

A Brief Introduction to managing data science workloads on HPC

Author: Matt Dwyer

Date: 29 April 2021

Agenda

- Housekeeping on your host machine
- Useful slurm commands (getting to know the system)
- Requesting an interactive job
- Containers, port-forwarding, interactive example (jupyter)
- Containers, port-forwarding, batch example (jupyter)
- Discussion

Preparing your host machine

- Modify your `~/.ssh/config`
- Copy your public ssh key to caviness

```
>> ssh-copy-id -i ~/.ssh/id_rsa.pub caviness
```

- `ssh caviness`

```
>> cat ~/.ssh/config
```

```
Host caviness
```

```
    Hostname caviness.hpc.udel.edu
```

```
    ForwardX11 yes
```

```
    User <your_udel_username>
```

Useful Slurm Commands: sview

Shows partitions, configurations, and status

Default time, Max Time

Max Nodes, Total Nodes

Total CPUs

Node Status: Idle Nodes, allocated, drained, and down

and much more

| Partition | Default | Part. State | Time Limit | Node Count | Node State | NodeList |
|-----------------|---------|-------------|-------------|------------|------------|--|
| afwallace | no | up | 7-00:00:00 | 56 | | r00n[01-17,45-55],r01n[01-17,45-55] |
| akanane | no | up | 7-00:00:00 | 83 | | r03n[29-57],r03n[24-26],r04n[00-23,50-76] |
| arclet | no | up | 7-00:00:00 | 6 | | r00n[18-20],r01n[18-20] |
| bagozzib | no | up | 7-00:00:00 | 25 | | r03n[00-23,28] |
| biophysics | no | up | 7-00:00:00 | 5 | allocated | r00g[02,04],r01g[02-04] |
| cbbi | no | up | 7-00:00:00 | 56 | | r00n[01-17,45-55],r01n[01-17,45-55] |
| ccm_gillespi | no | up | 7-00:00:00 | 162 | | r00n[01-17,45-55],r01n[01-17,45-55],r00n[21-44],r01n[21-44],r03n[29-57],r03g05,r03n[00-23,28],r03g[00-02] |
| cebg | no | up | 7-00:00:00 | 105 | | r00n[21-44],r01n[21-44],r00n[18-20],r01n[18-20],r03n[00-23,28],r04n[24-29,41],r04n[40,42-43,45-46,48],r04n[30-39,44,47,49] |
| cniel | no | up | 7-00:00:00 | 51 | | r04n[00-23,50-76] |
| dsai | no | up | 7-00:00:00 | 56 | | r00n[01-17,45-55],r01n[01-17,45-55] |
| dsasters | no | up | 7-00:00:00 | 129 | | r00n[21-44],r01n[21-44],r00n[01-17,45-55],r01n[01-17,45-55],r03n[00-23,28] |
| dsasters | no | up | 7-00:00:00 | 2 | | r03g[03-04] |
| dsasters | no | up | 7-00:00:00 | 136 | | r00n[01-17,45-55],r01n[01-17,45-55],r03n[29-57],r04n[00-23,50-76] |
| dsasters | no | up | 7-00:00:00 | 5 | idle | r00n[00-00],r00n[00,56],r01n[00,56] |
| dsasters | no | up | 7-00:00:00 | 56 | | r00n[01-17,45-55],r01n[01-17,45-55] |
| dsasters | no | up | 7-00:00:00 | 2 | | r03g[07-08] |
| ececis_research | no | up | 7-00:00:00 | 75 | | r00n[21-44],r01n[21-44],r00g03,r03g06,r03n[00-23,28] |
| ececis_research | no | up | 7-00:00:00 | 38 | mixed | r00n[21-23,26-27,29,31-33,36-38],r01n[24-26,28-29,31-34,36,38-40,42-44],r03n[00-01,04,11-13,16,19,21,28] |
| ececis_research | no | up | 7-00:00:00 | 29 | allocated | r00g03,r00n[24,30,34-35,40-44],r01n[21-23,27,37,41],r03g06,r03n[02-03,05-10,14,18,20,23] |
| ececis_research | no | up | 7-00:00:00 | 5 | idle | r00n[25,39],r03n[15,17,22] |
| ececis_research | no | up | 7-00:00:00 | 1 | drained | r01n30 |
| ececis_research | no | up | 7-00:00:00 | 2 | down* | r00n28,r01n35 |
| ecosys | no | up | 7-00:00:00 | 53 | | r04n[00-23,50-76],r04s[00-01] |
| elliottlab | no | up | 7-00:00:00 | 56 | | r00n[01-17,45-55],r01n[01-17,45-55] |
| geosci | no | up | 7-00:00:00 | 48 | | r00n[21-44],r01n[21-44] |
| gleghorn | no | up | 7-00:00:00 | 56 | | r00n[01-17,45-55],r01n[01-17,45-55] |
| icecube | no | up | 7-00:00:00 | 29 | | r03n[29-57] |
| it_css | no | up | 7-00:00:00 | 266 | | r00g[01-04],r00n[01-55],r01g[00-04],r01n[01-55],r02s00,r03n[00-57],r03g[00-08],r04n[00-76],r04s[00-01] |
| it_nss | no | up | 7-00:00:00 | 266 | | r00g[01-04],r00n[01-55],r01g[00-04],r01n[01-55],r02s00,r03n[00-57],r03g[00-08],r04n[00-76],r04s[00-01] |
| jayaraman_lab | no | up | 7-00:00:00 | 107 | | r00n[01-17,45-55],r01n[01-17,45-55],r04n[00-23,50-76] |
| jneun | no | up | 7-00:00:00 | 7 | | r04n[24-29,41] |
| kirby | no | up | 7-00:00:00 | 83 | | r00g01,r01g[00-01],r03n[29-57],r04n[00-23,50-76] |
| kuehl_group | no | up | 7-00:00:00 | 99 | | r00n[21-44],r01n[21-44],r04n[00-23,50-76] |
| kukuiKa_lab | no | up | 7-00:00:00 | 56 | | r00n[01-17,45-55],r01n[01-17,45-55] |
| lg-swap | no | up | 30-00:00:00 | 1 | idle | r02s01 |
| lianqlab | no | up | 7-00:00:00 | 53 | | r04n[00-23,50-76],r04s[00-01] |

