



---

# First Introduction to Using SLURM on Discover

Chongxun (Doris) Pan  
doris.pan@nasa.gov  
September 24, 2013



# SLURM ON!



- ◆ **Why Simple Linux Utility for Resource Management?**
  - ◆ An open-source, fault-tolerant, and highly scalable cluster workload scheduler
  
- ◆ **What to expect after we migrate from PBS to SLURM next month?**
  - ◆ Nearly Transparent. Most of your PBS scripts should run on SLURM without any changes.
  - ◆ A few things to bear in mind...
  
- ◆ **What if you cannot wait to go native SLURM?**
  - ◆ Kudos for the forward-thinking attitude!
  - ◆ Will show you how easy to convert to native SLURM





# What to Expect



- ◆ NCCS is working HARD to develop wrappers to make transparent conversions. But each workload scheduler has its unique features...
- ◆ Most of the PBS commands, including qsub, qstat, qdel, qalter, should function the same way as in PBS, along with their typical options
  - ◆ “qalter -o” is not supported with the current release. With SLURM, as soon as a job starts the output is written directly to the file that you specified in the “-o” directive. No temporary pbs\_spool files!
  - ◆ You may notice slight formatting difference for “qstat” output.
  - ◆ Interactive batch using “qsub -l” works the same way, so does “xsub -l”.



## What to Expect (*Cont'd*)



- ◆ For batch jobs, most of the commonly used PBS environment variables are defined, including PBS\_NODEFILE, PBS\_JOBID, PBS\_JOBDIR, and PBS\_O\_WORKDIR
- ◆ The PBS job scripts will run almost seamlessly in SLURM. Most of the PBS attributes work the same way, including
  - ◆ #PBS -l select=*i*:ncpus=*j*:mpiprocs=*k*:proc=west/sand
  - ◆ #PBS -l walltime=*hh:mm:ss*
  - ◆ #PBS -W group\_list, -W depend, -W umask
  - ◆ #PBS -o, -e
    - ◆ #PBS -j is ignored. By default, SLURM writes stdout and stderr to one file, as specified by #PBS -o.
    - ◆ To separate stdout and stderr, specify both #PBS -o and -e. If only #PBS -e is specified, the stdout will be written into a file named, slurm- $\$$ SLURM\_JOBID.out.
    - ◆ Without -e and -o, both the stdout and stderr will be written into slurm- $\$$ SLURM\_JOBID.out.



# Intel MPI



- ◆ For Intel MPI, the following Intel MPI modules are available and “mpirun” command works the same way:
  - mpi/impi-3.2.2.006
  - mpi/impi-4.0.3.008
  - mpi/impi-4.1.0.024
  - mpi/impi-4.1.1.036
- ◆ We intend to retire some older and all the beta versions of Intel MPI
  - We highly recommend switching to Intel MPI 4 modules, specially 4.0.3.008 and 4.1.0.024
- ◆ Please contact NCCS User Support for assistance in migrating to a newer version of MPI



# MVAPICH2 and OpenMPI



- For MVAPICH2 and OpenMPI, “mpirun” should work seamlessly as well in the SLUM environment
- We intend to hide some older versions, including:
  - other/mpi/mvapich2-1.6rc2/intel-11.1.072
  - other/mpi/mvapich2-1.7\*
  - other/mpi/mvapich2-1.8/\*
  - other/mpi/mvapich2-1.8a2/\*
  - mpi/openmpi-1.2.5/intel-9
  - mpi/openmpi-1.2.5/intel-10
  - other/mpi/openmpi/1.4.\*
  - other/mpi/openmpi/1.6-gcc\_4.8-20120401\_nag-5.3-854
  - other/mpi/openmpi/1.6.0\*
  - other/mpi/openmpi/1.6.3\*
  - other/mpi/openmpi/1.6.4\*



# Native SLURM Basics



- ◆ Six basic user commands should cover most of your needs
  - ◆ sbatch -- Submit job script (Batch mode)
  - ◆ salloc -- Create job allocation and start a shell (Interactive Batch mode)
  - ◆ srun – Run a command within a batch allocation that was created by sbatch or salloc
  - ◆ scancel -- Delete jobs from the queue
  - ◆ squeue -- View the status of jobs
  - ◆ sinfo – View information on nodes and queues
- ◆ --help option prints brief description of all options
- ◆ --usage option prints a list of the options
- ◆ “srun” or mpirun/mpirun/hydra?
  - ◆ srun is the best integrated with SLURM and supports process tracking, accounting, task affinity, suspend/resume and other features.



# SLURM commands vs. PBS commands



<b>sbatch myscript.j</b>	qsub myscript.j
<b>salloc -N 2 -n 16 --ntasks-per-node=8 -t 1:00:00 -p general --account=k3001</b>	qsub -I -l select=2:mpiprocs=8,walltime=1:00:00 -q general -W group_list=k3001
<b>squeue</b>	qstat
<b>squeue -j 4638171</b>	Qstat -f 4638171
<b>sinfo</b>	qstat -Q
<b>Squeue -u cpan2</b>	Qstat -u cpan2
<b>scancel 4638171</b>	qdel 4638171

For sbatch or salloc commands, almost all options have two formats:

- A single letter option (e.g. “-p debug”, “-N 2”, “-n 16”. Note single dash!), or
- A verbose option (e.g. “--**partition**=debug”, “--nodes=2”, “--ntasks=16”. Note double dash!)

Acceptable walltime “-t or --time=”formats include “minutes”, “minutes:seconds”, “hours:minutes:seconds”.



# SLURM env. variables vs. PBS env. variables



<b>SLURM_JOBID</b>	PBS_JOBID
<b>SLURM_NODELIST</b>	PBS_NODEFILE
<b>SLURM_SUBMIT_HOST</b>	PBS_O_HOST
<b>SLURM_SUBMIT_DIR</b>	PBS_O_WORKDIR
<b>SLURM_NNODES</b> (number of nodes requested)	N/A
<b>SLURM_NTASKS</b> (number of MPI procs requested)	N/A
<b>SLURM_CPUS_ON_NODE</b> (number of CPUS on the allocated node)	N/A
<b>SLURM_NTASKS_PER_NODE</b> (return “mpiprocs” number)	N/A



# A native SLURM job script vs. A PBS job script



```
#!/bin/csh

#SBATCH -J MyFirstSLURM
#SBATCH -A k3001
#SBATCH -N 2 -n 16 --ntasks-per-node=8
#SBATCH -t 1:00:00
#SBATCH --mail-user=doris.pan@nasa.gov
#SBATCH -p general
#SBATCH -o testmpi.out
##SBATCH -e testmpi.err
#SBATCH -d afterany:11697
```

```
module purge
module load comp/intel-13.1.3.192
module load mpi/impi-4.1.0.024
```

```
cd $NOBACKUP
mpif90 testmpi.f90
mpiexec.hydra -np 16 ./a.out
```

```
#!/bin/csh

#PBS -N MyFirstPBS
#PBS -W group_list=k3001
#PBS -l select=2:mpiprocs=8
#PBS -l walltime=1:10:00
#PBS -M doris.pan@nasa.gov
#PBS -q general
#PBS -o testmpi.output
#PBS -j oe
#PBS -W depend=afterany:11697
```

```
module purge
module load comp/intel-13.1.3.192
module load mpi/impi-4.1.0.024
```

```
cd $NOBACKUP
mpif90 testmpi.f90
mpiexec.hydra -np 16 ./a.out
```



To Learn More About SLURM...

[slurm.schedmd.com/](http://slurm.schedmd.com/)

<http://www.nccs.nasa.gov/primer/slurm/slurmPBS.html>