

PACE Orientation OIT-ART

Mehmet (Memo) Belgin, PhD
Research Scientist, OIT-ART

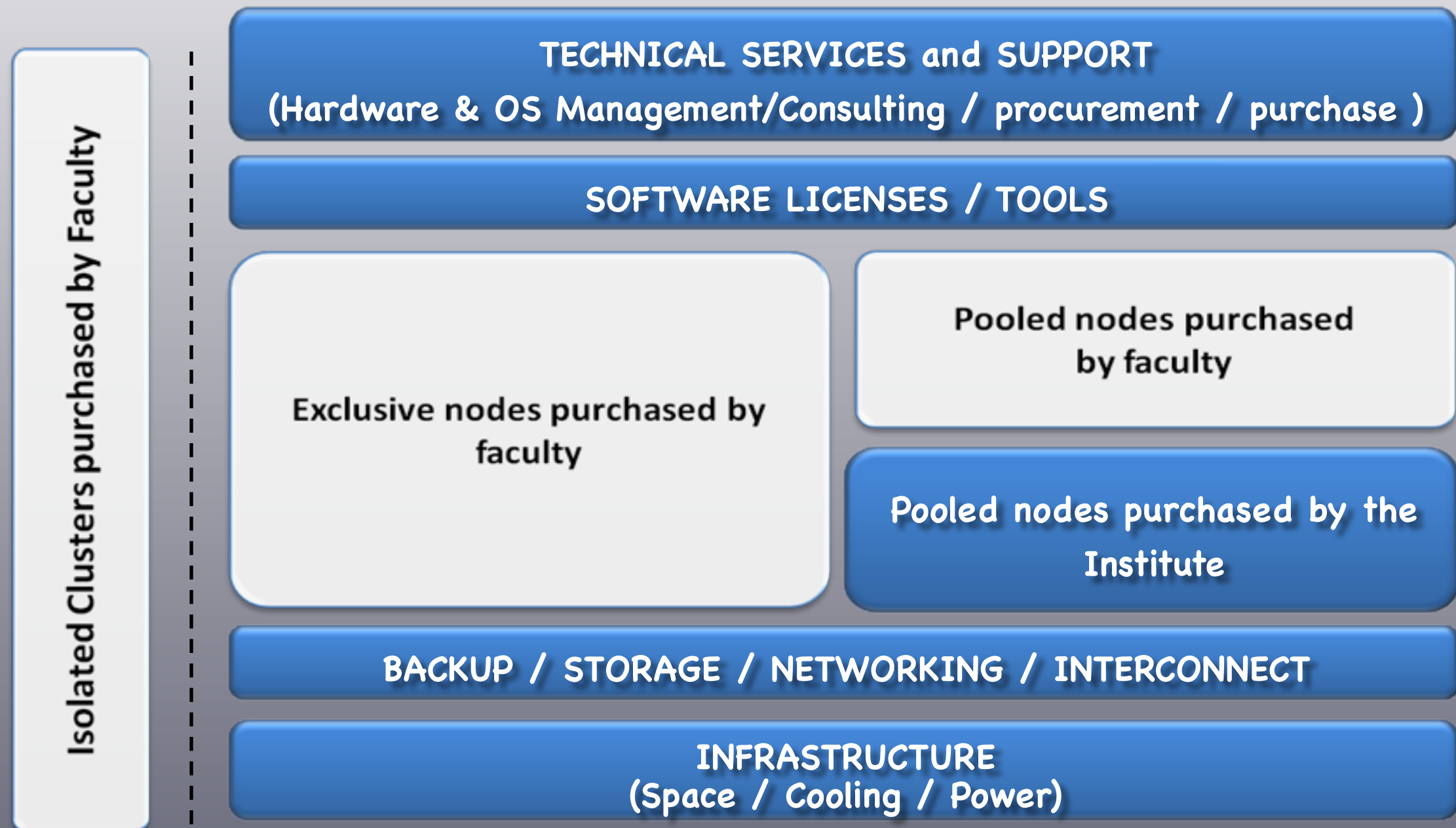
www.pace.gatech.edu

What is PACE

A Partnership for an Advanced Computing Environment

- Provides faculty and researchers vital tools to accomplish the Institute's vision to define the technological research university of the 21st century.
- A strong HPC environment through a tight partnership with our world-class students, researchers and innovators to help them make the greatest impact with their work.

The Big Picture



Legend:

