Introduction to HPC @ RCC

September 11, 2018

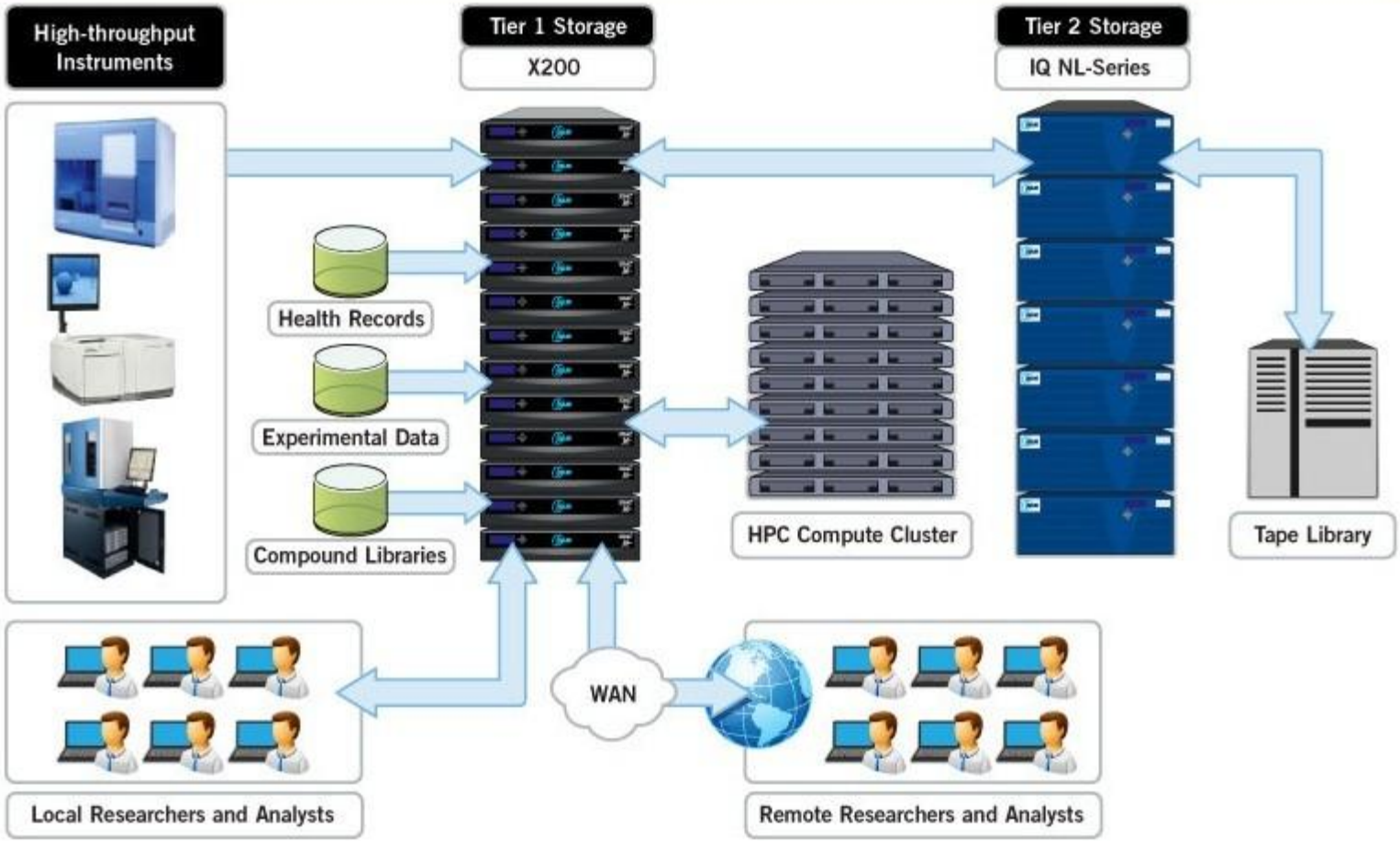
Research Computing Center

FLORIDA STATE UNIVERSITY
RESEARCH COMPUTING CENTER

What is HPC

"High Performance Computing most generally refers to the practice of aggregating computing power in a way that delivers much higher performance than one could get out of a typical desktop computer or workstation in order to solve large problems in science, engineering, or business"

Typical HPC Workflow



How to allocate resources?

