Intro to Monsoon and Slurm

ENABLE ZOOM! (1 remote)

2/12/2019
Slides:
https://rcdata.nau.edu/hpcpub/workshops/intro.pdf

NORTHERN ARIZONA UNIVERSITY

Get logged in!

- From a Windows pc:
  - Open the putty application
    - May need to search in start menu for it
  - In the hostname field:
    - your_louie_id@monsoon.hpc.nau.edu
  - Click open button
  - And accept the security key by typing ”y”

- From a Mac:
  - Open the terminal application
  - ssh your_id@monsoon.hpc.nau.edu

Introductions

- Introduce yourself
  - Name
  - Department / Group
  - What project(s) do you plan to use monsoon for?
  - Linux or Unix experience
  - Previous cluster experience?

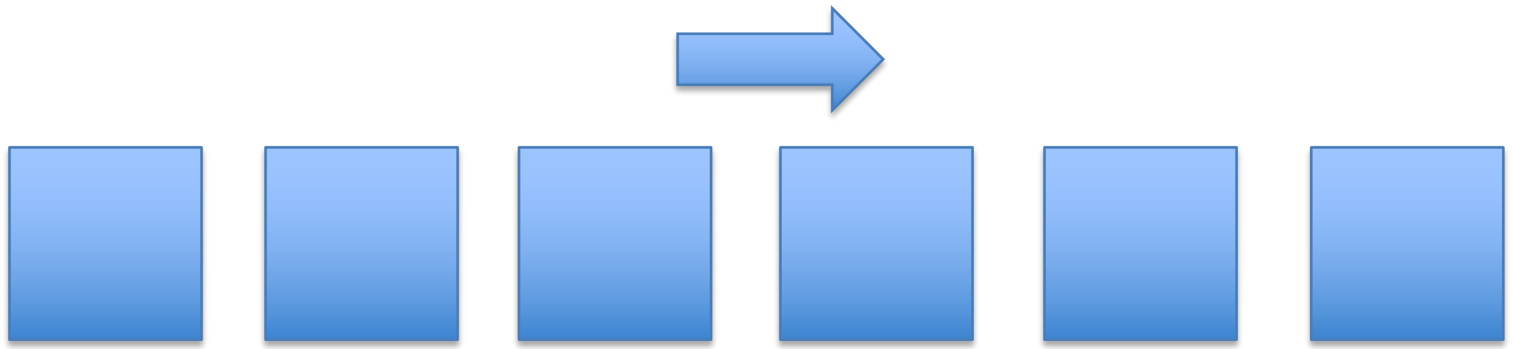
List of Topics

- Cluster education
  - What is a cluster, exactly?
  - Queues, scheduling and resource management
- Cluster Orientation
  - Monsoon cluster specifics
  - How do I use this cluster?
  - Group resource limits
  - Exercises
  - Question and answer

NAU NORTHERN ARIZONA UNIVERSITY

What is a queue?

- Normally thought of as a line, FIFO (Line at Starbucks)
- Queues on a cluster can be as basic as a FIFO, or far more advanced with dynamic priorities taking into consideration many factors



What is scheduling?

- *"A plan or procedure with a goal of completing some objective within some time frame"*
- Scheduling for a cluster at the basic level is much the same. Assigning work to computers to complete objectives within some time availability
- Not exactly that easy though. Many factors come into play scheduling work on a cluster.

Scheduling

- A scheduler needs to know what resources are available on the cluster in order to make accurate scheduling decisions
- Resource availability changes by the minute
- Assignment of work on a cluster is carried out most efficiently with the scheduler and resource manager working together