GT dollars

Faculty dollars

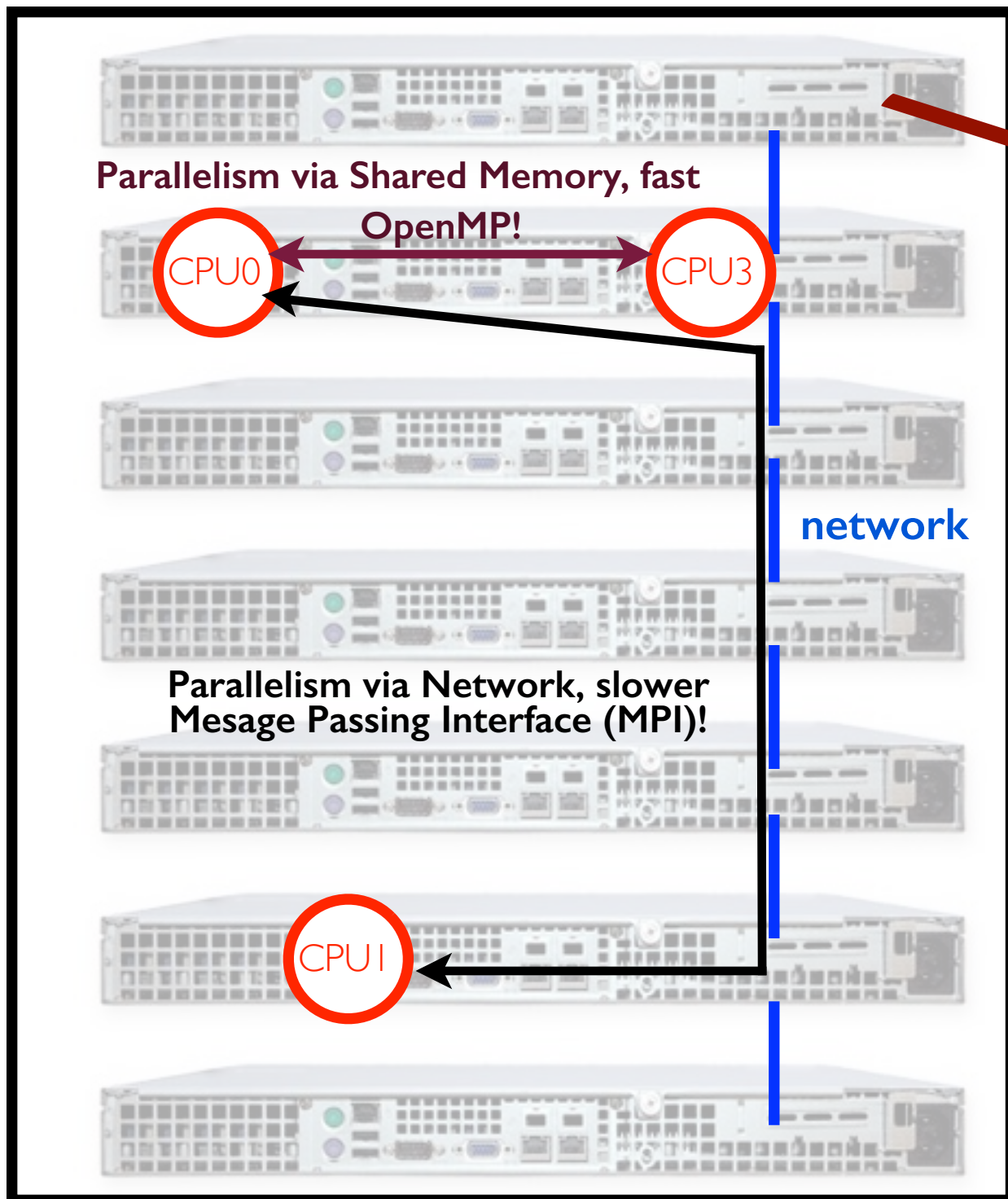
PACE cluster include a heterogeneous mix of nodes

E.g.: FoRCE-6 Cluster (subject to change)

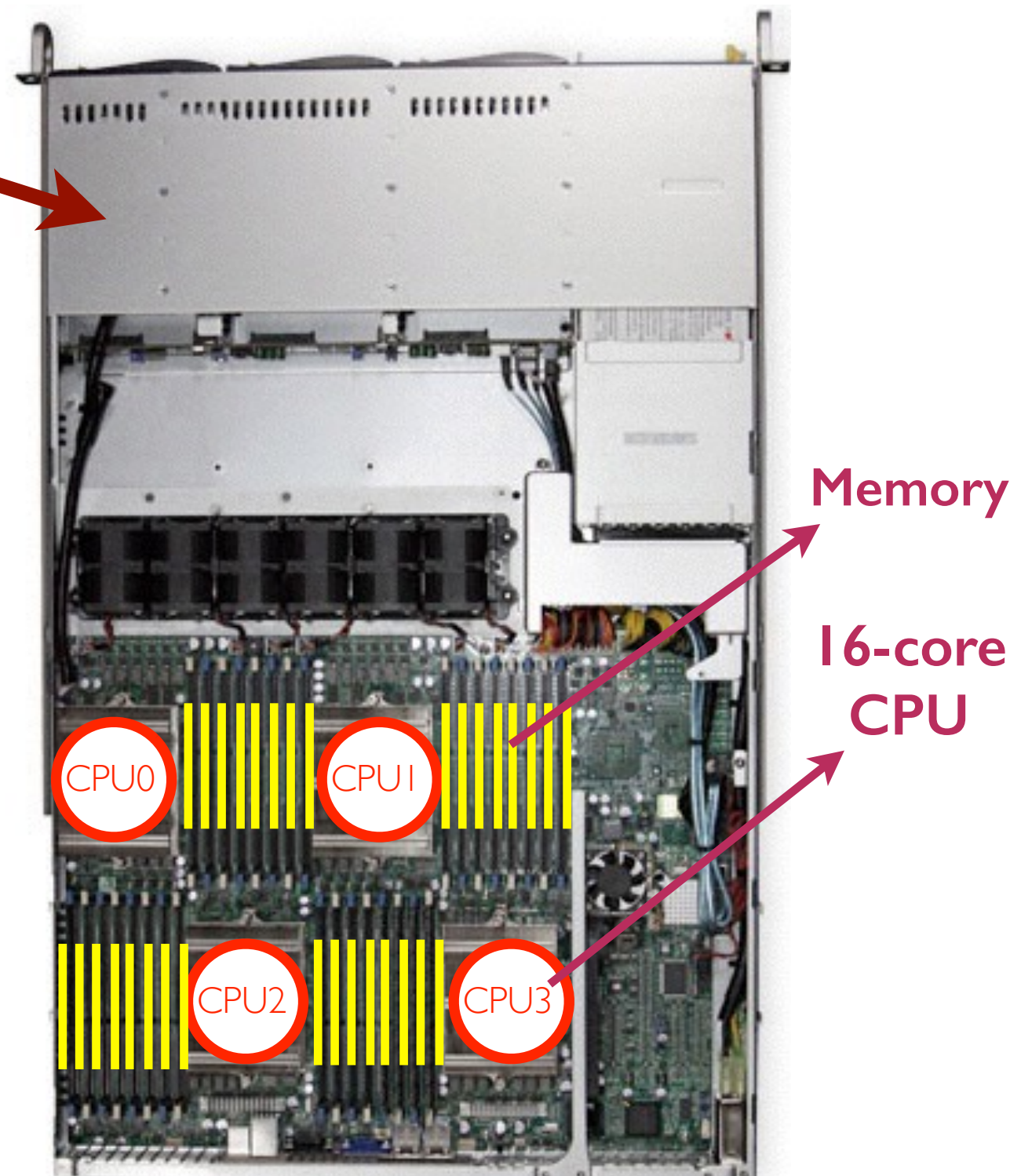
- Nodes available to GT researchers via a proposal process.
- 33 nodes consisting of:
 - 24-core, 64GB RAM, DDR Infiniband
 - 24-core, 64GB RAM, QDR Infiniband
 - 48-core, 128GB RAM, QDR Infiniband
 - 64-core, 256GB RAM, QDR Infiniband

