# WELCOME TO CLUSTERS 101 WORKSHOP

**Presenters:**

***Gladys Andino***

***Dan Dietz***

***Jieyu Gao***

***Steve Kelley***

***Xiao Zhu***

**Workshop material:**

https://www.rcac.purdue.edu/tutorials/clusters101

ITaP

PURDUE UNIVERSITY

# ACKNOWLEDGEMENTS


- The material in this workshop was prepared by the Purdue University ITaP Research Computing team.

- Special thanks to Dr. ***Boyu Zhang*** and ***Lev Gorenstein*** for their collaboration and slides preparation.

# VOCABULARY

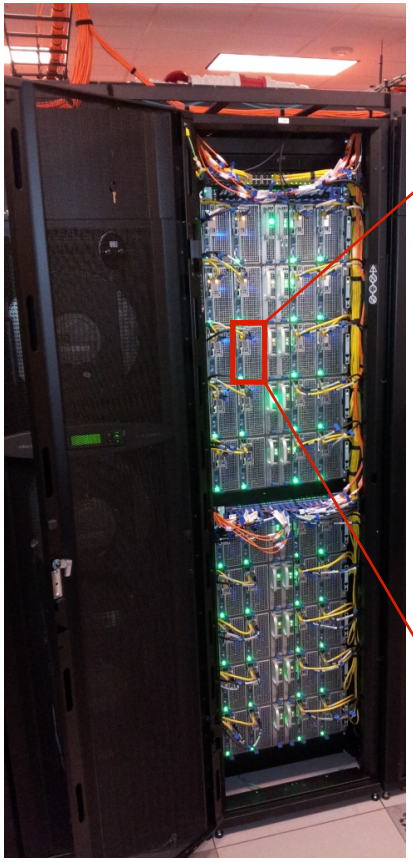
## A CLUSTER IS A CLUSTER IS A CLUSTER?

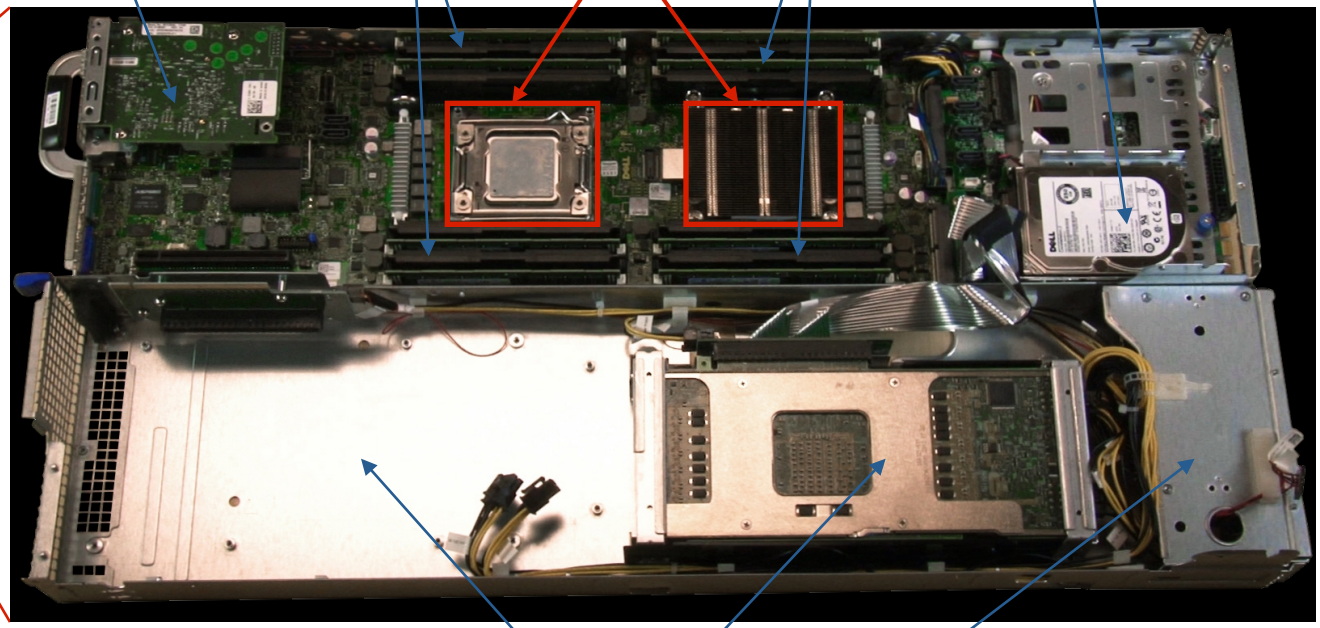
- **Cluster**:
  Hardware (compute nodes + interconnect + storage)
  + Software (OS + compilers + libraries + apps + queue manager)
  + Infrastructure (front-ends + power + cooling + data center + staff)

