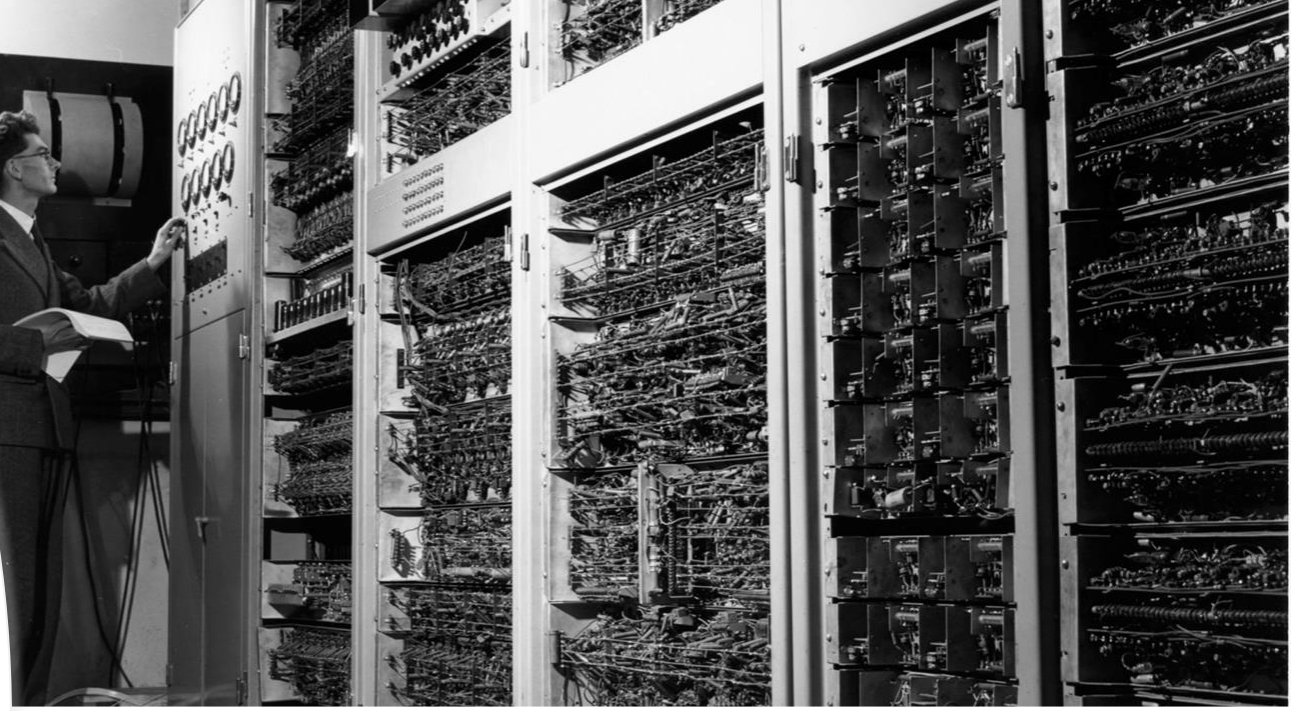
Introduction to Picotte

Bahrad A. Sokhansanj, Ph.D.