Racks, Nodes and Cores

A “rack” of “nodes”



“cores” in a single Node



Getting Help is Easy:

- Email to open tickets:

`pace-support@oit.gatech.edu`

- Preferred: run this script on the cluster:

`pace-support.sh`

PACE Accounts

- Requires a valid GT account for access. Temps & guests are OK.
- Participations: by request from the PI
- on FoRCE: via proposals reviewed by Faculty Governance Committee.

Complete the online form: <http://pace.gatech.edu/node/add/request>

(login first if you see “Access Denied”)

Accessing Clusters

- You will need an SSH Client (a.k.a. terminal):
 - Windows: PuTTY, **Xming** (free), **X-win32** (via software.oit.gatech.edu)
 - MacOSX: iTerm2, Terminal, **XQuartz**
 - Linux: System-default terminal (gnome/KDE)
- SSH access to PACE clusters:

```
ssh -X <GT_user_ID>@<headnode>.pace.gatech.edu
```
- You need to be on campus, or connect via VPN.

For information on VPN access, see:

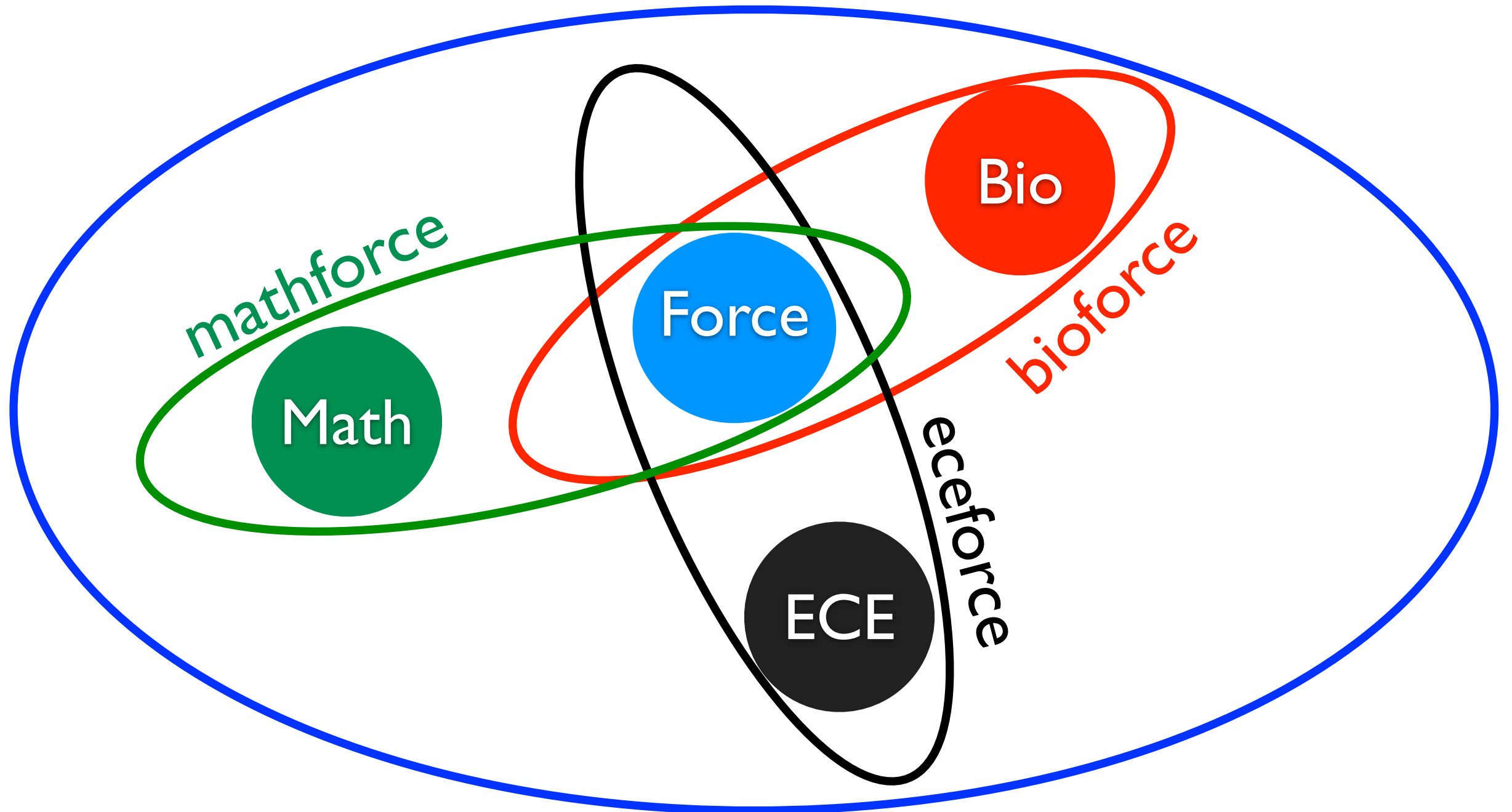
<http://faq.oit.gatech.edu/search/node/vpn>