# VOCABULARY

## NODES

- **<u>Node</u>:** a computer

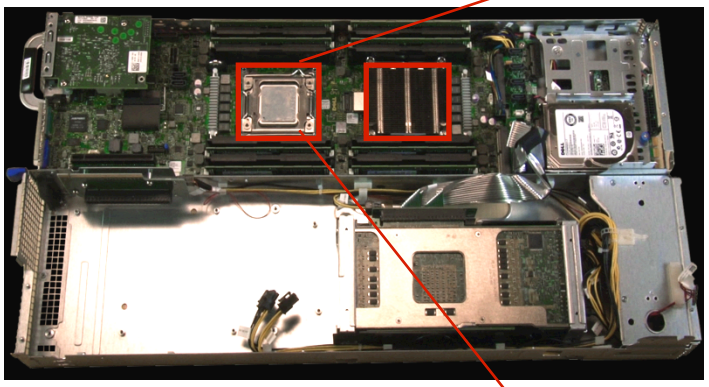


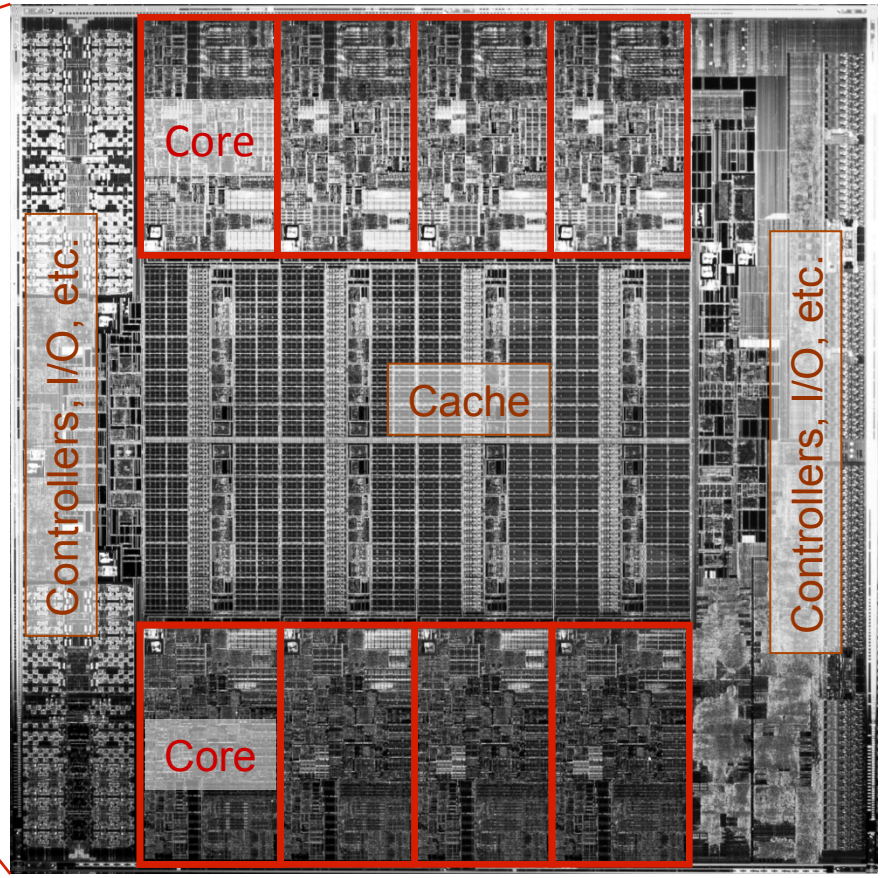


# VOCABULARY

## CPUS

- **<u>Processor</u>:** a.k.a chip, a.k.a. physical CPU





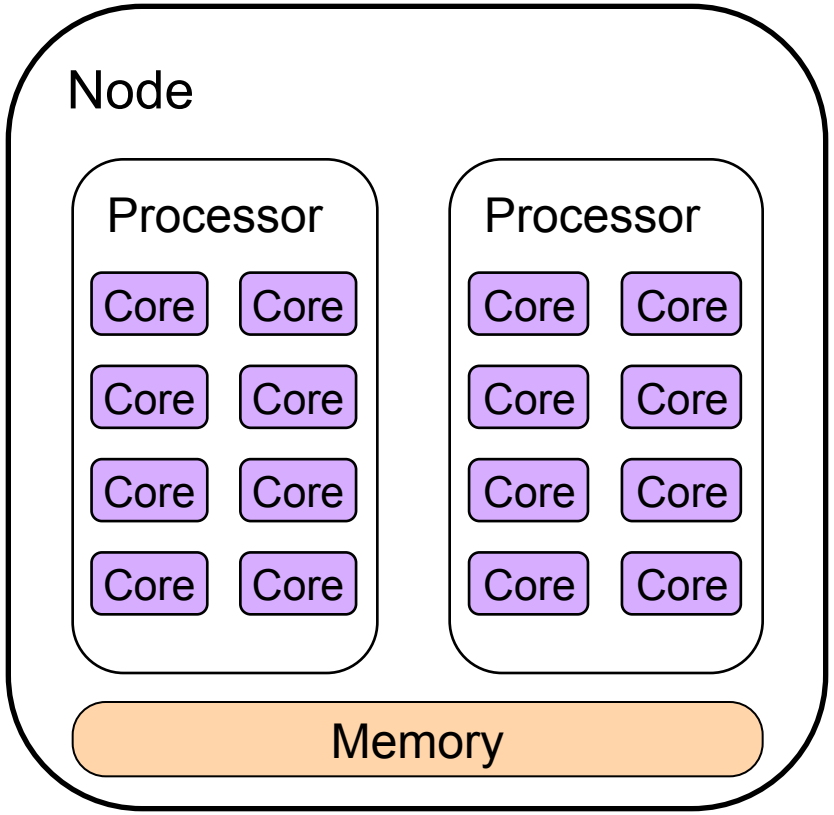
# VOCABULARY

## CORES

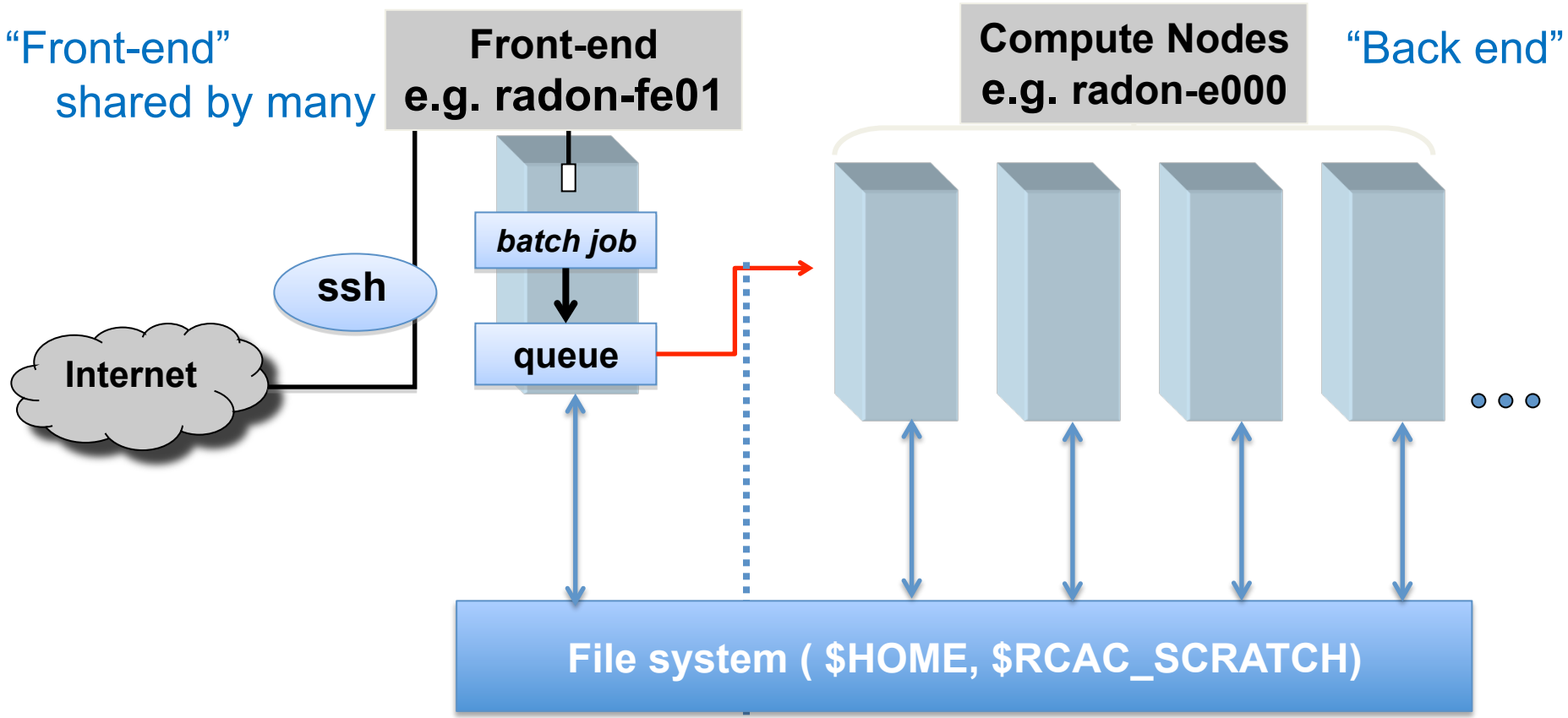
- **<u>Processor Core</u>:** individual compute unit ("slot") on the chip

- You see:
  2 physical processors, 8 cores ea.
- Queuing system sees:
  16 logical processors
  (hence 'nodes=X:ppn=16' effectively stands for "*processor cores per node*").