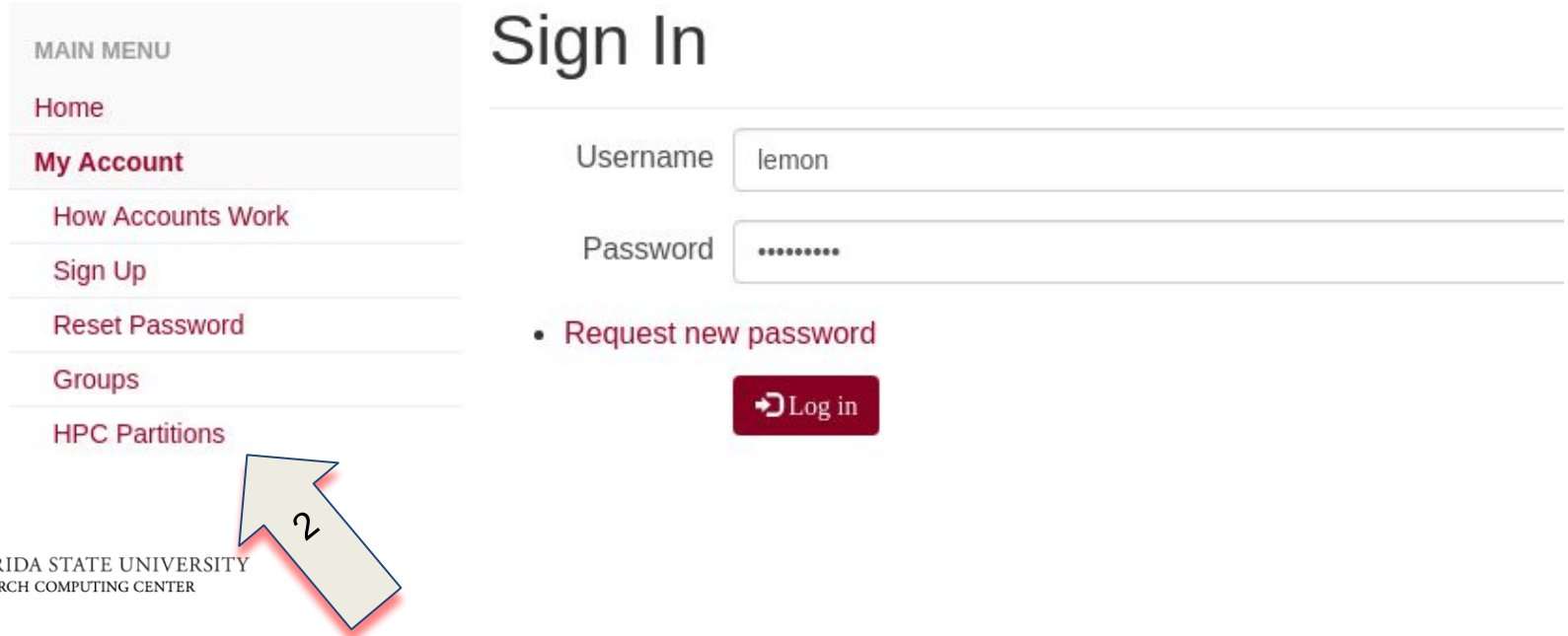
Job scheduler

Partitions?

- Collection of nodes
- Public (general access) and owner nodes
- Access is granted through a unix group
- Partitions spawn different architectures
  - Owner has bought nodes from different years
  - Jobs can not spawn different architectures

Request access to Partition

1. Login at https://acct.rcc.fsu.edu/account



Request access to Partition

2. Request membership to partition

SEC4M Partition	sec4m_q	2160:00:00	336:00:00	0	500	You Have Access
Engineering Partition	engineering_q	06:00:00	06:00:00	0	512	You Have Access
Engineering Long Partition	engineering_long	48:00:00	48:00:00	0	280	You Have Access
RCC Internal Partition	rcc_internal	72:00:00	36:00:00	0	512	You Have Access
Bleiholder Queue	bleiholder_q	2160:00:00	336:00:00	0	144	Get Access »
Deprince Queue	deprince_q	2160:00:00	336:00:00	0	160	Get Access »
Statistcs Queue	statistics_q	2160:00:00	336:00:00	0	80	Get Access »
Shangchao Lin Queue (AME)	lin_q	2160:00:00	336:00:00	0	208	Get Access »

How to submit a job

1. Command line
   - Ssh to hpc-login.rcc.fsu.edu
   - Use srun/sbatch
2. Web interface (script generator)
   - Currently only generates scripts and you have to use (1) to submit the script

How to submit a job (ssh)

1. **sbatch**
   non-interactive batch submission
   schedules job in background
2. **srun** & **salloc**
   interactive submission
   srun/salloc run program in foreground
   srun can also be used in batch script!

Submit jobs: sbatch

```
sbatch {flags} myscript
```

- man sbatch
- sbatch -p myqueue -n 10 myscript
  - request 10 cores from the myqueue queue and run *myscript* job script
- sbatch myscript
  - request 1 core from my default queue
- sbatch -D myproject/workdir myscript
  - start job in $HOME/myproject/workdir folder

srun to submit a job

- man srun
- srun from a submit node will start a new job
  - srun -p myqueue myprogram
- will not run in the background (unless &)
- srun -n *x* myscript.sh will start *x* instances of myscript.sh
  - srun will not “interpret” scripts: ignore #SBATCH flags
- The salloc is similar to srun, but be careful!

srun in job scripts

- slurm enabled replacement of mpirun
- mpirun is no longer supported (mvapich2)
- srun myprogram
  - will run myprogram on requested number of cores (sbatch -n *x*)
- srun -n *y* myprogam
  - will run myprogram on *y* number of cores
  - error if *y*>*x* (sbatch -n *x*)
- be careful when you use srun in a script submitted by srun