Rosen Group – EESI Lab
Drexel University
January 27, 2022



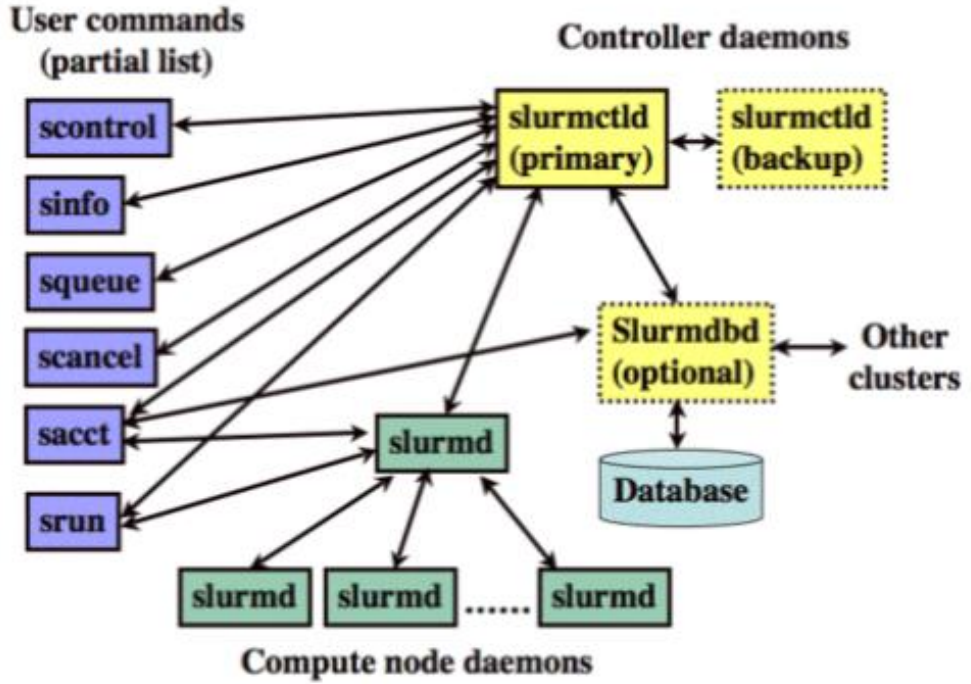
Linux

- Environment Variables are system wide variables
  - `$SHELL, $HOST, $USER`

- Type "`env`" or "`printenv`" in terminal to see all "`printenv VAR`" will display the value of `$VAR`

Slurm Workload Manager

- Originally "Simple Linux Utility for Resource Management" – open source cluster manager / job scheduler



Picotte Basics

- https://proteusmaster.urcf.drexel.edu/urcfwiki/index.php/URCF_Workshops_and_Talks
- https://proteusmaster.urcf.drexel.edu/urcfwiki/index.php/Introduction_to_Using_Picotte
- https://drexel.edu/now/archive/2021/April/Drexel-Names-New-Computing-_Cluster-After-Historic-Alumna/

Picotte

Login node:

- 1 Dell PowerEdge R640 server - Intel® Xeon® Platinum 8268 CPUs – 48 cores/server – 384 GB RAM/server

Compute nodes:

- 74 Dell PowerEdge R640 servers - Intel® Xeon® Platinum 8268 CPUs – 48 cores/server – 192 GB RAM/server
- 2 Dell PowerEdge R640 servers - Intel® Xeon® Platinum 8268 CPUs – 48 cores/server – 1.5 TB RAM/server
- 12 Dell PowerEdge C4140 servers - Intel® Xeon® Platinum 8260 CPUs – 48 cores/server – 192 GB RAM/server – 4 Nvidia Tesla V100-SXM2 32GB GPU devices/server

Storage:

- High performance parallel shared scratch: Dell EMC/BeeGFS Solution for HPC – 175 TB usable capacity utilizing HDR Infiniband providing aggregate 44 GB/s read, aggregate 41 GB/s write performance
- Persistent storage: Dell EMC PowerScale Isilon scale-out storage – 649 TB usable capacity utilizing 10 Gbps Ethernet
- Node-local scratch storage: 854 GB 12 Gbps SAS SSD (per node)

