URCF Picotte

High Performance Computing @ Drexel

David Chin urcf-support@drexel.edu

Feb 2021

Goal of this talk

1. Quick orientation for users of Proteus who want to migrate to Picotte
2. Brief intro to Picotte for those new to HPC

URCF Governing Board

- Geoffrey Mainland, Chair (Computer Science, CCI)
- Antonios Kontsos (MEM, CoE)
- Gail Rosen (ECE, CoE)
- Lindsay Shea (Dir. Policy & Analytics Center, AJ Drexel Autism Inst.)
- Loni Tabb (Epidemiology & Biostatistics, Dornsife)

Who am I?

- Former physicist, not a computer scientist
- First programming job in Physics Dept at Cornell 1990
- First sysadmin job in Physics Dept at Cornell 1992
- Sysadmin & computational physics TA at Oregon State
- Worked on various physics/astrophysics related projects in grad school: Mini BooNE (neutrinos), high energy density plasmas (supernovae)
- PhD work in LIGO (leaders won Nobel 2018): contributions to algorithms library, detector characterization, data acquisition hw/sw
- Own 20-node cluster as postdoc in radiation oncology
- Past 12 years a sysadmin in academia

Our Team

- Systems administration: David Chin
- Network administration: David Chin
- Storage administration: David Chin
- Software administration: David Chin
- Database administration: David Chin
- Application development: David Chin
- User support and training: David Chin
- Co-op supervisor: David Chin
- Data center supervisor: David Chin
- MORAL OF THE STORY: please watch URCFHPCUSERS-L and email urcf-support@drexel.edu

What is URCF?

- University Research Computing Facility
- Data center
    - Houses Proteus and Picotte, Drexel’s supercomputers
    - Colocation facility: houses various faculty’s servers
        - Charge? None for Drexel people, except for one-time network fee
- Fast connection
    - 80 Gbps fiber to campus backbone
    - In-room 10 Gbps Ethernet (copper), static IP

What is Picotte?

- High performance computing cluster
  - Cluster: many (88) computers (*nodes*) connected together, sharing storage
  - Connection: high bandwidth low latency (100 Gbps bandwidth, < 0.2μs latency; cf. Proteus 32 Gbps, 1.3 μs latency)
  - Storage
    - 649 TB of persistent self-encrypting storage on Isilon scale-out (cf. Proteus 162 TB)
    - 175 TB of fast parallel BeeGFS scratch storage (cf. Proteus 87 TB)
    - 854 GB of local scratch SSD on every node (cf. Proteus 421 GB HDD)
- Where is it?
  - URCF data center in Curtis Hall, formerly the rifle range
  - Space for 38 racks of equipment, each rack up to 40 servers

Who paid for Picotte?

- Total cost (with Isilon expansion and upgrade to self-encrypting drives to support work with Protected Health Information, and add. compute nodes)

  ~ $1.5 million

- Partially funded by NSF Major Research Infrastructure (MRI) award 1919691
  - G. Rosen, H. Ayaz, A. Kontsos, B. Urbanc
- Remaining funding
  - Drexel Office of Research
  - Faculty startup funds J. Lequieu

Who will pay for the next cluster?

- Cost recovery via usage charges
  - Charges are based on a projected usage and expected lifespan of 4 years
  - Aim is for user charges to recover all cost, in order to refresh at the end of the 4-year lifespan
- New grants?
  - A lot of uncertainty