Interactive Job: salloc

```
salloc -t hh:mm:ss -N <num_nodes> \  
-p <partition> -A <your_workgroup> \  
-J <jobname>
```

- -N, -p, -A are required
- There are plenty other features that may be useful to your workloads.. Run “man salloc” for a full list

* Wrap the command into a script with some logic for quick and easy interactive job allocations

```
./interactiveSession <time> <num_nodes>  
<partition> <workgroup>
```

```
>> cat interactiveSession  
#!/bin/bash  
  
wall_time=30  
num_nodes=1  
partition="ececis_research"  
account="ececis_research"  
  
if [ -z ${1+x} ]; then  
    echo "Using default value for walltime"  
else  
    wall_time=$1  
fi  
  
if [ -z ${2+x} ]; then  
    echo "Using default value for number of nodes"  
else  
    num_nodes=$2  
fi  
  
if [ -z ${3+x} ]; then  
    echo "Using default value for partition"  
else  
    partition=$3  
fi  
  
if [ -z ${4+x} ]; then  
    echo "Using default value for account"  
else  
    account=$4  
fi  
  
echo "Time: ${wall_time} minutes"  
echo "Number of nodes: ${num_nodes}"  
echo "Partition: ${partition}"  
echo "Account: ${account}"  
  
echo "Are these values correct?"  
read -r -p "Are you sure? [y/N] " response  
response=${response,,} # tolower  
if [[ "$response" =~ ^([yes|y])$ ]]  
then  
    cmd="salloc --time 00:${wall_time}:00 -N ${num_nodes} -p ${partition} -A  
${account}"  
    echo "Running command: ${cmd}"  
    ${cmd}  
else  
    echo "Exiting on user input"  
fi
```

Containers, Port-forwarding, Front-end Applications

Interactive Job: jupyter notebook example

- Connect to Caviness

```
>> ssh caviness
```

- Load the singularity package

```
>> vpkg_devrequire singularity
```

- Convert the jupyter docker image to singularity with singularity pull

```
>> singularity pull docker://jupyter/datascience-notebook:python-3.8.8
```

- Start up an interactive session

```
>> ./interactiveSession <time (min)> <num_nodes> <partition> <account>
```

- Launch container

```
>> singularity exec datascience-notebook_python-3.8.8.sif jupyter notebook
```

- In another shell, port forward to your running notebook

```
>> ssh -L 8888:localhost:8888 caviness
```

```
>> ssh -L 8888:localhost:8888 <compute-node>
```

Containers, Port-forwarding, Front-end Applications

Batch Job Jupyter

- Connect to Caviness

```
>> ssh caviness
```

- Create batch script
- Run batch script

```
>> sbatch batch_jupyter
```

- Find the node the job is running on using squeue

```
>> squeue -u $USER
```

- In another shell, port forward to your running notebook

```
>> ssh -L 8888:localhost:8888 caviness  
>> ssh -L 8888:localhost:8888 <compute-node>
```

```
>> cat batch_jupyter
```

```
#SBATCH -A "ececis_research"
```

```
#SBATCH -p "ececis_research"
```

```
#SBATCH -J "jupyter"
```

```
#SBATCH -N 1
```

```
#SBATCH -t 00:10:00
```

```
vpkg_devrequire singularity
```

```
singularity exec datascience-notebook_python-3.8.8.sif  
jupyter notebook
```

* On caviness, /opt/templates/slurm/ has batch script templates..
Use them, they're very detailed and they work

Useful Slurm Commands

Job Manipulation and Cluster configuration

| | |
|---|---|
| Cancel a job | <code>scancel <jobid></code> |
| Cancel all of your jobs | <code>scancel -u \$USER</code> |
| Get status of a job | <code>squeue <jobid></code> |
| Status of all of your jobs | <code>squeue -u \$USER</code> |
| Job Status in more detail | <code>scontrol show jobid -dd <job_id></code> |
| Slurm accounting info for historical jobs | <code>sacct -j <jobid></code> <code>--format=JobID,JobName,MaxRSS,Elapsed</code> |
| Information on partitions | <code>scontrol show partition</code> |
| Condensed view of node features | <code>sinfo -o "%20N %10c %10m %25f %10G "</code> |

Questions

- The Caviness system is very well documented. Refer to <http://docs.hpc.udel.edu/> for more information on HPC tasks