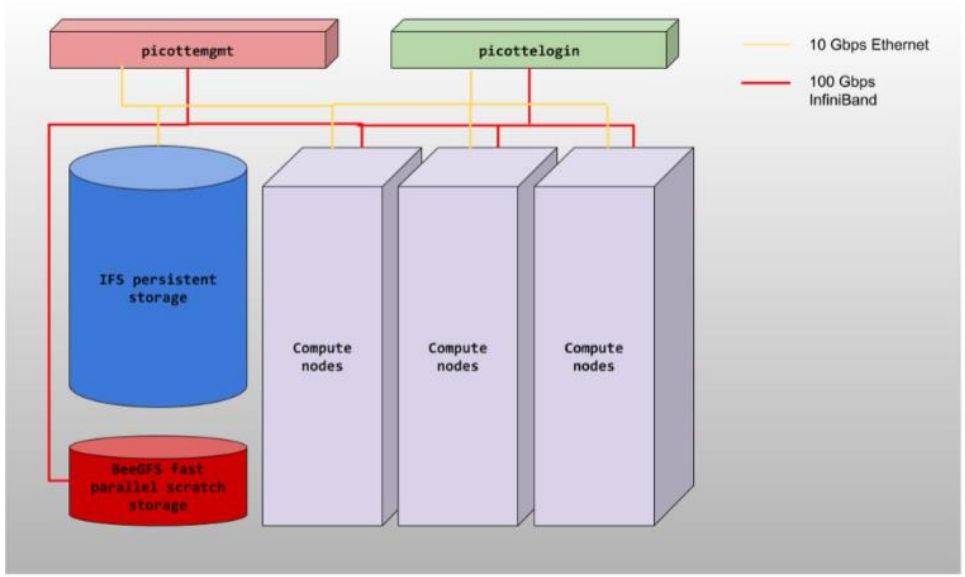
Network fabric

- 4X HDR Mellanox InfiniBand connected at 100 Gbps
- 10 Gbps Ethernet

Total of **4224 compute cores, 19.1 TiB RAM**

PICOTTE SOFTWARE

- Operating system: Red Hat Enterprise Linux 6 64-bit
- Job scheduler: Slurm



Using Picotte

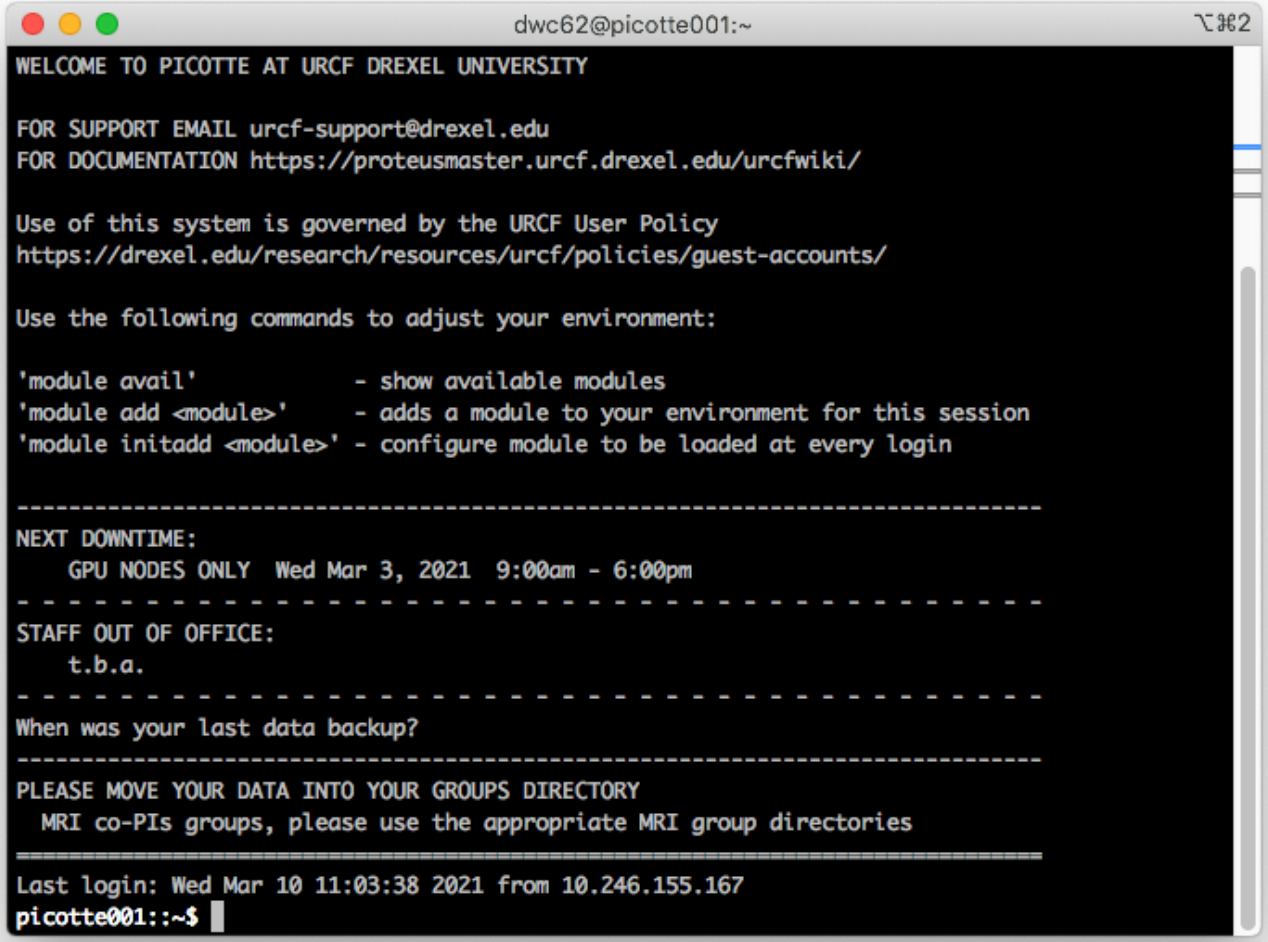
- Login
- Create or use installed program
- Create job script
- Submit the job script
- View the result