Interactive jobs --pty

srun --pty someprogram

srun --pty /bin/bash

srun --pty R

srun --pty gdb myprogram

- srun -n *x* --pty program will start 1 instance
- srun will start from your submit directory

Commands

SLURM		
sbatch	sbatch -p myqueue myjobscript.sh	Submit a batch script
srun & salloc	srun myprogram.exe salloc myprogram.exe	Submit an interactive program
squeue	squeue -p mypartition	Show jobs in a mypartition
squeue	squeue -j 1251 scontrol show job 1251	Inspect a specific job
squeue	squeue -j 1252 --start	Show start time of job
scancel	scancel 1251	Cancel a job
sinfo	sinfo -p mypartition	Shows nodes in mypartition

https://rcc.fsu.edu/docs/hpc-cheat-sheet

s* caveats

- Jobs will start in the current working directory (unless -D flag was used)
- Job environment is a copy of your working environment (except for limits)
  - environment variables
  - be careful what modules you autoload in your ~/.bashrc
- sbatch is not for interactive jobs

Common flags for s*

- *-n number* : Request *number* of cores
- *-p partition* : Run a job on this queue
- *-C feature* : Restrict job to nodes with this feature
- *--exclusive* : Do not share nodes with other jobs
- *-J jobname* : job name (not outputfile)
- *-o outputfile* : output file (default slurm)
- *--mail-type=X* : receive this type of notifications (ALL, BEGIN, END, FAIL)

Less Common flags

- *--begin=time* : Start a job at time *time*
- *--output=slurm.%N.%j.out* : output log
- *--input=inputfile.txt* : use text from file for std input
- --pty : interactive job, only for srun!

Memory

- Slurm takes memory in consideration
- Default is 4GB per core (2GB backfill{2})
- --mem-per-cpu=<MB> or --mem=<MB>
- Under the hood: memory is “mapped” to cores:
  - -n 1 --mem=5GB will reserve 2 cores on a node.
- Memory limit is enforced

Job script for parallel program

```
#!/bin/bash

#SBATCH -J MYJOBNAME
#SBATCH -n 10

module load gnu-openmpi
pwd
srun myprogram
```

Run a sequential program

```
#!/bin/bash

#SBATCH -J MY-R-CODE
#SBATCH --input myRinput.txt

pwd
module load R
R --no-save
```

Script Generator

https://rcc.fsu.edu/submit-script-generator

- Interactively generate a slurm script
- Limited syntax checking
- Templates available for some software
- Submit jobs directly from website (future)

Script Generator Demo

Job Title

MyProgram

Create a name for your job (alphanumeric, dashes, and spaces allowed)

Executable Call

test

Enter the program you wish to run for your job. If you pipe input or pass arguments, include those.

Separate Results and Verbose Output

☐ Separate Verbose Output from Results

Email Notifications

☑ On Job Start ☑ On Job End ☑ If Job Fails ☐ If Job Requeues

Please select the type of email notifications about your job you would like to receive.

HPC Partition

Backfill 2 Queue (backfill2) -- General Access

Number of Cores

4

Select the number of processor cores your job will run on.

Number of Nodes

2

Select the number of compute nodes your job will run on. Note that adjusting this does not guarantee that processes will be evenly distributed across all nodes. The default is "No Preference" and the Number of Processes `#SBATCH -n` are adjusted instead. If Number of Nodes is set to "No Preference"

SLURM Submission Script

```
#!/bin/bash
#SBATCH --job-name=MyProgram


#SBATCH --mail-type=BEGIN,END,FAIL
#SBATCH -n 4

#SBATCH -N 2
#SBATCH -p backfill2
#SBATCH -t 04:00:00









module load gnu-openmpi/2.1.0


## Submit Script Generator automatically adde
srun test
```

To use this script:

Why is my job not running?

- Partition does not have enough cores available?
- You ask for too much memory?

```
squeue -u $(whoami)
squeue -p mypartition
scontrol show job jobid
```

I need

- Submit a request to support@rcc.fsu.edu
- Include the path to your job script and output files
- Include the error you received
- If possible, include job id.

Job Array

- Job arrays are a way to efficiently submit large numbers of jobs.
- Single program with a lot of different datasets
- sbatch **--array=1-10** program.sh
  - $SLURM_ARRAY_TASK_ID

Job dependencies

- Scheduling of job is conditional
- For example, a job can only run when another job has finished
  - `#sbatch --dependency=afterok:`*`otherjobid`*
- For example, job can only run when no job of the same “type” (name) runs
  - `#sbatch --dependency=singleton`
    `#sbatch --job-name=`*`jobname`*

Shared MPI space

- Share MPI communicator space with multiple programs
- Define cpu mapping in `layoutfile`

```
0-7  ./prog1
8-15 ./prog2
```

- `srun -n16 --multi-prog layoutfile`