Resource Manager

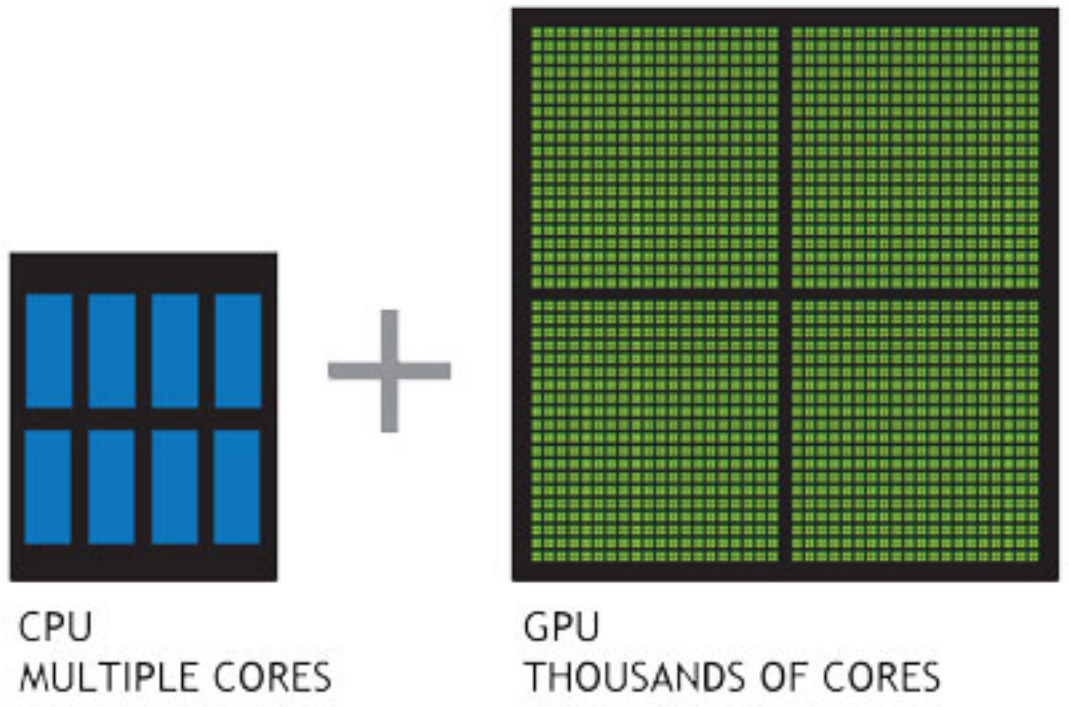
- Monitoring resource availability and health
- Allocation of resources
- Execution of resources
- Accounting of resources

Our Scheduling Goals

- Optimize quantity of work
- Optimize usage of resources
- Service all users and projects justly
- Make scheduling decisions transparent