PACE Queues

- A prioritized job list assigned to a pool of compute nodes.
- The scheduler (Moab) allows a fair use of shared resources within a queue (dynamic priorities, limits on walltime, CPU and RAM).
- Queues can be exclusive (private) or shared

Shared Queues

iw-shared-6



E.g. Biology Queues

- **biocluster-6**
 - 25 nodes
 - Priority: 270000
 - Max walltime: 90 days (or until maintenance, whichever comes first)
- **bioforce-6**
 - 58 nodes
 - Priority: 131042
 - Max walltime: 5 days
- **iw-shared-6**
 - 141 nodes
 - Priority: 0
 - Max walltime: 12 hrs

Account & Queue Info

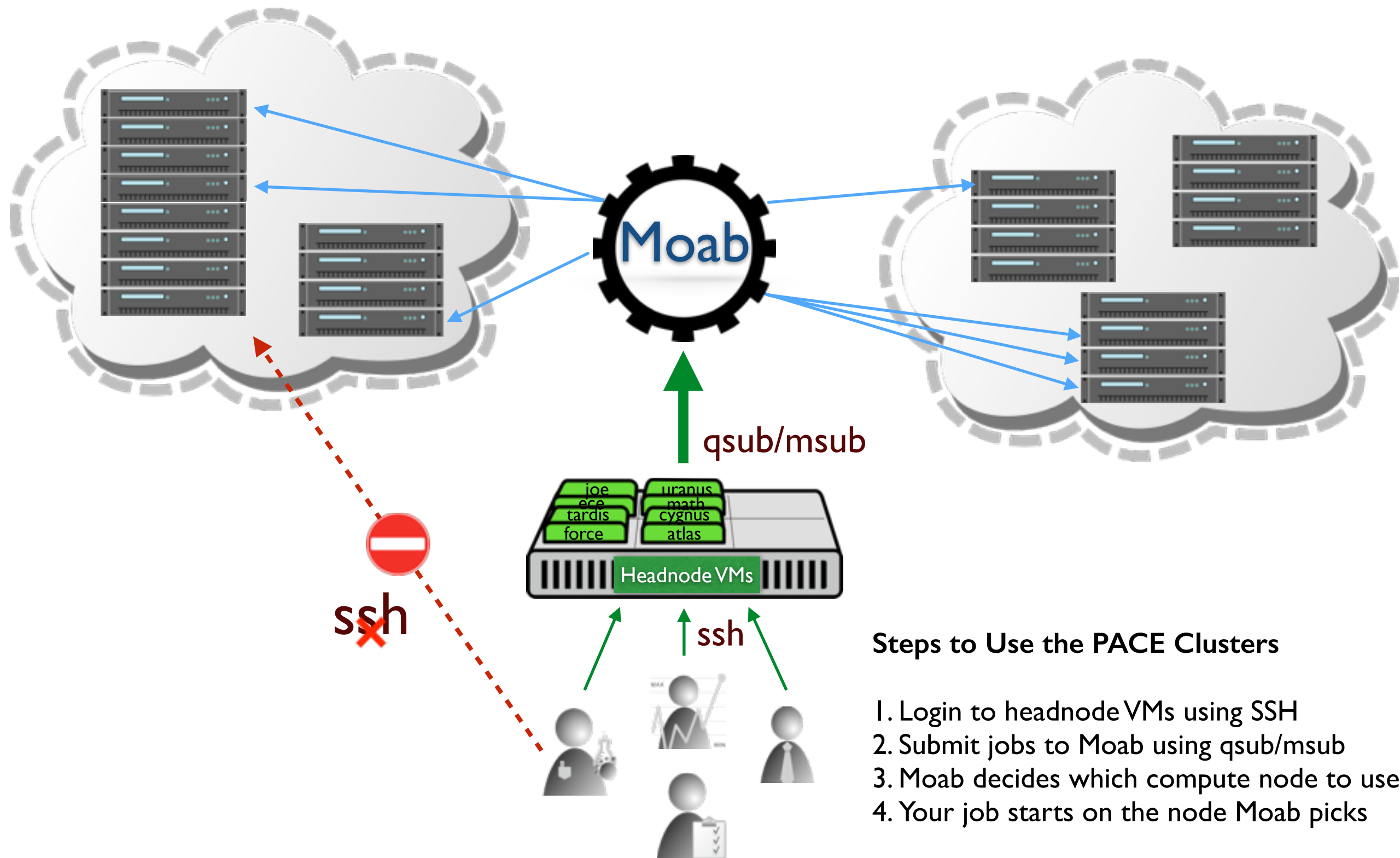
For your account & queue info, run:

