**Duke University**

Duke Computing Cluster

1. Login

ssh ed155@dcc-slogin.oit.duke.edu

All login nodes (01,02,03) files are shared

ssh ed155@research-tarokhlab-01.oit.duke.edu

1. File Transfer

scp data.txt netid@dcc-slogin-01.oit.duke.edu:.

scp netid@dcc-slogin-01.oit.duke.edu:data.txt .

• Use either

scp -r (small files) or rsync –av (large files)

• Pushing a directory:

rsync –av dir1/ netid@dcc-slogin-01.oit.duke.edu:. or

scp -r dir1/ netid@dcc-slogin-01.oit.duke.edu:.

• Pulling a directory:

rsync –av netid@dcc-slogin-01.oit.duke.edu:~/dir1 .

scp -r netid@dcc-slogin-01.oit.duke.edu:~/dir1 .

1. SLURM
   1. Run on partition

•  Most DCC partitions are dept-owned machines   
•  These can only be used by members of the group   
•  Submitting to a group partition gives “high-priority”   
•  Submit to partitions with “--partition=” or “-p“, e.g.   
**#SBATCH –p (partition name)** (in a script) or   
**srun –p (partition name)--pty bash –i**

**srun --pty bash -i**(interactively)  
•  The default DCC partition is called “common”   
•  The common partition gives “low-priority” to most ESX hosts

**COMMANDS**

•  sbatch   
Submit a batch job (like “qsub”)   
•  #SBATCH   
Specify job parameters (like “#$”)   
•  squeue (like “qstat”)   
Show lists of jobs   
•  scancel (like “qdel”)   
Delete one or more batch jobs   
•  sinfo (like “qhost”)   
Show info about machines   
•  scontrol   
Show cluster configuration information

1. DCC Partitions   
   There are different DCC partitions to which batch   
   jobs and interactive sessions can be directed:   
   •  **common**, for jobs that will run on the DCC core   
   nodes (up to 64 GB RAM).   
   •  **common-large**, for jobs that will run on the DCC   
   core nodes (64-240 GB GB RAM).   
   •  **gpu-common**, for jobs that will run on DCC GPU   
   nodes.   
   •  **Group partitions** (partition names varies), for   
   jobs that will run on lab-owned nodes
2. GPU nodes

To run a GPU batch job, add the job script lines   
**#SBATCH** -p gpu-common --gres=gpu:1   
**#SBATCH** -c 6   
To get an interactive GPU node session, type the command line   
**srun -p gpu-common --gres=gpu:1 -c 6 --pty bash –i**

tm103@dscr-slogin-02 ~ $ **srun -p gpu-common --gres=gpu:1 -c 6 --pty bash -i**tm103@**dscr-gpu-01** ~ $ /usr/local/cuda-7.5/samples/1\_Utilities/deviceQuery/  
deviceQuery  
...  
Detected 1 CUDA Capable device(s)  
Device 0: "Tesla K80"  
CUDA Driver Version / Runtime Version 7.5 / 7.5  
CUDA Capability Major/Minor version number: 3.7  
Total amount of global memory: 11520 MBytes (12079136768 bytes)  
(13) Multiprocessors, (192) CUDA Cores/MP: 2496 CUDA Cores

We have two generally available GPU partitions for use in the DCC:

**gpu-common** for jobs that will run on DCC GPU nodes

**scavenger-gpu** for GPU jobs that will run on lab-owned nodes in “low priority” (kill and requeue preemption).

Researchers that have purchased GPU nodes have a high priority partition for their GPUs that will preempt any jobs submitted to scavenger-gpu and are running on their nodes.

Examples of batch submission are here: <https://dcc.duke.edu/dcc/slurm/?h=#gpu-jobs> and there’s also OnDemand for interactive Jupyterlab and RStudio jobs: <https://dcc.duke.edu/OpenOnDemand/> via the OnDemand server [https://dcc-ondemand-01.oit.duke.edu](https://dcc-ondemand-01.oit.duke.edu/).