Cluster Resources

- Node
- Memory
- CPU’s
- GPU’s
- Licenses

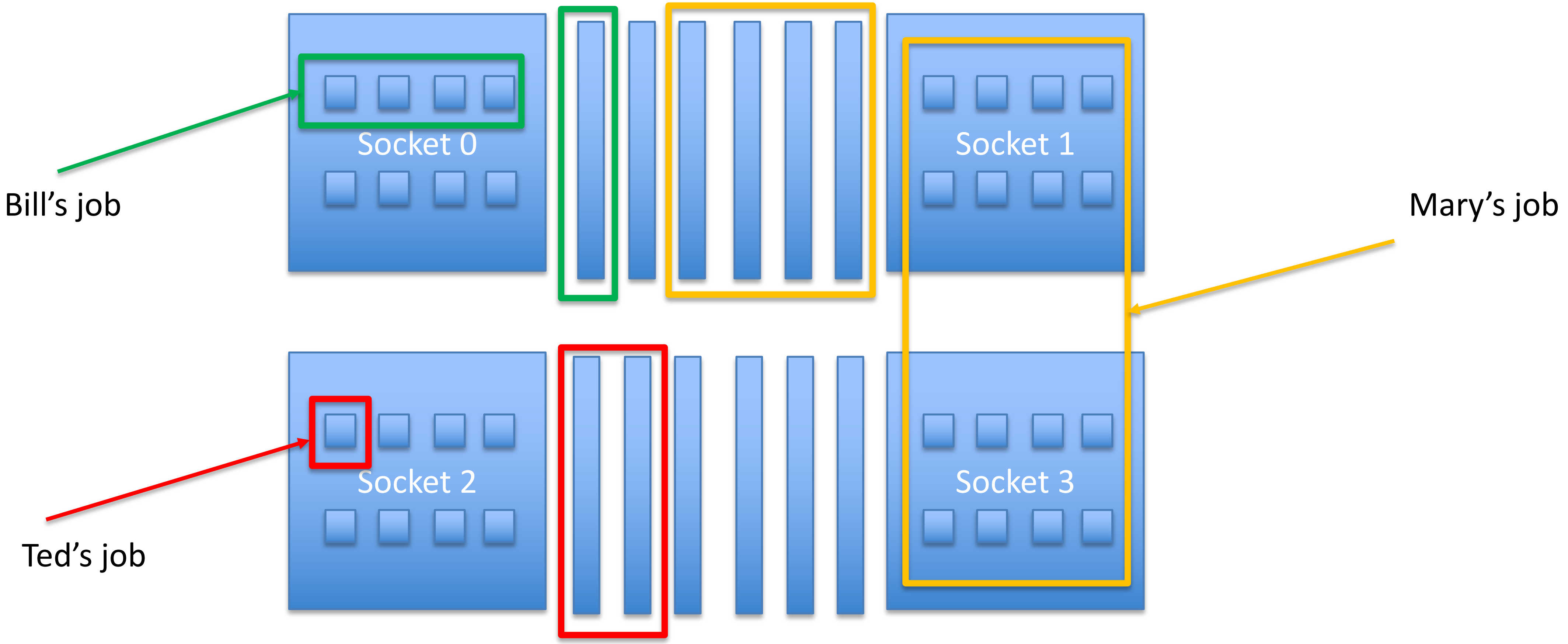


NORTHERN ARIZONA UNIVERSITY

Many scheduling methods

- FIFO
  - Simply first in first out
- Backfill
  - Runs smaller jobs with lower resource requirements while larger jobs wait for higher resource requirements to be available
- Fairshare
  - Prioritizes jobs based on a users recent resource consumption

Inside a Node



Monsoon Today

- The Monsoon cluster is a resource available to the NAU research enterprise
- 103 systems (nodes) – cn[3-105]
- 2860 Intel Xeon cores
- 16 GPUs, NVIDIA Tesla K80, and P100
- Red Hat Enterprise Linux 6.9
- 24TB memory - 128GB/node min, 1.5TB max
- 500TB high-speed scratch storage (Lustre)
- 615TB long-term storage (ZFS)
- High speed interconnect: FDR Infiniband

NORTHERN ARIZONA UNIVERSITY

Slurm … yummm

- Slurm (Simple Linux Utility for Resource Management)
- Excellent resource manager and scheduler
- Precise control over resource requests
- Developed at LLNL, continued by SchedMD
- Used everywhere from small clusters to the largest clusters:
  - Summit (#1), 2.4M cores, NVIDIA Volta GPUs, 200 PF, 9.7k kW - USA
  - Sierra (#2), 1.5M cores, 125 PF, 7.4k kW - USA

NAU NORTHERN ARIZONA UNIVERSITY

Small Cluster!

Dual core?

Largest Cluster!

2.4M cores

Monsoon scheduling

- Combination of scheduling methods
- Currently configured to utilize backfill along with a multifactor priority system to prioritize jobs

Factors attributing to priority

- Fairshare (predominant factor)
  - Priority points determined on users recent resource usage
  - Decay half life over 12 hours
- QOS (Quality of Service)
  - Some QOS have higher priority than others, for instance: debug
- Age – how long has the job sat pending
- Job size - the number of nodes/cpus a job is requesting

Storage

- /home – 10GB quota
  - Keep your scripts and executables here
  - Snapshotted twice a day: /home/.snapshot
  - Please do not write job output (logs, results) here!!
  - Run the command “quota” now
- /scratch – 500TB, 30 day retention
  - Very fast storage, capable of 11GB/sec
  - Quota: 20TB, 1M files
  - Checkpoints, logs
  - Keep all temp/intermediate data here
  - Should be your default location to perform input/output

NAU NORTHERN ARIZONA UNIVERSITY

Storage

- /projects – 615TB
  - Long-term storage project shares
  - 10TB is assigned to faculty member for group to share
  - $24/TB/year above 10TB
  - Snapshots available
  - Backups available - $.10/GB/month
- /common
  - Cluster support share
  - Contrib: place to put software/libs/confs/db's for others use

Data Flow

1. Keep scripts and executables in /home
2. Write logs/temp/intermediate data to /scratch
3. Copy data to /projects/<group_project>, for group storage and reference in other projects
4. Cleanup /scratch