Using Picotte: Login

- Use Drexel's Virtual Private Network (VPN)
- Download Cisco Anyconnect VPN Client
- Access https://vpn.drexel.edu
  - https://drexel.edu/it/help/a-z/VPN/

- Use SSH Program (Terminal app on Mac; OpenSSH, PuTTy, MobaXTerm on Windows; etc.)
  - SSH to picottelogin.urcf.Drexel.edu

Using Picotte: Login



Using Picotte: Create Job Script

- https://proteusmaster.urcf.drexel.edu/urcfwiki/index.php/Writing_Slurm_Job_Scripts

- Start with `#!/bin/bash`
- `#SBATCH`
  - Starting at the first character of line:
  - `#SBATCH` options: `--partition`, `--mem`, `--time`, `--nodes`
- If not specified:
  - `#SBATCH --nodes=1`
  - `#SBATCH --ntasks=1`
  - `#SBATCH --cpus-per-task=1`

- (If job script was made on Windows, use dos2unix)

SBATCH (cont.)

- `--mail-user=user@host`
- `--account=bioworkshopPrj`
- `-p, --partition=<partition_names>`
  - `Ex: --partition=def`
- `-N, --nodes=numNodes`
  - `Ex: --node=16`
- `-t, --time=hh:mm:ss`
  - `Ex: --time=24:30:30`
- `--mem=size[units]`
  - `Ex: --mem=2GB`

Using Picotte: Submit Job

- `sbatch jobscript.sh`
- Output = "`Submitted batch job ___`"

- `squeue` (or `squeue -u username`)

```
(base) [bas44@picotte001 ~]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
           2317878       def stata-mp     ok85  R    1:04:41      1 node001
           2317877       def iqtree_d   crs344  R    2:34:28      1 node041
           2317876       def iqtree_s   crs344  R    2:36:15      1 node002
           2317874       def iqtree_p   crs344  R    2:39:07      1 node039
           2317765       def 1cpn-aut    aag99  R 1-03:52:04      1 node027
           2317764       def 1cpn-aut    aag99  R 1-04:11:55      1 node062
           2317761       def 1cpn-aut    aag99  R 1-04:30:46      1 node059
           2317867       def      s24    gs589  R   11:52:51      4 node[033-036]
           2317755       def      s57    gs589  R 1-07:23:26     23 node[002-024]
           2317849       gpu ABhex_GM    ba553  R   22:53:47      4 gpu[002,004-006]
```

Using Picotte: While the Job is Running

- squeue – view all running jobs
- sacct – get resource usage information after job (benchmarking)