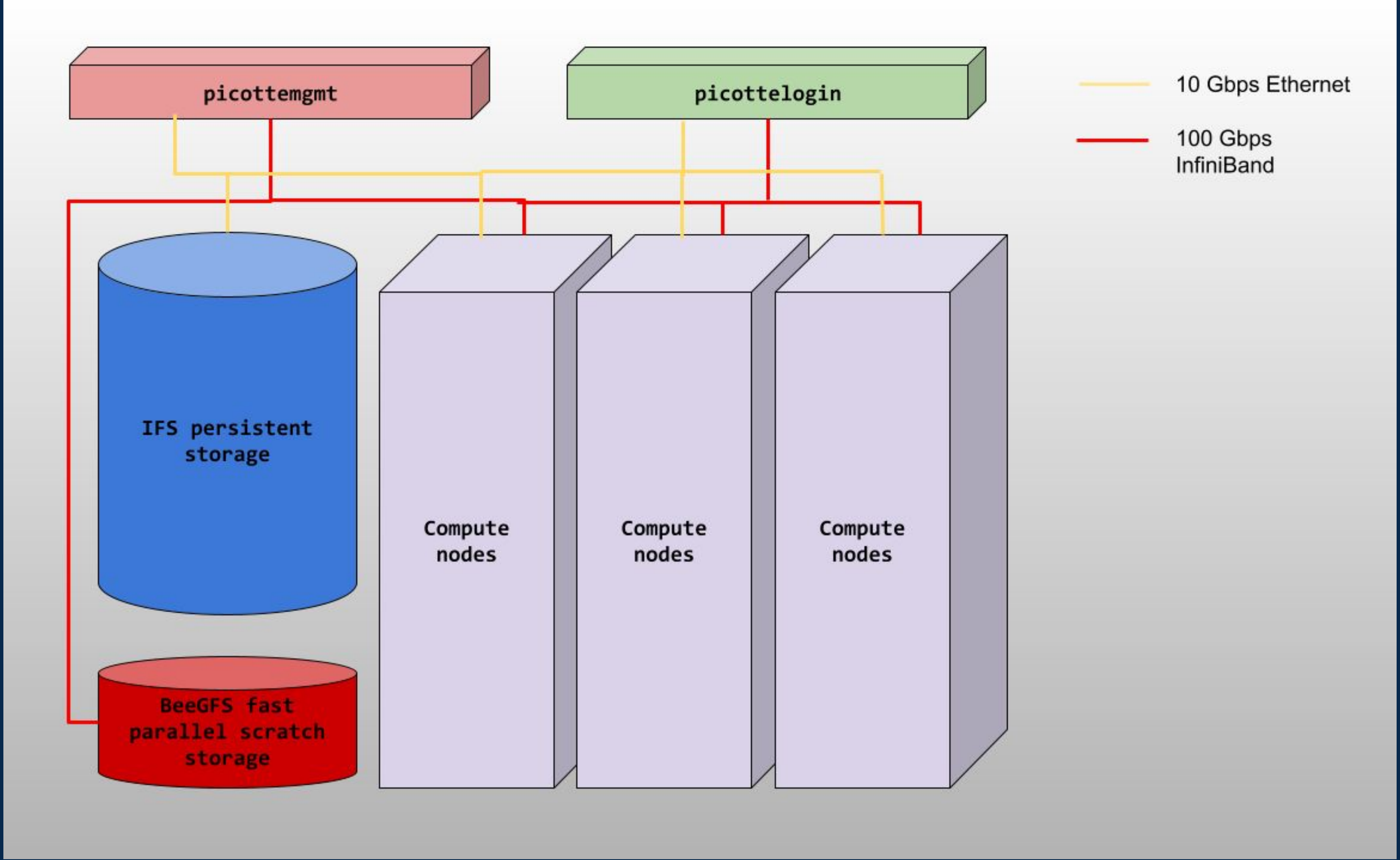
Current priorities

- Billing system, combining Picotte and Proteus charges
- Monitoring similar to old Ganglia

  https://proteusmaster.urcf.drexel.edu/ganglia-proteus/

Why do I need to know about the machines?

- How things work on a cluster depend strongly on the hardware involved
- A job which requires 1 TB of RAM (memory) to run will not run on a standard node (in the `def` partition)
- A job which launches an array of 1,000 separate processes, each writing to the local scratch drives of the nodes, and then copies all that output back to the group directory (on `/ifs/groups/somethingGrp`) may swamp the network-attached storage device that provides the group directory because at the end of the job you may have 1,000 separate processes all copying data to one directory at the same time


picottemgmt
picottelogin
10 Gbps Ethernet
100 Gbps InfiniBand
IFS persistent storage
Compute nodes
Compute nodes
Compute nodes
BeeGFS fast parallel scratch storage