** Remember, /scratch is a scratch filesystem, used for high-speed temporary, and intermediate data

Remote storage access

- scp
  - scp files nauid@monsoon.hpc.nau.edu:/scratch/nauid
  - WinSCP (windows)
  - Cyberduck (mac)
- samba / cifs
  - \\nau.froot.nau.edu\cirrus (windows)
  - smb://nau.froot.nau.edu/cirrus (mac)
- Globus
  - https://nau.edu/high-performance-computing/globus/

NAU NORTHERN ARIZONA UNIVERSITY

Groups

- NAU has a resource called Enterprise groups
- They are available to you on the cluster if you'd like to manage access to data
- https://my.nau.edu
  - "Go to Enterprise Groups"
  - Take a look at our FAQ :: nau.edu/hpc/faq
- If they are not working for you, contact ITS help desk
- What groups are you in? Run the command "groups", or "I"

Software

- ENVI / IDL
- Matlab
- Mathematica
- Intel Compilers, and MKL
- SAS
- R
- Qiime2
- Anaconda Python
- WRF
- Amber
- Tensorflow
- Lots of bioinformatics programs

- Full list here: “module avail”
- Request additional software to be installed!

NAU NORTHERN ARIZONA UNIVERSITY

Modules

- Software environment management handled by the *modules* package management system

- module avail – what modules are available
- module list – modules currently loaded
- module load <module name> - load a package module
- module display <module name> - detailed information including environment variables effected

MPI

- Quick note on MPI
- Message Passing Interface for parallel computing
- Open MPI set as default MPI
- Mpich, and Mvapich2 also available
  - module unload openmpi
  - module load mvapich2
- Example MPI job script:
  - /common/contrib/examples/job_scripts/mpijob.sh

Interacting with Slurm

- What resources are needed?
  - 2 cpus, 12GB memory, for 2 hours?
- What steps are required?
  - Run prog1, then prog2 … etc
  - Are the steps dependent on one another?
- Can your work, or project be broken up into smaller pieces? Smaller pieces can make the workload more agile.
- How long should your job run for?
- Is your software multithreaded, uses OpenMP or MPI?

Example Job script

- #!/bin/bash
- #SBATCH --job-name=test
- #SBATCH --output=/scratch/nauid/output.txt   # the stdout from your program goes here
- #SBATCH --time=20:00   # shorter time = sooner start
- #SBATCH --workdir=/scratch/nauid   # default location slurm searches

- # replace this module with software required in your workload
- module load anaconda/latest   # loads our supported conda distribution (v2)

- # example job commands
- # each srun command is a job step, so this job will have 2 steps
- srun sleep 300
- srun python -V

Job Parameters

You want	Switches needed
More than one cpu for the job	--cpus-per-task=2, or -c 2
To specify an ordering of your jobs	--dependency=afterok:job_id, or -d job_id
Split up the output, and errors	--output=result.txt --error=error.txt
To run your job at a particular time/day	--begin=16:00 --begin=now+1hour --begin=2010-01-20T12:34:00
Add MPI tasks/ranks to your job	--ntasks=2, or -n 2
To control job failure options	--norequeue –requeue
To receive status email	--mail-type=ALL

NAU NORTHERN ARIZONA UNIVERSITY

Contraints and Resources

You want	Switches needed
To choose a specific node feature (e.g. avx2)	--constraint=avx2
To use a generic resources (e.g. a gpu)	--gres=gpu:tesla:1
To reserve a whole node for yourself	--exclusive
To chose a partition	--partition

Submit the script

```
[user1@wind ~ ]$ sbatch jobscript.sh
Submitted batch job $\boxed{85223}$
```

- slurm returns a job id for your job that you can use to monitor or modify constraints

Login node vs Compute node

- When you log into “monsoon” you are placed on a login node
- The login node is a shared system used solely for:
  - Developing scripts
  - Transferring data
  - Submitting work to the scheduler
  - Analyzing results
  - Debug work less than 30 minutes in length
- The compute nodes are what make the supercomputer “super”.