- From now on, we will be mostly concerned with cores (logical processors), not physical chips
  ("I ran on 5 CPUs" == "on 5 processors" == "on 5 cores")



# FRONT-END VS COMPUTE NODE




Running Jobs: The goal is getting to the compute nodes

# VOCABULARY

## IBM SYSTEM 360 (1 COMPUTER, 1 PROCESSOR)

# TYPES OF NODES

## COMMUNITY CLUSTERS

**HPC (Rice):** Multiple cores or nodes, probably MPI. Benefit from high-performance network and parallel filesystem. The vast majority of campus - 80% of all work!

**Data-Intensive Life Science (Snyder):** Use entire node to get large amounts of memory. Less need for high-performance network. Needs large, fast storage.

# USER ENVIRONMENT

# USER ENVIRONMENT

## HELP WITH THE ENVIRONMENT AT ITAP RESEARCH COMPUTING

- Read the User Guide

Radon: https://www.rcac.purdue.edu/compute/radon/guide/

Rice: https://www.rcac.purdue.edu/compute/rice/guide/

Conte: https://www.rcac.purdue.edu/compute/conte/guide/

Carter: https://www.rcac.purdue.edu/compute/carter/guide/

- Submit a ticket (rcac-help@purdue.edu)

# USER ENVIRONMENT

## HELP WITH THE ENVIRONMENT AT ITAP RESEARCH COMPUTING

Home | News & Events | User Info | Accounts | Computation | Storage | Visualization | Purchase | Services | About


**Rice Info**

- About Rice
- Quick Reference Card
- User Guide
- Frequently Asked Questions
- John R. Rice Bio

### Rice – User Guide

Expand All





ITaP

PURDUE UNIVERSITY

# LOGGING IN

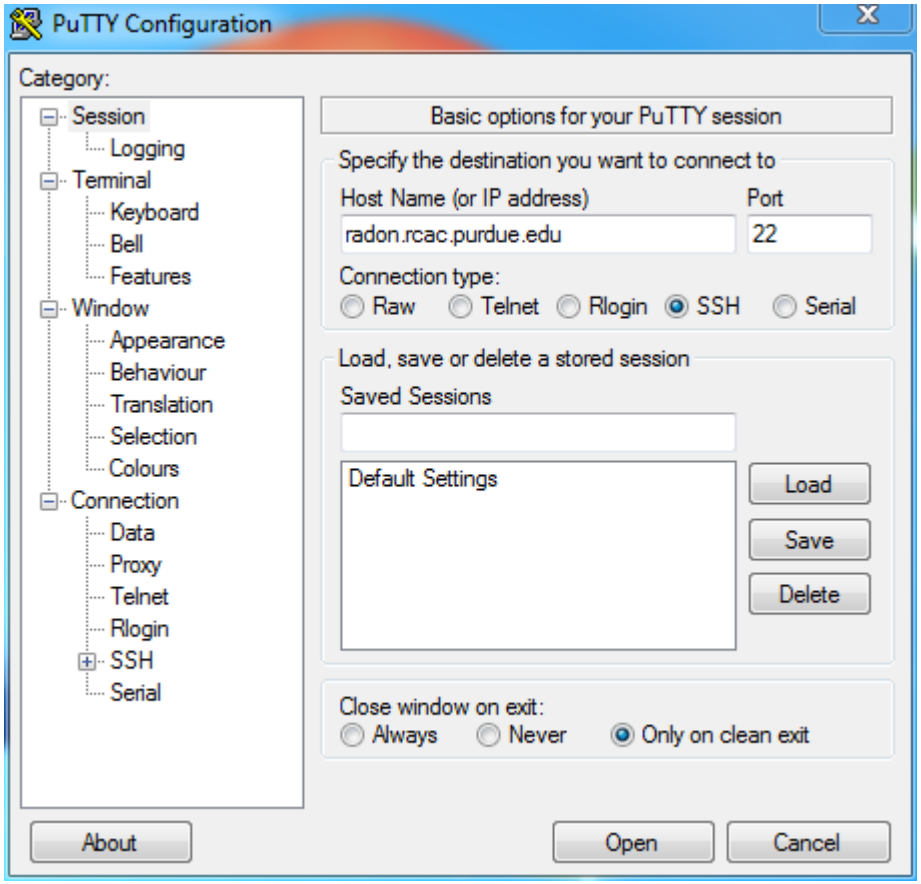
## WINDOWS

- Many clients are available for Windows
- We will use the PuTTY SSH client
- Download PuTTY, no install required
- http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html
  (or Google search “putty”)
- Download “putty.exe” for Intel x86 to your desktop