`pace-whoami`

```
* Headnode(s) that you can access      :  cygnus-6
* Your queue(s) (could take some time)

  Queue Name      Max. Walltime      Base Priority
  =====
  cygnus-hp-small  14:00:00:00      3100
  cygnus-xl        90:00:00:00      6000
  cygnus           90:00:00:00      6000
  cygnus-hp-lrg-6  10:00:00:00      3100
  iw-shared        12:00:00         100
  cygnus-6         90:00:00:00      6000
  cygnusforce-6    10:00:00:00      2298
  iw-shared-6      12:00:00         100
  testflight       6:00:00         6000
  nvidia-gpu       14:00:00:00      6000
```

Summary



Steps to Use the PACE Clusters

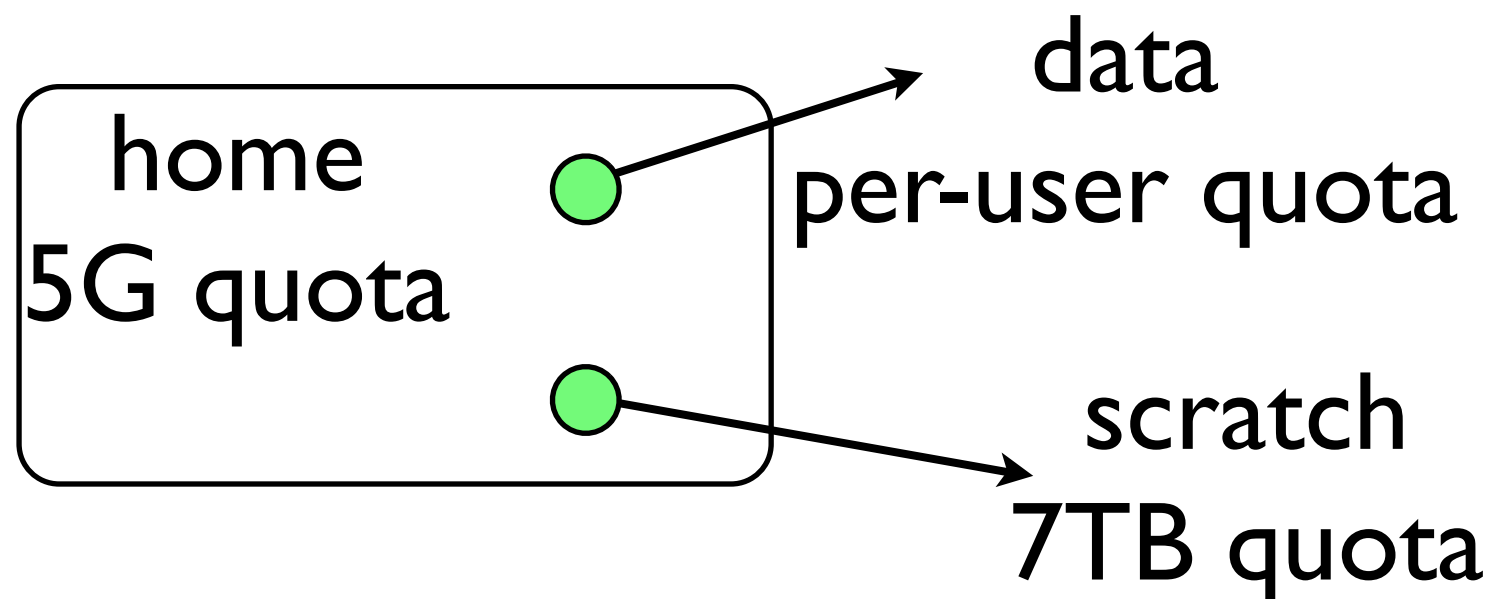
1. Login to headnode VMs using SSH
2. Submit jobs to Moab using `qsub/msub`
3. Moab decides which compute node to use
4. Your job starts on the node Moab picks

Headnodes vs. Compute Nodes

- **Head Nodes: The machines you use to log in**
 - Good for compiling, editing, debugging, etc.
 - Not good for actual computations or visualization!
 - Named like “force.pace.gatech.edu”
- **Compute Nodes: The machines that run all computations**
 - No direct access by users
 - Allocated per-job by the scheduler
 - Named like “iw-h41-l3.pace.gatech.edu”

Storage and Quotas

- Your data are accessible from all nodes (head and compute nodes)
- Three storage directories:
 - home (5GB quota for all users), backed up daily
 - data (Quota depends on department policies), backed up daily
 - scratch (no quota, but files > 60 days are deleted), no backups!
- ATDC users gets a no-quota and fast home directory (no data, scratch)



Storage and Quotas

A (not-so-robust) tool for checking file quota for PACE storage:

pace-quota

```
$ pace-quota
```

```
User      : mbelgin3
```

```
===== Checking 'home' =====
```

```
Quota Status : NOT over quota  
Current Usage : 2953M  
Soft limit   : 5120M  
Hard limit   : 5120M  
Full path    : /nv/hp16/mbelgin3
```

```
===== Checking 'data' =====
```

```
Quota Status : NOT over quota  
Current Usage : 2224G  
Soft limit   : 4000G  
Hard limit   : 4000G  
Full path    : /nv/pf2/mbelgin3
```

```
===== Checking 'scratch' =====
```

```
Quota Status : NOT over quota  
Current Usage : 0.000524288G  
Soft limit   : 5000G  
Hard limit   : 7000G  
Full path    : /panfs/iw-scratch.pace.gatech.edu/v12/mbelgin3
```


Data Transfers in/out

- Datamover machine, with fast connection to all storage servers. It allows for direct login. Always prefer it over headnodes for internally moving data!