358A
SERVER ROOM
NOTICE

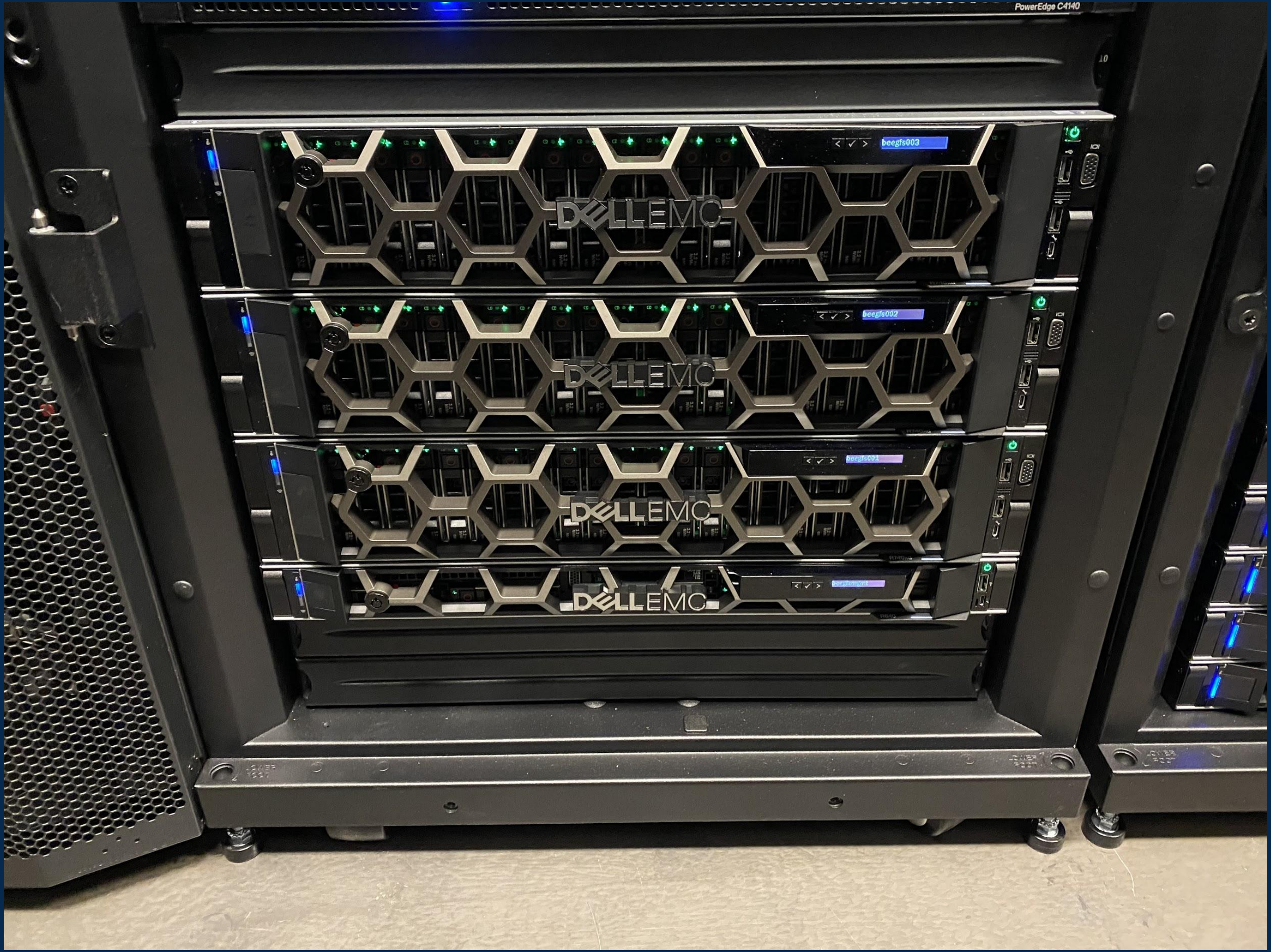
BM
BM

36
35
34
31
Casters:
50 lbs Max
1020 kg)
eveling Feet:
000 lbs Max
(1361 kg)
885-2771A
885-2661B(FRONT)
DELLEMC
2 TB SATA
R640