# LOGGING IN

## WINDOWS

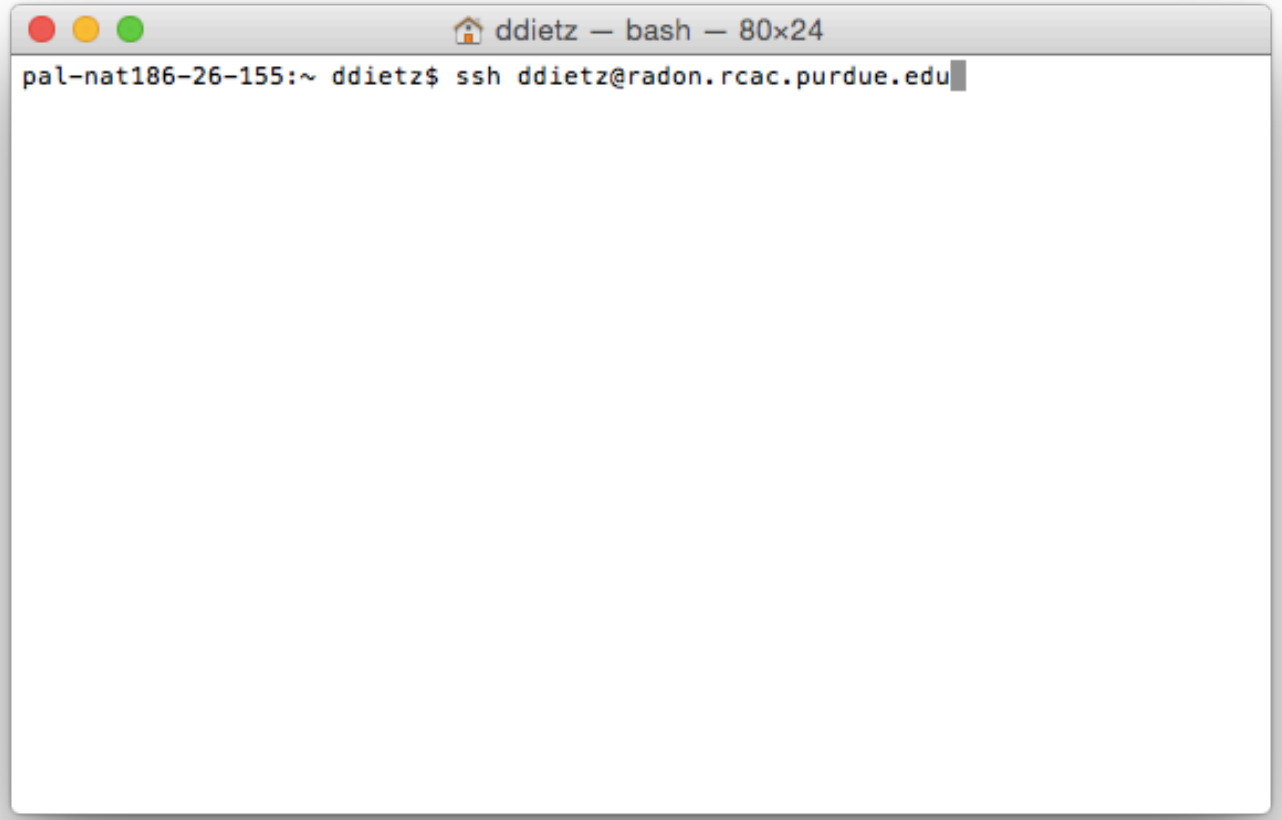
- Host Name for Radon is “radon.rcac.purdue.edu”



# LOGGING IN

## MAC

- Connect using `ssh username@radon.rcac.purdue.edu`



# MODULES

## WHAT IS A MODULE SYSTEM?

- All ITaP Research Computing clusters use a module system
- Module system provides for the dynamic modification of a user's environment
- Environment can be modified on a per-module basis using the module commands
- Set up the environment for running applications using one or two simple commands

```
$ module load bioinfo

$ module load bowtie
```

# MODULES

## RECOMMENDED ENVIRONMENT

- There are also modules for setting up one of several different programming environments. Most users will not need to change their programming environment
- Research Computing recommends a set of compilers, math libraries, and message-passing libraries for general optimal performance
- Recommended environment on each cluster contains different modules

```
$ module load devel
```

# MODULES

## KEY MODULE COMMANDS

**$ module avail**

Running the command on the node will give you the most up to date list. The full list contains modules for compilers, MPI libraries, user applications and utilities.

**$ module avail python**

"python" restricts the listing to the modules whose name starts with *python*.

```
$ module avail python

---------------------------- /opt/modules/modulefiles ----------------------------
python/2.6.5                    python/2.7.8_intel-15.0.2.164
python/2.7.2                    python/2.7.8_intel-15.0.3.187
python/2.7.5_intel-13.1.1.163 python/3.4.1
python/2.7.6_intel-14.0.2.144 python/anaconda(default)
python/2.7.8_intel-15.0.1.133
```

# MODULES

## KEY MODULE COMMANDS

```
$ module list                          # list loaded modules
$ module load <module>                 # add a module
$ module unload <module>               # remove a module
$ module purge                         # unload all modules
```

```
$ module list
No Modulefiles Currently Loaded.

$ module load devel
$ module load matlab/R2013a

$ module list
Currently Loaded Modulefiles:
1) intel/13.1.1.163   2) openmpi/1.6.3_intel-13.1.1.163   3) devel/20130708   4) matlab/R2013a

$ module unload matlab
$ module list
Currently Loaded Modulefiles:
1) intel/13.1.1.163   2) openmpi/1.6.3_intel-13.1.1.163   3) devel/20130708

$ module purge
$ module list
No Modulefiles Currently Loaded.
```

# MODULES

## WHAT DOES "MODULE LOAD" DO?

By loading a module, you will have just changed the environmental variables defined for that module. Typically this is $PATH, $MANPATH, and $LD_LIBRARY_PATH. Also a few extra module-specific environmental variables could be set.

```
$ module load vmd
```

*loads its module and adds to PATH and LD_LIBRARY_PATH*

```
PATH=/apps/rhel6/vmd-1.9.1/bin:$PATH
LD_LIBRARY_PATH=/apps/rhel6/vmd-1.9.1/lib:$LD_LIBRARY_PATH
```

*and sets*

```
VMD_HOME=/apps/rhel6/vmd-1.9
LIBGL_ALWAYS_INDIRECT=y
```

```
$ ls -l $VMD_HOME/bin
total 182K
-rwxr-xr-x 1 vvergara 12K Jul 27  2012 vmd
-rw-r--r-- 1 vvergara 12K Feb  1  2012 vmd.csh
-rwxr-xr-x 1 vvergara 12K Feb  1  2012 vmd.sh
```

```
$ module show <module>   # show the commands in the module file
```

# MODULES

## QUICK SUMMARY OF MODULE COMMAND

```
$ module help                    #lists options
$ module load <module>           #add a module
$ module avail                   #lists available modules
$ module unload <module>         #remove a module
$ module swap <mod1> <mod2>      #swap two modules
$ module purge                   #remove all modules
$ module show <module>           #show module commands
$ module help <module>           #module-specific help
```

# MODULES

## TRANSITION TO LMOD

- We recently switched to Lmod on our community clusters
- Lmod provides all of the functionality that the current module system does
- Supports software hierarchy
- Has nice features like “spider” and “save”

```
$ module spider                    # {lists all modules}
$ module spider petsc              {list all versions of petsc}
$ module save                      {save personal environment}

$ ml                               {abbrev for module list}
$ ml <module>                      {abbrev for module load}
$ ml -<module>                     {return to system defaults}
$ module reset                     {return to system defaults}
```



ITaP

PURDUE UNIVERSITY

# MODULES

## TIPS FROM US

- Not load in .bashrc/.profile/.login/.cshrc
- Not load more than needed for current task
- Do '`module purge`' and reload when in doubt
- Check with '`module show` …' what a given module does
- Check with '`module list`' what's actually loaded
- Script and automate your builds – even if it's just a 3-line snippet!