- module load slurm_util
  - sin ('sinfo --Node -o "%12N %.6D %4P %.11T %.4c %.8z %.8m %.8d %.6w %.8f %58E“’)
  - sacct_disk ('sacct -o "JobID%20,JobName,User,Account,Partition,NodeList,Elapsed,State,ExitCode,MaxDiskRead,M axDiskWrite”’)
  - sacct_mem ('sacct -o "JobID%20,JobName,User,Account,Partition,NodeList,Elapsed,State,ExitCode,ReqMem,MaxR SS, MaxVMSize"')

Using Picotte: View Results

- `.out` and `.err` files

```
Warning: [blastp] Query_841 Train_68531: Could not calculate ungapped Karlin-Altschul parameters due to an invalid query sequence or its
translation. Please verify the query sequence(s) and/or filtering options
Warning: [blastp] Query_36033 Train_3819783: Could not calculate ungapped Karlin-Altschul parameters due to an invalid query sequence or
its translation. Please verify the query sequence(s) and/or filtering options
slurmstepd: error: Detected 1 oom-kill event(s) in step 1937132.batch cgroup. Some of your processes may have been killed by the cgroup o
ut-of-memory handler.
```

```
'rawnoise.txt' -> '/local/scratch/1937132/rawnoise.txt'
'trainfasta.txt' -> '/local/scratch/1937132/trainfasta.txt'
total 50772
-rw-rw-r-- 1 bas44 bas44 47115418 Dec 21 16:18 rawnoise.txt
-rw-rw-r-- 1 bas44 bas44  4873050 Dec 21 16:18 trainfasta.txt


Building a new DB, current time: 12/21/2021 16:18:33
New DB name:   /local/scratch/1937132/tempdb
New DB title:  /local/scratch/1937132/trainfasta.txt
Sequence type: Protein
Keep MBits: T
Maximum file size: 1000000000B
Adding sequences from FASTA; added 5616 sequences in 0.111545 seconds.
```

Job Array Results

- As a for-loop in Slurm.
- Each **subjob** is called an **array task**.
- Each array task is an iteration.
- All tasks have the same resource request
- Array job scripts include "`$SBATCH --array[0-N]`"
  - `N` is the number of array tasks

Job Arrays (cont.)

- `#SBATCH --array=n[-m[:s]][%c]` where:
- **n** -- start ID
- **m**--endID
- **s** -- step size
- **c** -- maximum number of concurrent tasks
- (items in [] are optional)

- Should also enable `requeue` for tasks to restart if they end unexpectedly

Important Environmental Variables

- Slurm sets some environment variables in every script and begin with "SLURM_”:

- `SLURM_JOB_ID`: Job ID
- `SLURM_CPUS_PER_TASK`: Number of CPU cores requested per task of your job
- `SLURM_ARRAY_JOB_ID`: Job Id of an array job
- `SLURM_ARRAY_TASK_ID`: Index number of the current array task.

Storage

- Scratch vs. Persistent

- Local (scratch)
  - Small and fast -internal SSD or HDD on nodes
  - `$TMP` or `$TMPDIR` automatically created for each job; deleted at end of job
- Path is `TMP=/local/scratch/${SLURM_JOB_ID}`

Storage (cont.)

- BeeGFS(or Lustre) (parallelscratch) –
  - Big and fast; can handle parallel I/O, which means multiple reads/writes from different nodes can be done simultaneously
  - Downside is that it may underperform for small files
  - Paths begin with "`/beegfs`"
    - `$BEEGFS_TMPDIR` –per-job automatically created
    - You can also create your own directory, for example: `/beegfs/scratch/myname`

Storage (cont.)

- NFS (persistent)
  - Big and slow(ish) -long term storage of research data, code
  - Paths begin with "`/ifs`" --different for every system

  - In general, avoid using NFS for any jobs, unless you're not doing lots of I/O, such as writing occasional status information.

Thank you

- For any questions or follow-up, please feel free to attend office hours (see weekly emails for details).
- Recordings and presentations from past workshops, including Intro to Picotte/Slurm, are available at the wiki site (accessible through VPN):
  - https://proteusmaster.urcf.drexel.edu/urcfwiki/index.php/URCF_Workshops_and_Talks
- Credit to Hoang Oanh Pham for her previous work on this presentation for past webinars