NAU NORTHERN ARIZONA UNIVERSITY

Interactive / Debug Work

- Run your compiles and testing on the cluster nodes by:
  - srun –p all gcc hello.c –o a.out
  - srun –qos=debug -c12 make -j12
  - srun Rscript analysis.r
  - srun python analysis.py

  - Try this now:
    - srun hostname
    - hostname

Long Interactive work

- salloc
  - Obtain a SLURM job allocation that you can work with for an extended amount of time interactively. This is useful for testing/debugging for an extended amount of time.

```
 [user1@wind ~ ]$ salloc -c 8 --time=2-00:00:00
 salloc: Granted job allocation 33442
[user1@wind ~ ]$ srun python analysis.py
[user1@wind ~ ]$ exit
salloc: Relinquising job allocation 33442
[user1@wind ~ ]$ salloc -N 2
salloc: Granted job allocation 33443
[user1@wind ~ ]$ srun hostname
cn3.nauhpc
cn2.nauhpc
[user1@wind ~ ]$ exit
salloc: Relinquising job allocation 33443
```

NAU NORTHERN ARIZONA UNIVERSITY

Monitoring your job

- squeue
  - view information about jobs located in the SLURM scheduling queue.
- squeue --start
- squeue -u login
- squeue -o “%j %u … “
- squeue -p partitionname
- squeue -S sortfield
- squeue -t <state> (PD or R)

NORTHERN ARIZONA UNIVERSITY

Cluster info

- sinfo
  - view information about SLURM nodes and partitions.
- sinfo -N –l
- sinfo –R
  - List reasons for downed nodes and partitions

Monitoring your job

- sprio
  - view the factors that comprise a job’s scheduling priority
- sprio –l

  -- list priority of users jobs in pending state
- sprio -o “%j %u … “
- sprio -w

Monitoring your job

- sstat
  - Display various statistics and information of a running job
- sstat -j jobid
- sstat -o AveCPU,AveRSS

- Only works with jobs where analysis is executed with “srun”

Controlling your job

- scancel
  - Used to signal jobs or job steps that are under the control of Slurm.
- scancel -j jobid
- scancel -n jobname
- scancel -u mylogin
- scancel -t pending (only yours)

Controlling your job

- scontrol
  - Used to view and modify Slurm configuration and state.
  - Can change job constraints while it’s in pending state, once the job starts, it can no longer be modified
- scontrol show job 85224
- scontrol update jobid=6880341 timelimit=4:00:00

NAU NORTHERN ARIZONA UNIVERSITY

Job Accounting

- sacct
  - displays accounting data for of your jobs and job steps in the SLURM job accounting log or SLURM database
- sacct -j jobid -o jobid,elapsed,maxrss
- sacct -N nodelist
- sacct -u mylogin

- Try our alias “jobstats”
  - jobstats -r
  - jobstats -j <jobid>

Job Accounting

- sshare
  - Tool for listing the shares of associations to a cluster.
- sshare -l : view and compare your groups cpu minutes usage
- sshare -a : view all users fairshare
- sshare –A –a <account> : view all members in your account (group)
- group_efficiency <account>

Account hierarchy

- Your user account belongs to a parent faculty account (group)
- Your user account shares resources that are provided for your group
- Example:
  - account1
    - user1
    - user2
- View the account structure you belong to with: “sshare -a –A <account>”
- Example:
  - sshare -a -A account1

Limits on the account (group)

- Limits are in place to prevent intentional or unintentional misuse of resources to ensure quick and fair turn around times on jobs for everyone.
- Groups are limited to a total number of cpu minutes in use at one time: 5M, and gpu minutes: 64K
- This cpu resource limit mechanism is referred to as: “TRESRunMins”.
- This limiting mechanism has nothing to do with priority!

TRESRunMins Limit

- What the heck is that!?
- A number which limits the total number of remaining cpu minutes which your ***running*** jobs can occupy.
- Enables flexible resource limiting
- Staggers jobs
- Increases cluster utilization
- Leads to more accurate resource requests

- Sumofjobs(cpus * timelimit remaining)

Examples

- 14400 = 10 jobs, 1 cpu, 1 day in length
- 144000 = 10 jobs, 10 cpu, 1 day in length
- 720000 = 10 jobs, 10 cpu, 5 days in length
- 720000 = 1000 jobs, 1 cpu, ½ day in length
- 1105920 = 1 job, 1024 cpus, 18 hrs in length

Questions?