# LAB 1 – USING MODULES

- List current loaded modules

```
$ module list
```

- Loaded recommended modules on the cluster

```
$ module load devel
```

- List available modules

```
$ module avail
```

- Search available modules

```
$ module avail xmgrace
$ module avail x
```

- List the help information of one module

```
$ module help vmd  #or any modules you are interested
```

- Show detailed of one module

```
$ module show vmd
```

- Load extra modules as you like
- Remove all loaded modules and reset environment

```
$ module purge
```

# LAB 1 – USING MODULES

- Find if module `cufflinks` available on Radon.
- Search all modules containing `"mpi"` in their name.
- Search all modules that end with `"mpi"` in their name.
- Load module `trinity/2.2.0`
- Please finish the following sequences of module operations:
  1. clear all modules
  2. load default version `matlab`, `python/2.7.8_intel-16.0.2.181`, `r/3.1.0` and `spark/2.0.0`
  3. save the current module setting in the name of `datajob`
  4. clear all modules and restore the all modules saved in `datajob`

# TYPES OF WORK

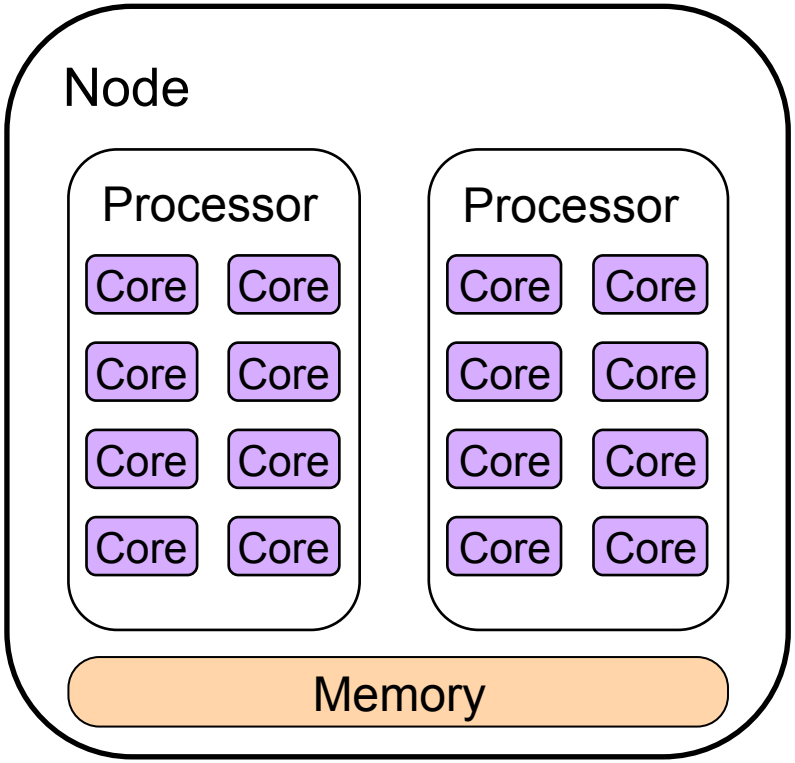
## SERIAL VS PARALLEL

More than one mower!

Mow the lawn faster …

… or mow a bigger lawn

**Lawn mowers = cores**






ITaP

PURDUE UNIVERSITY

# TYPES OF WORK

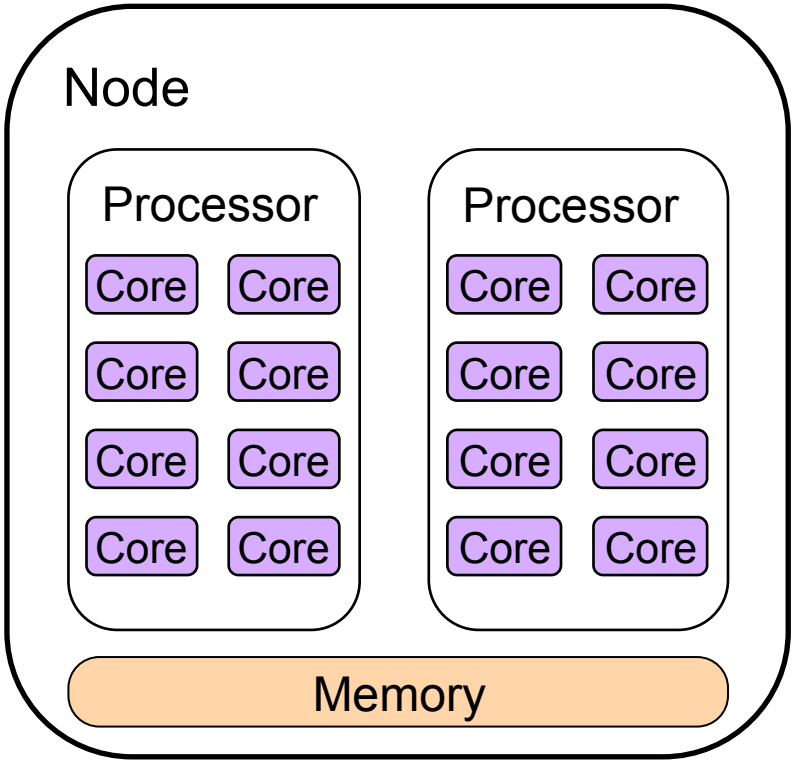
## SERIAL VS PARALLEL

More than one mower!

