Getting started with the new Oscar cluster

Why a new Oscar?

BROWN UNIVERSITY CENTER FOR COMPUTATION & VISUALIZATION

Improvements:

-Upgrading the Linux operating system to CentOS 6.3

-Replacing the existing scheduling and batch system (Torque/Moab) with a new system called SLURM

-Switching to a node-packing policy where the fundamental unit is a "core" instead of an entire "node" to better support serial and high-throughput jobs

-Adding new 16-core Intel "Sandy Bridge" nodes

-Updating and optimizing many software modules

New module system

BROWN UNIVERSITY CENTER FOR COMPUTATION & VISUALIZATION

Basic commands:

```
module list
module swap <module name>/<version>
module load <module name>/[<version>]
module avail -v <module name>
module unload <module name>
```

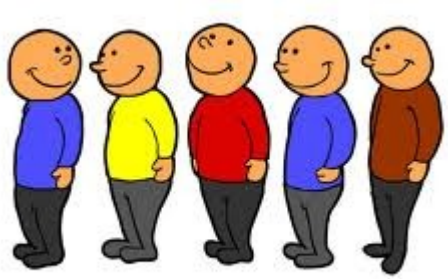
New default modules:
`intel, centos-updates, centos-libs` – these are needed for your Oscar to work. Please don't unload them.

-New module system won't work under csh, tcsh, etc. At this time we ONLY support using bash on the new cluster. (You can change your shell with **ypchsh**.)
-If you absolutely, positively must use a shell other than bash for your work, please e-mail support@ccv.brown.edu to discuss this with us.

Queues (or partitions)

BROWN UNIVERSITY CENTER FOR COMPUTATION & VISUALIZATION

Queues on the old system are now known as partitions.



default-batch – suitable for all jobs
small-batch – 60 minute limit
tiny-batch – 5 minute limit
debug – 30 minute limit, max 4 nodes

timeshare – smp nodes
gpu – gpu-equipped nodes
sandy-batch-mpi and default-batch-mpi – for running multi-node mpi jobs

Machines can now be shared between partitions (example: node401 belongs to tiny-batch, small-batch, and default-batch partitions)

Running your job

BROWN UNIVERSITY CENTER FOR COMPUTATION & VISUALIZATION

-Your environment settings and working directory from the login node are now propagated to the compute node.

-Output of job is now stored “on the fly”, in a file called slurm-<jobid>.out.

-Jobs now run in a container similar to a VM. Each node can support multiple job containers, even from different users.

-Memory is now a consumable resource, like CPUs. You can tell your job to wait in the queue until you can get a node with a certain amount of memory.

-Segmentation fault error? Make sure you are requesting enough memory for your job.

Scheduler commands

BROWN UNIVERSITY CENTER FOR COMPUTATION & VISUALIZATION

Some commands still work on new cluster:

```
myq   allq    interact
```

Deprecated commands:

	REPLACED WITH...
`qsub`	`sbatch`
`qdel`	`scancel`
`qstat`	`squeue`
`showq`	`squeue`

New commands:

`sacct:` show list of your recent jobs

`nodes:` view list of all compute nodes by partition

Batch scripts

BROWN UNIVERSITY CENTER FOR COMPUTATION & VISUALIZATION

TORQUE/MOAB:

```
#!/bin/bash
#PBS -N MATLAB
#PBS -t 0-15,25
#PBS -l nodes=1:ppn=1
#PBS -l walltime=1:00:00
#PBS -j oe

cd $PBS_O_WORKDIR

# This script runs 16 independent MATLAB tasks across 2 Oscar nodes by creating
# a "job array" of 16 separate 1-core jobs.

echo "Starting job $PBS_ARRAYID on $HOSTNAME"
matlab -nodisplay -nojvm -r "my_func($PBS_ARRAYID); quit;"

# Alternatively, instead of using the integers {1..16} as the input to your
# MATLAB function, you can use lines 1 through 16 of a text file, say
# 'inputs.txt', that is in the same directory as where you submit this script.

ID=$(awk "NR==$PBS_ARRAYID" inputs.txt)
echo "Starting job '$ID' on $HOSTNAME"
matlab -nodisplay -nojvm -r "my_func($ID); quit;"
```

SLURM:

```
#!/bin/bash
#SBATCH -J MATLAB
#SBATCH --ntasks=1
#SBATCH --ntasks-per-node=1
#SBATCH --time=1:00:00
#SARRAY --range=1-16,25

cd $PWD

# This script runs 16 independent MATLAB tasks across 2 Oscar nodes by creating
# a "job array" of 16 separate 1-core jobs.

echo "Starting job $SLURM_ARRAYID on $HOSTNAME"
matlab -nodisplay -nojvm -r "my_func($SLURM_ARRAYID); quit;"

# Alternatively, instead of using the integers {1..16} as the input to your
# MATLAB function, you can use lines 1 through 16 of a text file, say

# 'inputs.txt', that is in  the same directory as where you submit this script.

ID=$(awk "NR==$SLURM_ARRAYID" inputs.txt)
echo "Starting job '$ID' on $HOSTNAME"
matlab -nodisplay -nojvm -r "my_func($ID); quit;"
```