DELLEMC

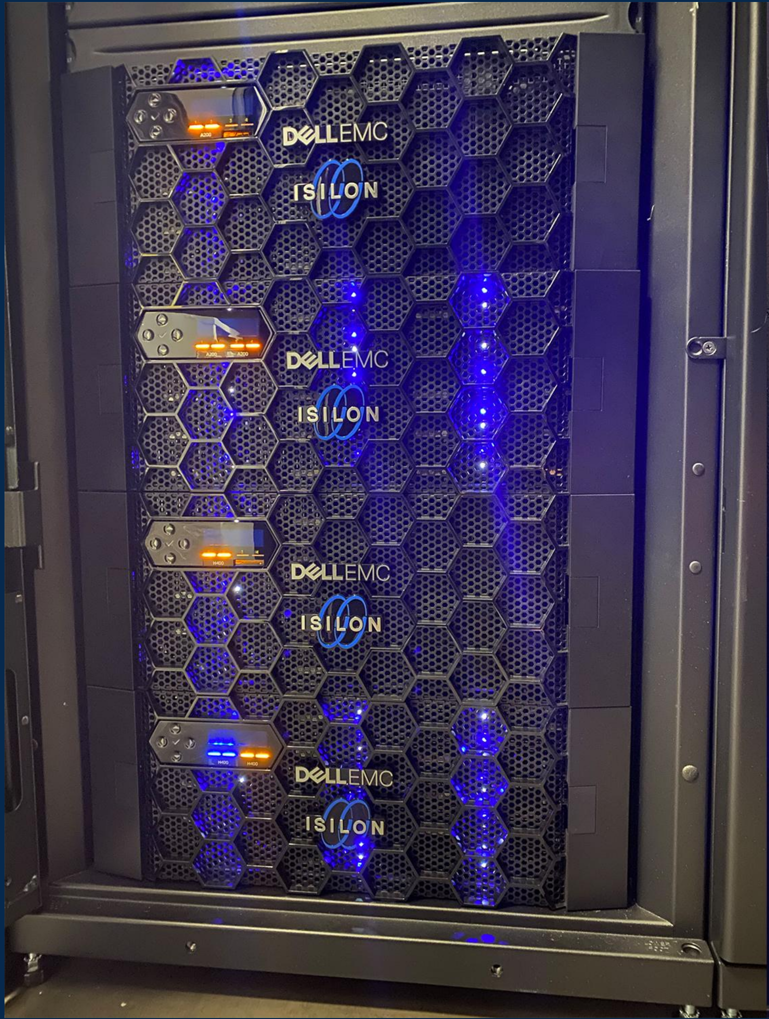
gpu006
DELL EMC
PowerEdge C4140
gpu005
DELL EMC
PowerEdge C4140
gpu004
DELL EMC
PowerEdge C4140
gpu003
DELL EMC
PowerEdge C4140
gpu002
DELL EMC
PowerEdge C4140
gpu001