- Check your groups cpu min usage:
  - sshare -l

NORTHERN ARIZONA UNIVERSITY

TRES run minutes (demo)

- Say, groupA’s total cpu minute limit is: 5000
- Example, groupA submits three jobs
  - Job1:
    - 1 core
    - 1 day timelimit (1440 minutes)
    - 1 GB memory
  - Job2:
    - 2 core
    - 1 days (1440 minutes)
    - 16 GB memory
    - 2880 minutes total !
  - Job 3:
    - 1 core
    - 1 day (1440 minutes)
    - 1GB memory

TRES run minutes

– Assuming there are available monsoon resources
– How many jobs start?

– How many cpu minutes are in use?

– When is job 3 ELIGIBLE to start?

TRES run minutes

- Assuming there are available monsoon resources
- How many jobs start?
  - 2
- How many cpu minutes are in use?
  - 1440+2880 = 4320
- When is job 3 ELIGIBLE to start?
  - After ~6 hours (6*60 = 360), and 2 jobs (360*2) = 720 minutes

  - We have only 5000-4320 = 680 minutes available initially
  - After ~ 1/4 day goes by (360 minutes) * 2 (two jobs) = 720 minutes
  - 680 + 720 = 1400
  - After another 40 minutes we’ll have 1440 at which point job starts

Helpful Linux Commands

List Files	ls options -l – to show more information
Change Directory	cd <directory path> cd by itself will return you to your home directory
Show/print current working directory	pwd
Copy Files	cp <source> <destination> use a period for the destination to copy a file to your current directory
Move or rename a file	mv <source> <destination>
Delete a file	rm <filename>
Create a directory	mkdir <directory name>
View contents of a file	more <filename> less <filename> cat <filename>
Edit a file	nano <filename>
Exit your terminal session (log off)	exit

NORTHERN ARIZONA UNIVERSITY

Exercise 1

Get to know monsoon and Slurm, on your own.

1. How many nodes make up monsoon?
   – Hint: use “sinfo”
2. How many nodes are in the **all** partition?
3. How many jobs are currently in the running state ?
   – Hint: use “squeue -t R”
4. How many jobs are currently in the pending state? Why?
   – Hint: use “squeue –t PD”

Exercise 2

- Create a simple job in your home directory
- Example here: /common/contrib/examples/job_scripts/simplejob.sh (copy it if you like ☺)
- Name your job: “exercise”
- Name your jobs output: “exercise.out”
- Output should go to /scratch/<user>/exercise.out
- Load the module “workshop”
- Run the “date” command
- And additionally, the “secret” command
- Save your job
- Submit your job with sbatch, i.e. “sbatch simplejob.sh”

NAU NORTHERN ARIZONA UNIVERSITY

Exercise 3

- Edit the jobscript file from previous exercise 2
- Make your job sleep for 5 minutes (sleep 300)
  - Sleep is a command that creates a lazy process that … sleeps and does nothing
- Monitor your job
  - squeue -u your_nauid
  - squeue -t R
  - scontrol show job jobnum
  - sacct -j jobnum
    - Inspect the steps

- Cancel your job
  - scancel jobnum

Exercise 4 (note Slurm bug)

- Copy job script and edit:
  - /common/contrib/examples/job_scripts/lazyjob.sh
- Edit the job with your user id
- Submit the job, it will take 65 sec to complete
- Use sstat and monitor the job
  - sstat -j <jobid>
- Review the resources that the job used
  - jobstats -j <jobid>
- We are looking for "MaxRSS", *MaxRSS is the max amount of memory used*
- Edit the job script, reduce the memory being requested in MB and resubmit, edit "--mem=" , e.g. --mem=600
- Review the resources that the optimized job utilized once again
  - jobstats -j <jobid>

- Ok, memory looks good, but notice that the usercpu is the same as the elapsed time

*Usercpu = num utilized cpus * elapsed time*

- This is because the application we were running only used 1 of the 4 cpus that we requested
- Edit the lazy job script, comment out first srun command, and uncomment the second srun command.
- Resubmit
- Rerun jobstats -j <jobid>, notice now usercpu is a multiple times the elapsed time, in this case (4). Because we were allocated 4 cpus, and **used** 4 cpus.
- Now address the egregious time estimate!

