COLLEGE OF ENGINEERING
UNIVERSITY OF MICHIGAN

# Flux to Great Lakes Training

Research Computing Consultants
Malcolm Miranda mmiranda@umich.edu
Jason Sonk jsonk@umich.edu

https://www-personal.umich.edu/~mmiranda/GLSlides.pdf

190822

# Outline

What we will cover today:

- High level description of Great Lakes, Lighthouse, & Armis2
- New LMOD software layout
- PBS to Slurm script conversion
- Discuss on demand usage, CPU & RAM
- Hands on examples using Beta:
    - Software modules
    - Finding resources in Slurm
    - Simple Slurm example
    - MPI Slurm example

# Clusters

New cluster and changes to existing clusters:

- Great Lakes
  - Slurm and new software layout
  - Standard nodes: Skylake 36 Cores 3.0Ghz, 192GB
  - Large Memory nodes: 1.5 TB
  - GPU nodes: 2 x Tesla V100 16 GB
  - Maximum wall clock time: Two weeks
- Armis2
  - Slurm and new software layout
  - Standard nodes: Haswell
  - ARC-TS will start charging for the service at some point

# Lighthouse

- Flux Operating Environment nodes will migrate to Lighthouse
  - Eligible nodes only!
  - Actual compute hardware does not change
- New LMOD software layout
- Will have its own login node: lighthouse.arc-ts.umich.edu
- Beta home directories become Lighthouse home directories
- /scratch
  - Turbo NFS volume, not Luster
  - Files are auto purged!

# Dates:

- August 14, 2019
  - Great Lakes general availability
- November 25, 2019
  - Lighthouse migration must be completed
  - Flux goes offline
  - Billing for Great Lakes starts

# Software Module Layout

- The current Flux software layout is flat
  - Everything shows up regardless of whether you can use it or need it!
- New LMOD software layout
  - Tree structure
  - Only software that is truly available to you is visible
  - Cleaner module listing
  - Software collections:
    - Bioinformatics
    - OnCampusAccessOnly
    - RestrictedLicense

# /scratch

- Is for active data only.
  - Not for storage
- Date is automatically after 60 days!
  - Files are auto purged!
- Each cluster has its own /scratch system.
  - Armis2 and Lighthouse: Turbo NFS volume, not Luster
  - Great Lakes: GPFS

# Slurm

Slurm is a scalable cluster management and job scheduling system for Linux clusters.

- Replaces PBS/Torque
- Faster
  - Can start many jobs quickly
  - Can predict job startup time
- Scales better, more jobs
- More flexible
  - Job steps
- Easier to use for simple jobs
  - Does more for you by default
- Complex features for complex jobs

# PBS to Slurm

What is the best best way to convert my job submission scripts?

- A few strategies:
  - Rewrite everything from scratch
  - **Rewrite only the preamble and try to reuse the rest.** (recommended)
  - Translate the lines that need to be changed
- This is a chance to learn the best practices.
  - Slurm can work better than PBS but only if used correctly.
  - Accurately request the resources (CPU/RAM) you need.
    - Jobs will start faster
    - More jobs will run
    - Job start prediction will be better

# Converting Scripts to Slurm

Rewrite only the preamble and try to reuse the rest.

1) Identify what the PBS preamble is requesting:
    a) How many nodes and CPU-cores are needed?
    b) How much RAM?
    c) How much time (wall clock limit)?
    d) Anything special?
2) Write the Slurm preamble.
3) Skip the change directory part!
4) Copy the actual compute part of the script.
5) Test the script.

# Converting Scripts to Slurm

```
####  PBS preamble
#PBS -N fmm_test
#PBS -m ae
#PBS -A test_flux
#PBS -q flux
#PBS -l nodes=1:ppn=2,pmem=2000mb
#PBS -l walltime=1:00:00
#PBS -j oe
#PBS -V
####  End PBS preamble
```

What you asked for:

- 1 node with 2 cores
- 2000 MB per core
- 1 hour
  - Usually takes 40 minutes
- Email at end and abort

```
##########  Slurm preamble
#SBATCH --job-name=fmm_test
#SBATCH --mail-type=END,FAIL
#SBATCH --account=test
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=2
#SBATCH --mem-per-cpu=2000m
#SBATCH --time=1:00:00
########## End of preamble!
```

```
if [ -s "$PBS_NODEFILE" ] ; then
    echo "Running on"
    uniq -c $PBS_NODEFILE
fi
```

Not needed, but you may want it.

```
if [[ $SLURM_JOB_NODELIST ]] ; then
    echo "Running on"
    scontrol show hostnames $SLURM_JOB_NODELIST
fi
```

```
~~if [ -d "$PBS_O_WORKDIR" ] ; then~~
~~    cd $PBS_O_WORKDIR~~
~~    echo "Running from $PBS_O_WORKDIR"~~
~~fi~~
```

Slurm does this for you.

```
#  Put your job commands after this line
fmm_engin input.dat output.dat
```

```
#  Put your job commands after this line
fmm_engin input.dat output.dat
```

# Avoid in Slurm Scripts

It is best to avoid these:

- Always requesting the maximum wall clock time
  - Higher wall clock request will have later start times.
- Always asking for 4000mb per core.
  - You are charged for what you use.
  - What you do not use now may be available for use later.
  - Remember this on demand billing.
- Asking for cores instead of nodes and cores per a node
  - In PBS this is procs=Z instead of nodes=X:ppn=Y, where Z=X*Y
  - Job may start faster, but they will run slower!

# On Demand

Great Lakes and Armis2 will not have monthly allocation like Flux.

Instead, they will use a model similar to Flux on Demand.

- We do not have the exact pricing, yet.
  - Cost will be based on actual monthly usage.
- Usage is determined by the percentage of the node used by the job.
  - Using all the CPU cores is the same as using all the RAM, that is using the whole node.
- The details are not final at this point.

# Questions?

- Keep in mind that the details are not final at this point.

- Any questions on the topics just covered?
  - Clusters: Great Lakes, Armis2, & ect...
  - On Demand
  - Something not mentioned?

- Please hold your detailed questions on the following topics until the end:
  - LMOD Software Modules
  - Slurm and Job Submission
  - PBS to Slurm script conversion

# Hands on Examples

- Please login to:

  greatlakes.arc-ts.umich.edu

- Link to examples:

  https://www-personal.umich.edu/~mmiranda/GLTutorial.pdf

# Thank You!

- Please contact hpc-support@umich.edu for help.

- Any questions?

  You can contact us:

  - coe-research-computing@umich.edu
  - https://caen.engin.umich.edu/hpc/