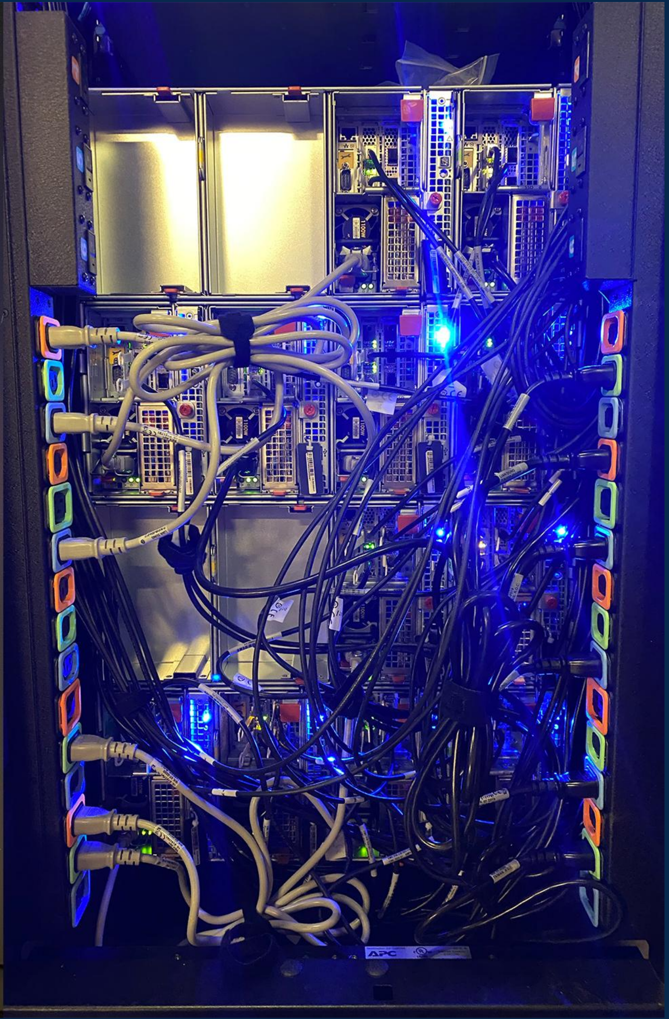
DELL EMC
ISILON
A200
H400


PowerEdge C4140
beegfs003
DELL EMC
beegfs002
DELL EMC
beegfs001
DELL EMC
DELL EMC

IB switch
Eth switch
Mellanox
R02 U29 PS1
R02 U30 PS1
R02 PDU2 B1
EPP 750W

41
40
39
38
37


DELLEMC
ISILON
DELLEMC
ISILON
DELLEMC
ISILON
DELLEMC
ISILON


APC

Typical workflow

- Login (ssh) to `picottelogin`
    - Drexel VPN required if off campus
    - Edit, test, debug
    - Write job script: specify resources and time needed
- Submit job script to scheduler
- Scheduler holds it in pending state until resources are available
- Job runs
- Current limits:
    - 1,776 cores simultaneously per project (may be changed depending on load etc.)
    - 48 hours wall clock time; up to 192 hours wall clock time on “long” partitions

Monitoring

- T.b.a.
- Goal: similar functionality to Proteus monitoring https://proteusmaster.urcf.drexel.edu/ganglia-proteus/

Nodes

- Standard compute
  - Partitions: `def, long`
  - 74 nodes: 2x Intel® Xeon® Platinum 8268 2.90 GHz 24-core; 192 GiB RAM
- Big memory compute
  - Partition: `bm`
  - 2 nodes: 2x Intel® Xeon® Platinum 8268 2.90 GHz 24-core; 1,536 GiB RAM
- GPU
  - Partitions: `gpu, gpulong`
  - 12 nodes
    - 2x Intel® Xeon® Platinum 8260 2.40 GHz 24-core; 192 GiB RAM
    - 4x NVIDIA Tesla V100-SXM2 (NVLink); 32 GiB RAM

Storage

- 649 TB NFS persistent storage
  - Dell EMC Isilon scale-out storage
  - Home and group directories
  - No backups; but snapshots available (r.s.n.)
- 175 TB fast parallel scratch
  - BeeGFS - cost about the same as the Isilon persistent storage
  - HDFS (Hadoop File System) interface available
  - Special feature: BeeOND (BeeGFS ON Demand)
    - ad hoc (per job) distributed parallel FS utilizing local scratch on compute nodes
- 854 GB node-local scratch per node (def); 1.7 TB RAID0 for GPU and bigmem

Network

- Internal
  - 2x HDR InfiniBand - up to 100 Gbps data rate; < 0.2μs latency
- Storage
  - Isilon connected to Picotte via 6x 10 Gbps Ethernet
  - Isilon connected to campus via 2x 10 Gbps Ethernet: storage space can be purchased
  - BeeGFS connected to Picotte via InfiniBand
- External
  - 10 Gbps Ethernet to outside world
- Server room connection
  - 80 Gbps fiber to campus backbone

Performance

- Measured performance (at commissioning) using LINPACK (dense linear algebra; basis of TOP 500 list `top500.org`)
  - 74 standard nodes: 145.9 TFLOPS
  - 2 big memory nodes: 5.5 TFLOPS
  - 12 GPU nodes (LINPACK CUDA): 251.0 TFLOPS
  - **Aggregate: 402.4 TFLOPS**
  - Compare Proteus: ~ 30 TFLOPS (theoretical peak performance) total
- BeeGFS i/o benchmarks
  - 15 threads write: 39.07 GiB/s
  - 15 threads read: 42.22 GiB/s
  - Compare best NVMe SSDs ~ 6.5 GiB/s

Software

- Commercial
    - Matlab
    - Mathematica
    - Stata
    - SAS
    - Abaqus
- Development
    - Open MPI
    - GCC 9.2; Intel Composer XE 2020u4 compiler & optimized math libraries
    - Java
    - CUDA 11.0

Software (cont.)