Scheduler flags

BROWN UNIVERSITY CENTER FOR COMPUTATION & VISUALIZATION

-n # : request # cores
-t HH:MM:SS : request walltime of HH:MM:SS
--mem=xxx : request total memory for the jobof xxx (example: 4 GB)
-p xxx : request a specific partition
-J xxx : specify the job name that will be displayed when listing the job

`man sbatch` for full list of options

Flags can be placed in your batch script with the #SBATCH directive:

`jobscript.sh:`

```
#!/bin/bash
#SBATCH -n 4
#SBATCH -t 1:00:00
...
```

or passed as arguments to `sbatch` at submission time.

```
$ sbatch -n 4 -t 1:00:00 --mem=16G jobscript.sh
```

Scheduler flags

BROWN UNIVERSITY CENTER FOR COMPUTATION & VISUALIZATION

-n # : request # cores
-t HH:MM:SS : request walltime of HH:MM:SS
--mem=xxx : request total memory for the jobof xxx (example: 4 GB)
-p xxx : request a specific partition
-J xxx : specify the job name that will be displayed when listing the job

`man sbatch` for full list of options

Flags can be placed in your batch script with the #SBATCH directive:

`jobscript.sh:`

```
#!/bin/bash
#SBATCH -n 4
#SBATCH -t 1:00:00
...
```

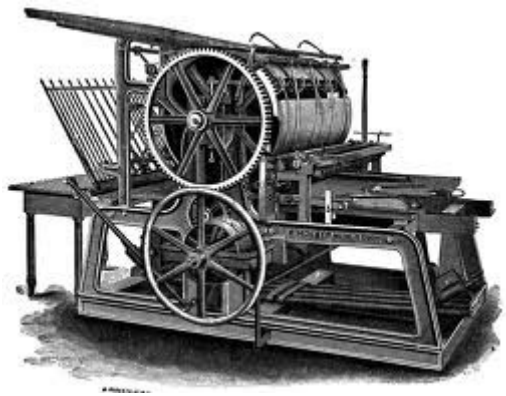
or passed as arguments to `sbatch` at submission time.

```
$ sbatch -n 4 -t 1:00:00 --mem=16G jobscript.sh
```

Converting old batch scripts

BROWN UNIVERSITY CENTER FOR COMPUTATION & VISUALIZATION

```
convert_batch_script scriptname
```



-Will attempt to convert your Torque/PBS script into a file called scriptname_SLURM that you can use as a starting point for running your script on SLURM.
-Read the output of this command to see if any commands were not converted.
-For more information on how to convert your batch script consult

https://computing.llnl.gov/linux/slurm/sbatch.html, or `man sbatch`

E-mail support@ccv.brown.edu if you are stuck.

Job arrays

Place the line `#SARRAY --range=<range-spec>` in your script. Range-spec is a comma separated list of integers (starting with 1) or ranges, separated by a dash.

```
#SARRAY --range=1-15
#SARRAY --range=1-10,12,14,16-20
```

Values in the range can be accessed in your script from the variable `$SLURM_ARRAYID` (old name: $PBS_ARRAYID)

Pass all your other flags in as #SBATCH commands, and submit your job with `sarray <jobscript>.`

This will create multiple jobs with the same name but different job ids. You can delete the jobs with

```
groupcancel JOBNAME
```

This will delete all the jobs with a name matching the jobname. (You can use this command to delete multiple jobs even if they are not part of the same job array).

MPI jobs

BROWN UNIVERSITY CENTER FOR COMPUTATION & VISUALIZATION

```
mpirun -n # jobscript.sh
```

We are supporting mvapich2 (OSU mvapich library) as the primary MPI library on the cluster. Software on the cluster has been built with mvapich2 almost exclusively.

When compiling your own MPI software, use mvapich2. OpenMPI should only be used in cases where mvapich2 does not work!

For multiple node MPI jobs, please request one of the mpi queues (ex. default-batch-mpi, sandy-batch-mpi). This will ensure the architecture is the same on all of the machines running your job.

Potential issues

BROWN UNIVERSITY CENTER FOR COMPUTATION & VISUALIZATION

Module warning message

```
module: warning: unknown module '----'

  Please contact CCV support (support@ccv.brown.edu) if you think that software
with this module name should be installed.
```

-First, check to see if module is already installed (module avail -v xxxx)
-If you need software and it is still missing, e-mail support@ccv.brown.edu.

Module command is missing

```
module: Command not found
```

-Check if you are using a shell other than bash.

Where to get help

BROWN UNIVERSITY CENTER FOR COMPUTATION & VISUALIZATION

Example batch scripts:
~/batch_scripts directory (slurm subdirectory)

SLURM Website:
https://computing.llnl.gov/linux/slurm

Updated Oscar documentation:
http://ccv.brown.edu/doc

HAVE FUN!