`ssh username@iw-dm-4.pace.gatech.edu`

- Mounter applications mount remote storage so you can drag/drop or edit in place as if the files are on your local machine
 - Windows : webdrive (free via software.oit.gatech.edu)
 - OSX : macfusion (open source)
 - Linux : SSHFS, autofs (open source, no GUI)
- Free FTP applications that work with PACE:
 - FileZilla is a free FTP tool for Windows, OSX and Linux.
 - ~~“Cyberduck” a native FTP tool for OSX~~ (no longer free :/)
 - FileZilla is preferred for Linux; gFTP is a good alternative

Running Jobs: Overview

- Users make requests to **Moab scheduler** specifying the requirements of the code:
 - The number of Nodes and/or Cores per node.
 - The total Memory or Memory-per-core.
 - An estimated Runtime (walltime, not CPU time)
 - Specific hardware resources, e.g. GPU

<http://pace.gatech.edu/how-can-i-ask-particular-node-and-or-property>

- Allocated resources can only be used by the user for the duration of requested walltime. This is the only time users can directly login to compute nodes.

Operation Modes

Two modes of operation:

- **Batch:** Submit & forget. Job waits in the queue until resources become available, runs, emails user on exit.
- **Interactive:** Allows interactive use, no different than remotely using any workstation

(required for using GUI, such as MATLAB, R, COMSOL, ANSYS, visualization, etc.)

Job Submission: msub vs. qsub

Command(s) for job submission: `msub` or `qsub`

- `qsub`: faster and more reliable than `msub`, preferred!
- `msub`: was required when `qsub` was buggy, but still available.

There are slight differences between `qsub` and `msub`. See their man pages for more details:

`man qsub`

`man msub`

Submitting Batch Jobs

- Everything needs to be scripted. Not for codes that require user interaction (e.g. press 'y' to continue).
- A 'PBS script' that includes resource requirements, environmental settings, and tasks
- 'qsub' or 'msub' to submit the job

`qsub example_PBS_Script.pbs`

- The output and error logs are printed on files, as they would appear on the screen.

PBS Script Example

comment

This is an example PBS script

#PBS -N hello

#PBS -l nodes=2:ppn=4:nvidiagpu

#PBS -l mem=2gb (or pmem)

#PBS -l walltime=15:00:00

#PBS -q force-6

#PBS -j oe

#PBS -o myjob.out

#PBS -m abe

#PBS -M youremail@gatech.edu

command

cd ~/test_directory

echo "Started on `/bin/hostname`"

module load gcc mvapich2/2.0ga

mpirun -np 8 ./hello

PBS Script Example

```
# This is an example PBS script
```

```
#PBS -N hello
```

 A name for this run, can be anything

```
#PBS -l nodes=2:ppn=4:nvidiagpu
```

```
#PBS -l mem=2gb (or pmem)
```

```
#PBS -l walltime=15:00:00
```

```
#PBS -q force-6
```

```
#PBS -j oe
```

```
#PBS -o myjob.out
```

```
#PBS -m abe
```

```
#PBS -M youremail@gatech.edu
```

```
cd ~/test_directory
```

```
echo "Started on `bin/hostname`"
```

```
module load gcc mvapich2/2.0ga
```

```
mpirun -np 8 ./hello
```

PBS Script Example

```
# This is an example PBS script
```

```
#PBS -N hello
```

```
#PBS -l nodes=2:ppn=4:nvidiagpu
```

 2 “GPU” nodes, 4 cores in each

```
#PBS -l mem=2gb (or pmem)
```

```
#PBS -l walltime=15:00:00
```

```
#PBS -q force-6
```

```
#PBS -j oe
```

```
#PBS -o myjob.out
```

```
#PBS -m abe
```

```
#PBS -M youremail@gatech.edu
```

```
cd ~/test_directory
```

```
echo "Started on `bin/hostname`"
```

```
module load gcc mvapich2/2.0ga
```

```
mpirun -np 8 ./hello
```


PBS Script Example

```
# This is an example PBS script
#PBS -N hello
#PBS -l nodes=2:ppn=4:nvidiagpu
#PBS -l mem=2gb (or pmem) 2 GB "Total" memory requirement
#PBS -l walltime=15:00:00
#PBS -q force-6
#PBS -j oe
#PBS -o myjob.out
#PBS -m abe
#PBS -M youremail@gatech.edu

cd ~/test_directory
echo "Started on `/bin/hostname`"
module load gcc mvapich2/2.0ga
mpirun -np 8 ./hello
```

PBS Script Example

```
# This is an example PBS script
```

```
#PBS -N hello
```

```
#PBS -l nodes=2:ppn=4:nvidiagpu
```

```
#PBS -l mem=2gb (or pmem)
```

```
#PBS -l walltime=15:00:00
```