Mow the lawn faster …

… or mow a bigger lawn

**Lawn mowers = cores**





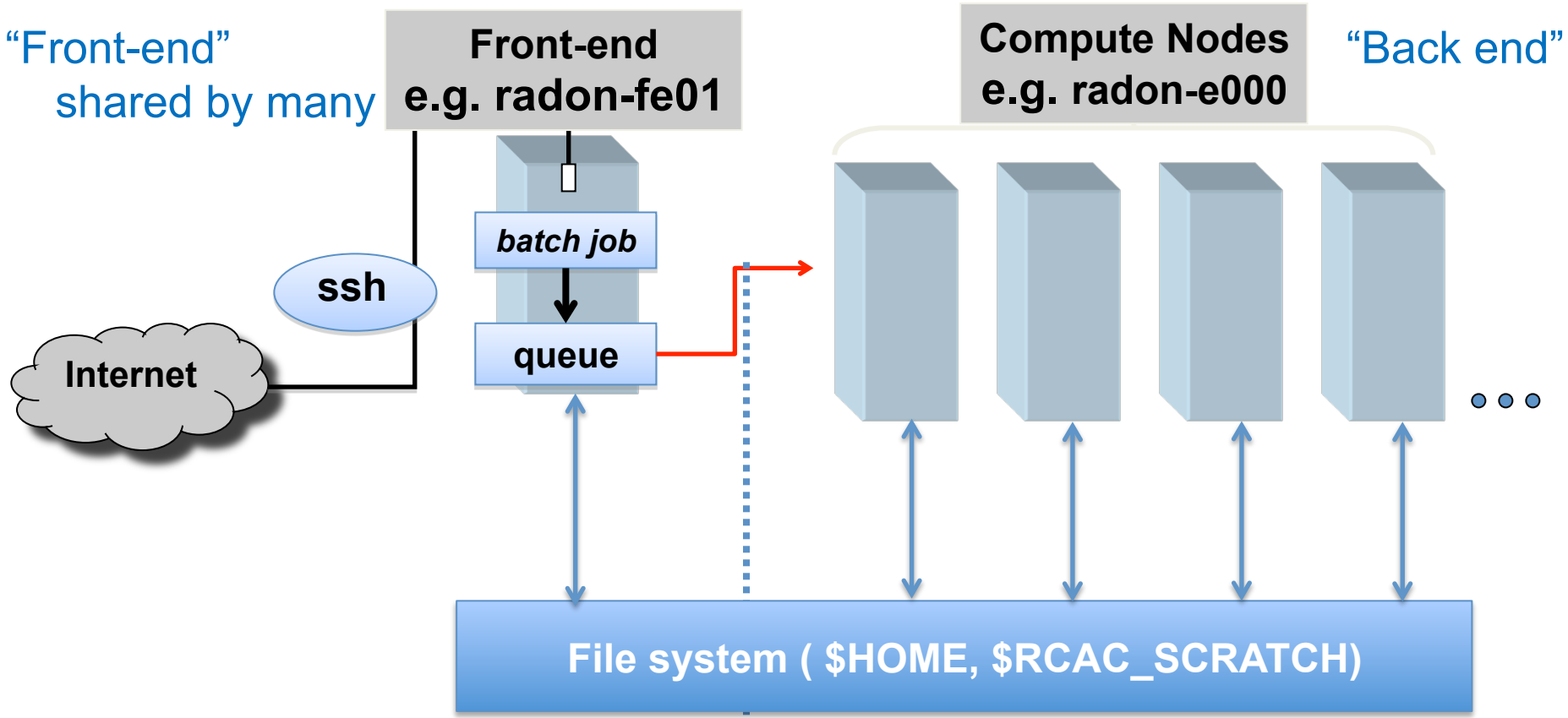

ITaP

PURDUE UNIVERSITY

# PARALLEL COMPUTATIONS

- MPI – Message Passing Interface, multiple nodes
- OpenMP – confine yourself to one node
- Hybrid – mix MPI and OpenMP
- *Accelerators/coprocessors – GPU, Intel Xeon Phi*
- *Parallel R, Matlab, Hadoop & Spark*

# FRONT-END VS COMPUTE NODE




Running Jobs: The goal is getting to the compute nodes

# FRONT-ENDS: INAPPROPRIATE USE

- What does this mean?
- Do not do science on the front end
  - Either it's simply not possible
  - Or you'll annoy the system administrators and other users
- Instead: submit a batch job. We'll get to that.

# FRONT-ENDS: APPROPRIATE USE

- Building software
  - are compilers also visible on compute nodes?
- Managing files
  - editing, transfers, `tar`, `gzip`, `hsi`
- Submitting, monitoring, managing batch jobs
- Launching interactive sessions
- Modest post-processing

# THE KEYS TO GOOD CITIZENSHIP

- Remember you are sharing resources
  - login nodes, file systems, bandwidth
- Use components for intended purposes

# JOBS, QUEUES, AND PBS

# JOBS, QUEUES, AND PBS

## WHAT IS A JOB?

- A job is simply a set of tasks to be performed by a cluster
- Instructs cluster precisely what to do to complete your work
- Self contained to be executed without any interaction
- Submit and forget!

# JOBS, QUEUES, AND PBS

## WHERE NOT TO RUN A JOB

- Remember, cluster front-end nodes are shared resources for
  - Creating, submitting, and monitoring jobs
  - File transfers
  - Preparing inputs
  - Editing and compiling code
  - Small-scale testing
- May be used by 50+ people simultaneously
  - Check out the **`who`** command
  - Jobs should have no interference by other people
  - Want jobs carefully arranged on compute nodes

https://en.wikipedia.org/wiki/Rapid_transit#/media/File:Moscow_MetroCrowded_(pixinn.net).jpg

# JOBS, QUEUES, AND PBS

## WHERE TO RUN A JOB

- Jobs are submitted to the cluster
- Cluster executes jobs on back-end compute nodes
- Jobs are carefully scheduled and arranged on the compute nodes

https://en.wikipedia.org/wiki/SNCF_TGV_Duplex#/media/File:TGV_Dupex_First_Class.jpg

# JOBS, QUEUES, AND PBS

## RESOURCE ALLOCATION

- Jobs need to specify the resources they require
  - Three basic units:
    - Number of nodes
    - Number of cores
    - Time
  - Memory
  - Other resources
- Cluster will allocate requested resources once they are available
- Job starts once resources are allocated

# JOBS, QUEUES, AND PBS

## RESOURCE ALLOCATION – NODES & CORES

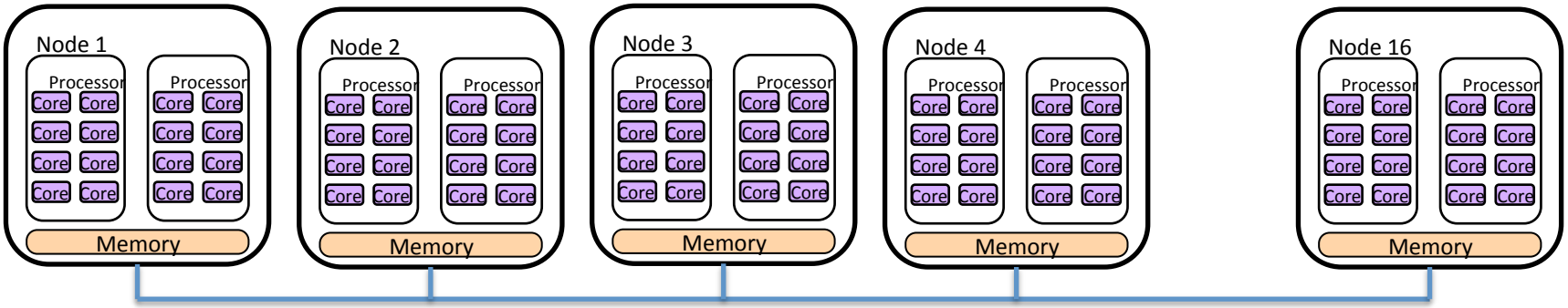
- The most basic unit you can request is nodes
  - Serial applications probably only need one node
  - Parallel applications may need 10s or 100s of nodes

- Your job should also request how many cores per node it needs
  - Cluster allows your job to use only the cores you request
  - Often your job should request all of the cores in a node
  - Remember, we want to avoid other people!

