How do I use the RI Odin cluster?

Background

The Odin cluster was purchased through National Science Foundation (NSF) Research Infrastructure (RI) grant EIA-0202048 and is used for research purposes in the school. The system is listed as being in the "RI Cluster Domain" on this page:

What Linux systems are available?

The 128-node Odin cluster requires the use of the SLURM job scheduler in order to allocate resources and run jobs. This page provides a basic introduction to using SLURM. Please see the SLURM Homepage for more detailed information about using SLURM.

Head Nodes and Compute Nodes

The Odin cluster has 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>Odin Cluster</td>
<td>odin.cs.indiana.edu</td>
<td>odin001.cs.indiana.edu through odin128.cs.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

SSH logins to the Odin compute nodes (odin001 - odin128) are limited to coming only from IU hosts. If you need ssh login access from non-IU systems, you will have to request that your domain be added to the allow list.

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. You can allocate a single node for ssh logins using the `salloc` command and then see which node you were allocated using the `squeue` command. For example, you can ssh into the head node odin.cs.indiana.edu and allocate a node in the Odin cluster as follows:

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

In this example (and those that follow) the command prompt is displayed as the host name in brackets followed by a dollar sign (eg. "[odin]$") 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 modes 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 is a 4 day limit on the time you can allocate a node and you will lose 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:

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

Then, we can run this script on 4 compute nodes as follows:

```
[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 than to the terminal. For example:

```
[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:
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.