Where can I find Copperhead user notes?

Tell Me

1. The Copperhead Cluster can be accessed via SSH to "hpc.uncc.edu." This will connect the user to one of the Interactive/Submit nodes located in the Copperhead Cluster pool. From those nodes, a user can submit jobs requesting the following resources:
2. General Compute Nodes (63 x 16 cores ea + 2 x 32 cores ea = 1072 CPU cores)
3. GPU Compute Nodes (4 x 2 NVIDIA K80 GPUs and 16 cores ea = 8 GPUs and 64 CPU cores)
4. Large Memory Compute Nodes (16 CPU cores and 768GB RAM)
5. Jobs should always be submitted to the "copperhead" queue unless directed otherwise by URC Support.

Copperhead Defaults

To make more efficient use of the resources, user jobs are now submitted with a set of default resource requests which can be overridden on the qsub command line or in the job submit script via qsub directives. If not specified by the user, the following defaults are set:

```
#PBS -l walltime=8:00:00  # Max Job Run time is 8 hours
#PBS -l mem=2GB           # Allow up to 2GB of Memory per job
```

See the discussions below for more details

**Nodes= vs Procs=**

In the older URC clusters i.e. viper, cobra, python, etc. if a job needed a particular number of processors (procs) and did not care how they were distributed among the compute nodes, the following syntax was allowed:

```
#PBS -l nodes=16
```

This would reserve the first 16 procs available regardless of where they were located. On Copperhead, this would actually cause the job to attempt to reserve 16 nodes with 1 proc each which may not be desirable. To make more efficient use of resources and for clarification of the request, this syntax is no longer valid. Instead use the following:

If you really want X procs on Y nodes:

```
#PBS -l nodes=Y:ppn=X
```

If you just want X procs and don’t care which or how many nodes they are on, use:

```
#PBS -l procs=X
```

This will allow the scheduler to better utilize available resources and may help your job get scheduled quicker.

**Walltime**

This determines the actual amount of time a job will be allowed to run. If not specified by the user, the default value is now 8 hours. This value (or less) automatically labels a job as being a “short” job and therefore will have more potential nodes on which it can run. Jobs requiring longer than 8 hours are considered "long" jobs and are restricted as to potential nodes they can run on. In most cases, the longer the requested walltime, the lower the priority a job will have in competition with other, shorter, jobs.

Example: #PBS -l walltime=8:00:00  # 8 hours, 0 minutes, 0 seconds

**Memory (mem)**

mem – Is the amount of memory to be allocated to a job. If not set by the user, it defaults to roughly 2GB per core requested. “mem” applies to the entire job and is therefore a separate resource request which can be specified either as part of the nodes specification (separated by a comma)

Example: #PBS -l nodes=2:ppn=2,mem=8GB

or as a separate directive

Example: #PBS -l nodes=2:ppn=2

```
#PBS -l mem=8GB
```

These two examples are equivalent and request a total of 8GBs of memory spread across 4 cores on 2 nodes.

Jobs whose memory requirements will not allow it to fit on a regular compute node will automatically be scheduled on a large memory compute node. Regular compute nodes have 128 GBs of memory and 16 cores.

Jobs that exceed the requested memory will be terminated by the scheduler.

**GPUs**

GPUs are requested like procs. Currently there are 4 Copperhead nodes each of which contains 2 addressable GPUs. Note that gpus= is **part of the node’s properties**, therefore you use a colon to separate it (like :ppn=), as opposed to separate resource request (like ,mem=), which uses a comma to separate.
Example:  
```bash
#PBS -l nodes=1:ppn=1:gpus=1
```

# (1 node with 1 cpu and 1 gpu)

Note that you **cannot** use the following on Copperhead:

```bash
#PBS -l gpus=N
```

While Torque will accept this by defaulting to `nodes=1`, this becomes confusing if you attempt to ask for `gpus=3` on Copperhead since there is a maximum of 2 gpus per node. If you need more than 2 gpus, for a job, the following are examples of valid requests:

```bash
#PBS -l nodes=3:ppn=1:gpus=1    # 3 Nodes, 1 gpu each (3 GPUs)
#PBS -l nodes=2:ppn=1:gpus=2    # 2 Nodes, 2 gpus each (4 GPUs)
```

---

**Related FAQs**

- Where can I find Copperhead user notes?
- How do I configure GPU compute jobs for the cluster?
- Can I run parallel jobs on the cluster?
- Can I log in to Research Computing clusters from home?
- HPC Cluster Questions