# JOBS, QUEUES, AND PBS

## RESOURCE ALLOCATION – NODES & CORES

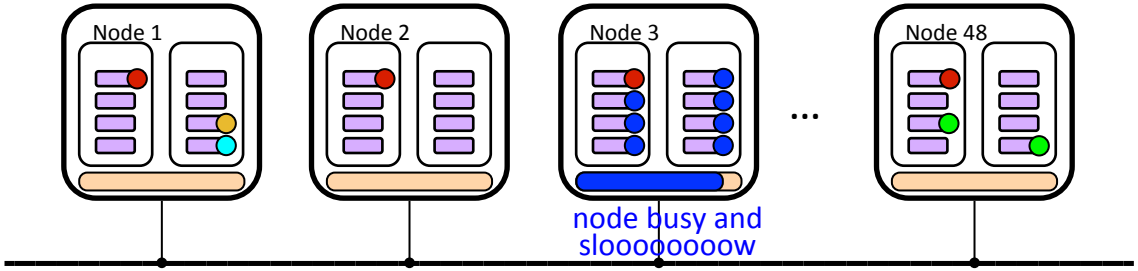
- Nodes are physically different pieces of hardware
- A job with 16 nodes and 16 cores each, is **not** 256 threads
  - 16 individual computers!
  - Program needs to be written with MPI to tie all the nodes together



# JOBS, QUEUES, AND PBS

## RESOURCE ALLOCATION – NODES & CORES

- Nature of the job: how much inter-process communications?
- 1 core on 48 nodes, or for 48 cores on 1 node?
  - Communication inside a node is faster than between nodes
  - Collisions with other people's jobs (48 nodes x 47 jobs = 2256 jobs)
  - Most performant, most predictable, least vulnerable, least interfering



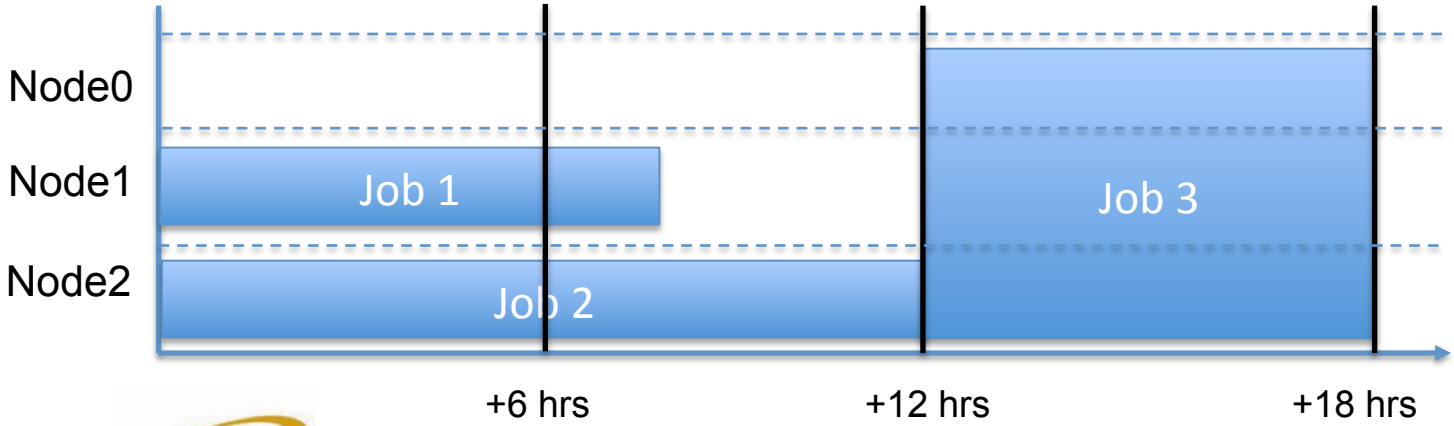

- Tighter packing is almost always preferred
- Newer ITaP clusters enforce exclusive nodes unless explicitly specified for single node jobs

# JOBS, QUEUES, AND PBS

## RESOURCE ALLOCATION – TIME

- The third basic unit jobs can request is walltime
- The cluster won't let your job run forever!
  - Cluster needs to have a time limit for your job
- Helps the cluster efficiently schedule you on the nodes
- Make request accurate with some safety buffer
  - Tune your walltime request through testing and experimenting

https://en.wikipedia.org/wiki/Hourglass#/media/File:Wooden_hourglass_3.jpg



## RESOURCE ALLOCATION – MEMORY

- Jobs need memory, but…
- Clusters can **not** limit your job to an amount memory
  - All of a compute node's memory is available to your job
- All the memory is available to any other jobs on the node!
  - 1 core job needs 16 GB x 16 jobs = 256 GB
  - But wait, my node only has 32 GB!!
    - Jobs crash and burn

# JOBS, QUEUES, AND PBS

## RESOURCE ALLOCATION – MEMORY

- Memory across multiple compute nodes is distributed
- Memory is discrete chunks of memory in different address spaces
- Can't just "add more nodes" to get more memory
- 16 nodes with 16GB memory each is **not** 256 GB of memory
  - 16 discrete chunks of 16GB memory
  - Can't malloc() 256GB of memory
  - Need MPI to tie memory together

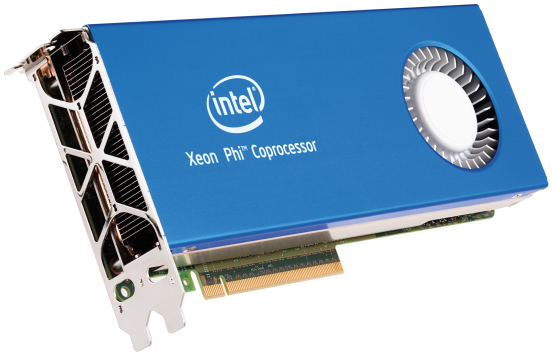
## RESOURCE ALLOCATION – MEMORY

- Best bet is to always request all cores of a node
- Get full control of the node and it's memory
- You can try requesting proportionate numbers of cores but **not** a guarantee!!
- You can always run multiple instances in a single job
  - Maximize utilization of the node
  - Only have to worry about yourself

# JOBS, QUEUES, AND PBS

## RESOURCE ALLOCATION – OTHER RESOURCES

- There are a number of other resources you can request in a job
  - Accelerators
  - Specific types of nodes
  - Certain licenses
    - MATLAB & Toolboxes




# JOBS, QUEUES, AND PBS

## WHAT IS A QUEUE

- Cluster can't run all jobs at once so sometimes you must wait
- Jobs are submitted into queues
- Jobs wait until cluster and queue has free resources
- Queues set constraints on jobs that can be submitted
  - Max nodes
  - Max walltime
  - Sets initial priorities
- Our community clusters have several

https://en.wikipedia.org/wiki/Queue_area#/media/File:Waiting_in_line_at_a_food_store.JPG