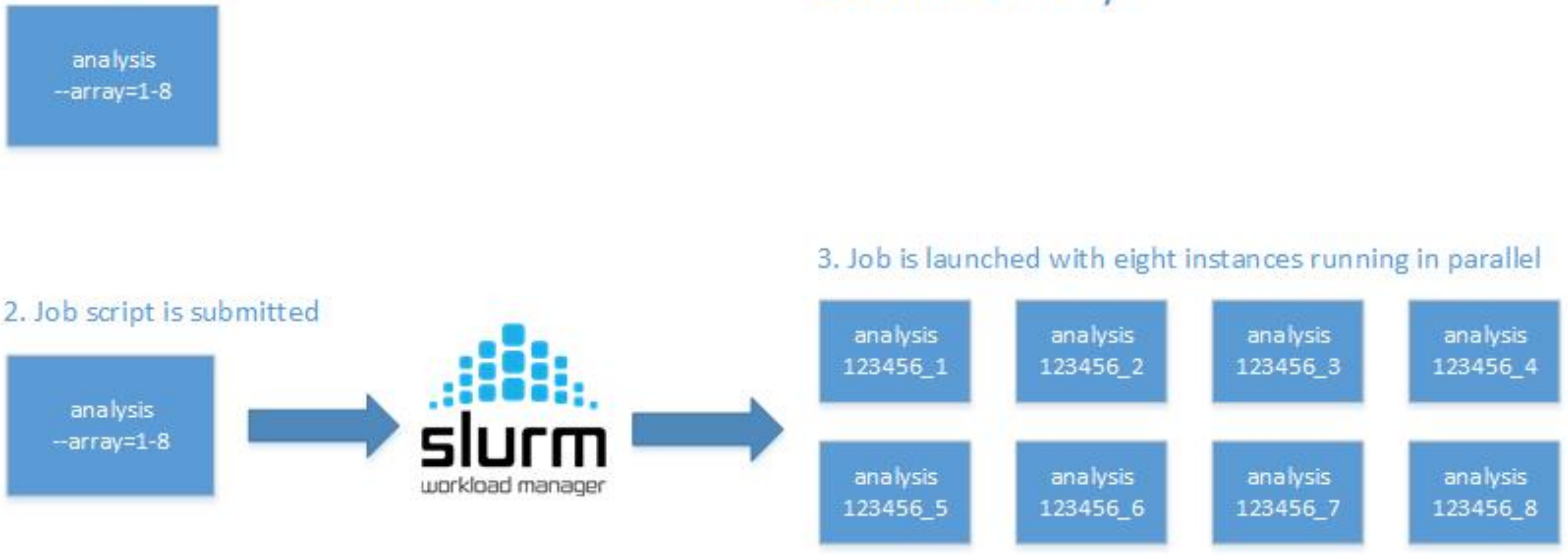
NORTHERN ARIZONA UNIVERSITY

Slurm Arrays

1. Job script is created

2. Job script is submitted

3. Job is launched with eight instances running in parallel



Useful environment variables

SLURM_ARRAY_JOB_ID:	the job array's ID (parent)
SLURM_ARRAY_TASK_ID:	the id of the job array member n (child)
%A	
%a	

Slurm Arrays Exercise

- From your scratch directory: “/scratch/nauid”
- tar xvf /common/contrib/examples/bigdata_example.tar
- cd bigdata
- edit the file “job_array.sh” so that it works with your nau id replacing all NAUID with yours
- Submit the script “sbatch job_array.sh”
- Run “squeue”, notice there are 5 jobs running, how did that happen!

MPI Example

- Refer to the MPI example here:
  - /common/contrib/examples/job_scripts/mpijob.sh
- Edit it, for your work areas, then experiment:
  - Change number of tasks, nodes … etc
- Also can run the example like this:
  - srun --qos=debug –n4 /common/contrib/examples/mpi/hellompi

Keep these tips in mind

- Know the software you are running, is it multi-threaded?
- Request resources accurately
- Supply an accurate time limit for your job
- Don’t be lazy, it will effect you and your group negatively

Question and Answer

- More info here:

  http://nau.edu/hpc

- FREE – Linux command line book:
  - http://linuxcommand.org/tlcl.php
- FREE - Intro to Linux command line video:
  - https://www.lynda.com/Linux-tutorials/Welcome/435539/482226-4.html
  - Info here: https://nau.edu/HPC/Linux-External-Resources/
- And on the nauhpc listserv
  - nauhpc@lists.nau.edu

NAU NORTHERN ARIZONA UNIVERSITY