 15 hrs “max”, after which job is killed!!

```
#PBS -q force-6
```

```
#PBS -j oe
```

```
#PBS -o myjob.out
```

```
#PBS -m abe
```

```
#PBS -M youremail@gatech.edu
```

```
cd ~/test_directory
```

```
echo "Started on ` /bin/hostname `"
```

```
module load gcc mvapich2/2.0ga
```

```
mpirun -np 8 ./hello
```

PBS Script Example

```
# This is an example PBS script
#PBS -N hello
#PBS -l nodes=2:ppn=4:nvidiagpu
#PBS -l mem=2gb (or pmem)
#PBS -l walltime=15:00:00
#PBS -q force-6
#PBS -j oe
#PBS -o myjob.out
#PBS -m abe
#PBS -M youremail@gatech.edu
```

submitting to queue named “force-6”

```
cd ~/test_directory
echo "Started on `bin/hostname`"
module load gcc mvapich2/2.0ga
mpirun -np 8 ./hello
```

PBS Script Example

```
# This is an example PBS script
#PBS -N hello
#PBS -l nodes=2:ppn=4:nvidiagpu
#PBS -l mem=2gb (or pmem)
#PBS -l walltime=15:00:00
#PBS -q force-6
#PBS -j oe
#PBS -o myjob.out
#PBS -m abe
#PBS -M youremail@gatech.edu
```

Put output and error files in specified format

```
cd ~/test_directory
echo "Started on `/bin/hostname`"
module load gcc mvapich2/2.0ga
mpirun -np 8 ./hello
```

PBS Script Example

```
# This is an example PBS script
#PBS -N hello
#PBS -l nodes=2:ppn=4:nvidiagpu
#PBS -l mem=2gb (or pmem)
#PBS -l walltime=15:00:00
#PBS -q force-6
#PBS -j oe
#PBS -o myjob.out
```

```
#PBS -m abe
#PBS -M youremail@gatech.edu
```

Notify on start, finish and error, via email

```
cd ~/test_directory
echo "Started on `bin/hostname`"
module load gcc mvapich2/2.0ga
mpirun -np 8 ./hello
```

PBS Script Example

```
# This is an example PBS script
#PBS -N hello
#PBS -l nodes=2:ppn=4:nvidiagpu
#PBS -l mem=2gb (or pmem)
#PBS -l walltime=15:00:00
#PBS -q force-6
#PBS -j oe
#PBS -o myjob.out
#PBS -m abe
#PBS -M youremail@gatech.edu
```

```
cd ~/test_directory
echo "Started on ` /bin/hostname ` "
module load gcc mvapich2/2.0ga
mpirun -np 8 ./hello
```




Actual Computation

Interactive Jobs

- Same PBS commands, but this time on the command line:

Allows GUI

commas to bind multiple values
for a parameter (-l)

qsub -I  -q force-6 -l nodes=2:ppn=4 , walltime=15:00:00 , mem=2gb

- The scheduler logs the user in a compute node when resources become available.
- Session is terminated when:
 - The user exits
 - The terminal is closed
 - Walltime is exceeded

Monitoring Jobs

qstat lists your queued jobs and their state.

qstat -u <username> -n

Job ID	Username	Queue	Jobname	SessID	NDS	TSK	Memory	Time	S	Time
7693767.shared-sched.p iw-h29-16+iw-h29-12+iw-h29-11	mbelgin3	force-6	Testrun_3	45994	32	64	1900m	120:00:00	R	23:55:47
7693771.shared-sched.p iw-h29-15+iw-h29-7	mbelgin3	cygnus64	Testrun_1	8552	16	32	48gb	120:00:00	R	23:55:17
7693775.shared-sched.p iw-h29-10+iw-h29-7+iw-h29-13+iw-h29-17	mbelgin3	force-6	Testrun_2	64492	16	64	1900m	120:00:00	R	23:51:47
7693778.shared-sched.p iw-h29-10+iw-h29-11+iw-h29-15	mbelgin3	force-6	Testrun_3L	1006	32	64	1900m	120:00:00	R	23:46:00
7693780.shared-sched.p iw-h29-12+iw-h29-10+iw-h29-13+iw-h29-16+iw-h29-15+iw-h29-17+iw-h29-9 +iw-k30-15	mbelgin3	force-6	Testrun_3L	13369	32	128	1900m	120:00:00	R	23:45:09
7695869.shared-sched.p iw-h29-16+iw-h29-11+iw-h29-7+iw-h29-13+iw-h29-8+iw-h29-9+iw-k30-21+iw-k30-22 +iw-k30-19+iw-k30-17+iw-k30-16+iw-h31-19	mbelgin3	force-6	L6.246full	38241	16	128	1900m	120:00:00	R	09:17:47

Monitoring Jobs (cont.)