- More development
  - Julia, Go, Haskell,
  - Git: git, gh
  - Bazel
  - Cmake
- Open source etc.
  - Python, Perl, Go, Julia, R, octave, NCBI BLAST+ & NGS tools, etc.
- Containers using Singularity
  - Will run Docker images
- Spack for building software not currently provided https://spack.io

User Interface

- Command line terminal access with ssh
- Remote display of programs, e.g. Matlab, Mathematica
  - Requires additional software on your PC/laptop
- Limited availability
  - JupyterHub

Use Cases

- Designed for non-interactive computations
  - Able to handle millions of jobs in queue
- Not very suitable for interactive use, especially requiring graphics
  - Can be done but not ideal; using remote X11 display (slow, laggy)
- Not suitable for 3D-graphics-intensive interactive use
  - None of the nodes (including login node) has 3D graphics display capability
    - GPUs on GPU nodes are compute co-processors
- Not suitable for kernel-space and hardware development

Cost

- Colocation is free of charge
  - Except for one-time network port activation fee
- Charges with RCM implementation
  - Rates differ from Proteus; more bang (compute work) per buck with Picotte
    - Base rate: $0.0139 per Service Unit (SU)
    - Standard nodes: 1 core-hour = 1 SU ($0.0139) -> 1 node-hour = $0.667
    - GPU nodes: 1 GPU-hour = 43 SU ($0.598)
    - Big memory nodes: 1 TiB-hour = 68 SU ($0.945)
    - Storage: 1 TiB-hour = 1.48 SU -> $3.46 per TiB-week
      - Home directories: free use up to 64 GiB

Migrating from Proteus to Picotte

- Only 1 login node on Picotte: picottelogin.urcf.drexel.edu
  - Must use Drexel VPN for access, or be on campus
- DO NOT COPY LOGIN SCRIPTS OVER (`.bashrc`, `.bash_profile`, etc.)
- Job scheduler changed:
  - Proteus: Grid Engine
  - Picotte: Slurm
    - Slurm is in wide use; used in XSEDE
  - Proteus job scripts CANNOT be used on Picotte; sge2slurm script does partial conversion
- Storage paths changed
  - Proteus: /mnt/HA ➜ Picotte: /ifs
- Can ssh direct to node IF you have a job running on that node

Grid Engine to Slurm translation 1

Function	Grid Engine	Slurm
Interactive session	`qlogin`	`srun --pty bash`
Run prog. interactively	`n/q`	`srun --pty program`
Submit job	`qsub`	`sbatch`
Terminate job	`qdel`	`scancel`
Status of own jobs	`qstat`	`squeue --me`
Status of all jobs	`qstat -u \*`	`squeue -a`
Summ. node & q/part. state	`n/a`	`sinfo`

Grid Engine to Slurm translation 2

Grid Engine	Slurm
queue	partition
project	account
share tree (job scheduling)	fair tree
node down	node drained

Grid Engine to Slurm translation 3

- Script which partially automates translation from Proteus (Grid Engine) job script to Picotte (Slurm) job script:
    - `sge2slurm (use “--help” for online help)`
    - Usage:

```
sge2slurm oldscript.sh > newscript.sh
```

- NOTE
    - Translation is not perfect
    - Result must still be edited: check every line
    - See wiki for details - search for “Rosetta Stone”

Slurm documentation

- URCF Wiki:
  - https://proteusmaster.urcf.drexel.edu/urcfwiki/index.php/Category:Slurm
- In terminal: use `man`, like any other Linux command
- On web:
  - https://slurm.schedmd.com/documentation.html
  - `NB web documentation version may not match version installed on Picotte`
- Mailing lists:
  - SLURM-USERS (high traffic): https://lists.schedmd.com/cgi-bin/mailman/listinfo/slurm-users

Contact

- David Chin
  - urcf-support@drexel.edu
  - 215.571.4335
- Documentation:
  - https://proteusmaster.urcf.drexel.edu/urcfwiki/ (may change soon)

Questions?