB hardware RAID1 filesystem</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Local on sigma in 900GB single drive (no RAID)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Local on tesla in 500GB hardware RAID1 filesystem</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>No disk redundancy on sigma</td>
<td></td>
</tr>
<tr>
<td>dagger</td>
<td>/data</td>
<td>• Local on each daggerNN compute node in 1.8TB software RAID0 filesystem</td>
<td>No backups</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Local on swarm in 1.8TB hardware RAID0 filesystem</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• No /data filesystem on sigma or tesla</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>No disk redundancy</td>
<td></td>
</tr>
<tr>
<td>dagger</td>
<td>/shared</td>
<td>• NFS mounted 7TB filesystem hosted by dagger-nfs on software RAID5 filesystem</td>
<td>No backups</td>
</tr>
<tr>
<td>Hive</td>
<td>$HOME</td>
<td>• NFS mounted from central Luddy file servers</td>
<td>Nightly backups</td>
</tr>
<tr>
<td>Hive</td>
<td>/nix</td>
<td>• Local on hive in 800GB hardware RAID1 filesystem</td>
<td>No backups</td>
</tr>
<tr>
<td>Tatooine</td>
<td>$HOME</td>
<td>• NFS mounted 15TB filesystem hosted by tatooine on hardware RAID6 filesystem</td>
<td>Monthly backups</td>
</tr>
</tbody>
</table>