“checkjob” tells all you need to know about a job.

```
$ checkjob -v 532356
```

```
job 532356 (RM job '532356.repace.pace.gatech.edu')
```

```
AName: STDIN
```

```
State: Removed
```

```
Complete Time: Thu Aug 18 16:10:52
```

```
Completion Code: 0
```

```
Creds: user:mbelgin3 group:pace-admins class:iw-shared
```

```
WallTime: 00:00:00 of 1:00:00:00
```

```
SubmitTime: Thu Aug 18 16:10:26
```

```
(Time Queued Total: 00:00:50 Eligible: 00:00:00)
```

```
Total Requested Tasks: 40
```

```
Req[0] TaskCount: 40 Partition: ALL
```

```
Memory >= 0 Disk >= 0 Swap >= 0
```

```
Opsys: --- Arch: --- Features: ---
```

```
Dedicated Resources Per Task: PROCS: 1 MEM: 2048M
```

```
NodeAccess: SHARED
```

```
TasksPerNode: 8
```

```
UMask: 0000
```

```
OutputFile: repace.pace.gatech.edu:/nv/hp16/mbelgin3/STDIN.o532356
```

```
ErrorFile: repace.pace.gatech.edu:/nv/hp16/mbelgin3/STDIN.o532356
```

```
Execution Partition: ALL
```

```
SrcRM: repace DstRM: repace DstRMJID: 532356.repace.pace.gatech.edu
```

```
StartPriority: 0
```

```
PE: 40.00
```

```
Message[0] job cancelled - MOAB_INFO: job was rejected - job violates class configuration 'wclimit too high for class 'iw-shared' (86400 > 43200)'
```

More on PBS Jobs

- Cancelling a submitted job:

`qdel <jobID>` (prefer if submitted using `qsub`)

`canceljob <JobID>` (prefer if submitted using `msub`)

- Querying for specific users/queues

`showq -w class=<QueueName>,user=<UserID>` => All jobs for the queue & user

`showq -r -w user=<UserID>` => All “running” jobs for the given user

`showq -b -w class=<UserID>` => All “blocked” jobs for the given queue

Checking the queue status

pace-check-queue <queueName>

```
$pace-check-queue cygnus64-6
```

```
=== cygnus64-6 Queue Summary: ===
```

```
Last Update           : 04/30/2015 14:45:01
Number of Nodes (Accepting Jobs/Total) : 18/23 (78.26%)
Number of Cores (Used/Total)           : 591/1472 (40.15%)
Amount of Memory (Used/Total) (MB)     : 153364/4335453 ( 3.54%)
```

```
=====
Hostname      tasks/np  Cpu%   loadav%  used/totmem(MB)  Mem%  Accepting Jobs?
=====
```

iw-h29-10	0/64	0.0	0.3	3303/133939	2.5	No (ERROR Health check failed: [pace_basic] high disk usage)
iw-h29-11	0/64	0.0	0.3	3247/133939	2.4	No (ERROR Health check failed: [pace_basic] high disk usage)
iw-h29-12	26/64	40.6	84.5	4039/133939	3.0	Yes (free)
iw-h29-13	0/64	0.0	0.2	3332/133939	2.5	No (ERROR Health check failed: [pace_basic] high disk usage)
iw-h29-14	21/64	32.8	79.0	3947/133939	2.9	Yes (job-exclusive)
iw-h29-15	33/64	51.6	90.8	7702/133939	5.8	Yes (job-exclusive)
iw-h29-16	28/64	43.8	91.0	11390/133939	8.5	Yes (job-exclusive)
iw-h29-17	20/64	31.2	95.2	5117/133939	3.8	Yes (job-exclusive)

PACE Software Stack

Everything is in “/usr/local/pacerepov1”

- Licensed software packages:
 - Common license: Matlab, Fluent, Mathematica, Abaqus, Comsol...
 - Individual license: Vasp, Gaussian, ...
- Open source packages and HPC libraries:
 - BLAS, PETSc, NAMD, NetCDF, FFTW, BLAST, LAMMPS...
- Compilers:
 - C/C++ & Fortran: GNU, Intel, PGI, NAG
 - Parallel Compilers: OpenMP, MPICH, MPICH2, MVAPICH
 - GPU compilers: CUDA, PGI
- Scripting Languages: Python, Perl, R, ...

Modules (RHEL6 only)

- Painless configuration for software environment and switching between different versions:
No more editing PATH, LD_LIBRARY_PATH, etc!
- Main Commands:

- ▶ `module avail` : Lists all available modules that can be loaded
- ▶ `module list` : Displays all the modules that are currently loaded
- ▶ `module load` : Loads a module to the environment.
- ▶ `module rm` : Removes a module from the environment
- ▶ `module purge` : Removes all loaded modules (buggy)

- Modules may depend on, or conflict with, each other

```
$ module load matlab/r2009b
```

```
matlab/r2009b(7):ERROR:150: Module 'matlab/r2009b' conflicts with the currently loaded module(s) 'matlab/r2010a'
```

- **Must-Read:** PACE-specific use cases, examples, and gotchas

<http://www.pace.gatech.edu/using-software-modules>

Loading Modules at login

- Automated way to load modules when you log in
- Edit this file (create if does not exist)

to edit : “ nano ~/.pacemodules ”

- Example:

```
module load gcc/4.9.0
module load mvapich2/2.1
...
```

- All of the modules you will include in this file will be loaded at login
- Pay attention to dependencies and conflicts between modules. Test the modules in the same order before putting them in ~/.pacemodules
- This is NOT mandatory. Some users prefer not having a default set. In that case, make sure you include module load commands in your PBS scripts!

PACE Email Lists & Blog

- **User Lists:**

- All users:

- pace-availability (non-optional subscription)
 - pace-discuss (optional unsubscription)

- Cluster-specific lists: (non-optional subscription)

- pace-biocluster
 - pace-atlantis
 - pace-joe
 - ...

- **PACE blog:**

<http://blog.pace.gatech.edu/>

THANK YOU!

Slides available at:

http://www.pace.gatech.edu/orientation/PACE_Orientation.pdf

(This link is also available at the PACE Orientation Website:

<http://www.pace.gatech.edu/content/orientation>)