# JOBS, QUEUES, AND PBS

## TYPES OF QUEUES

- Owner queues
  - Named for your PI, lab, or group
  - Number of nodes set by what your PI buys into
  - Allows for longest jobs - 336 hours
  - Jobs should start in 4 hours or less
    (assuming your queue isn't full of other jobs in queue)
- Not tied to specific nodes in the cluster
  - Allow you to use the number of nodes somewhere in the cluster
  - Whichever nodes are free, functioning, and type queue can access

# JOBS, QUEUES, AND PBS

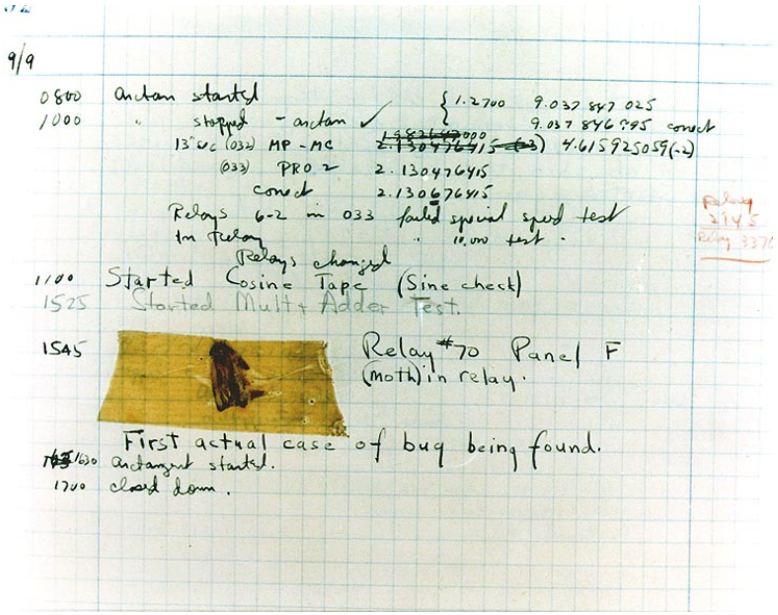
## TYPES OF QUEUES

- Standby queue
  - Uses idle nodes from owner queues
  - Everyone on cluster gets access
  - Limited to 4 hours
    - Owners must be able get quick access from their queue
  - No limit on nodes
    - Probably won't get whole cluster though
  - Lowest priority jobs, no promises on turnaround time
    - Can be minutes or days
  - If you can run under 4 hours, go for it!

# JOBS, QUEUES, AND PBS

## TYPES OF QUEUES

- Debug queue
  - Allows for running small jobs for testing and debugging
  - Up to 2 nodes for 30 minutes at a time
  - Highest priority
  - Starting time <1 minute



# JOBS, QUEUES, AND PBS

## TYPES OF QUEUES

- In this workshop we are using Radon, a general access cluster
- Has none of the above queues!
- workq
  - Sort of like standby
    - Not competing with owner queues
    - Longer jobs
- For this workshop we have set up a special workshop queue
  - Jobs given priority and have several nodes reserved

# JOBS, QUEUES, AND PBS

## WHAT IS PBS?

- Portable Batch System
- Original developed for NASA in 1991
- Open source fork and commercial fork remain
- We use the open source fork, TORQUE

https://en.wikipedia.org/wiki/Portable_Batch_System

# JOBS, QUEUES, AND PBS

## WHAT DOES PBS DO?

- Resource manager
  - Jobs
    - Job submission
    - Job status
    - Executes jobs
  - Queues
  - Nodes

# JOBS, QUEUES, AND PBS

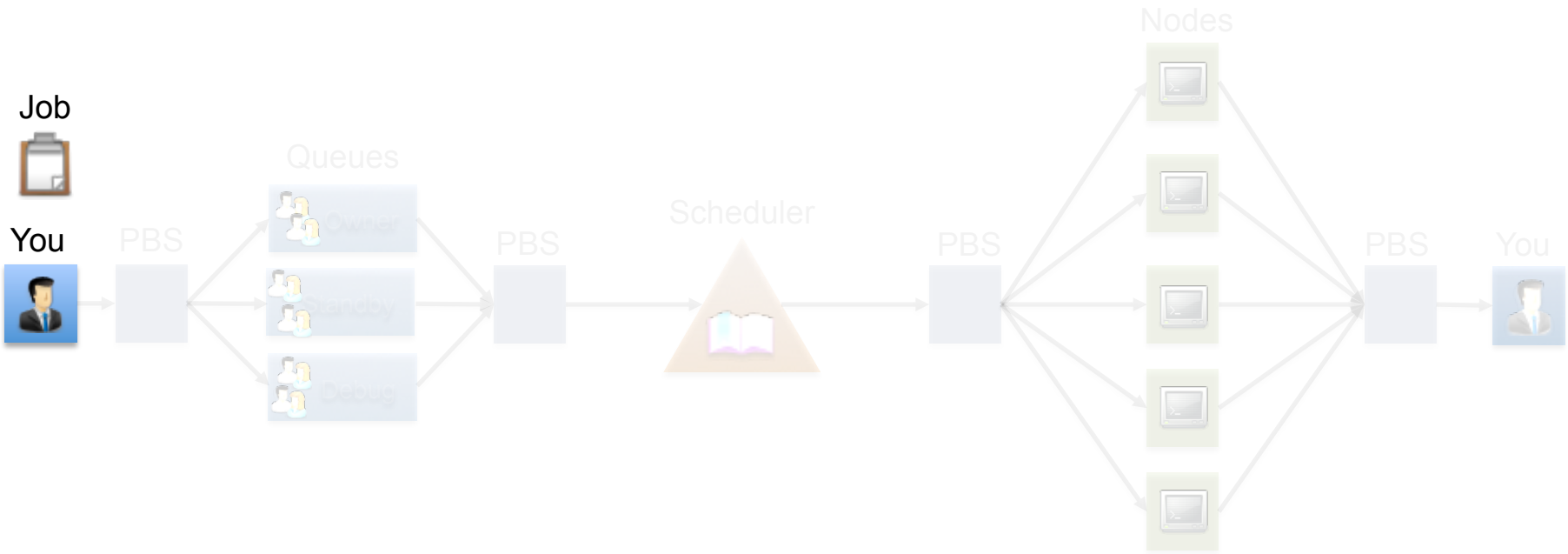
## SCHEDULER

- Scheduler is actually separate software
- We use a scheduler called Moab
- Decides which jobs to run and where
- Typically “PBS” == “Scheduler” == “Batch system” == “Server”
  - We often interchange them in conversation
  - Referring to the whole system of servers, schedulers, managers

# JOBS, QUEUES, AND PBS

## LIFE AS A JOB

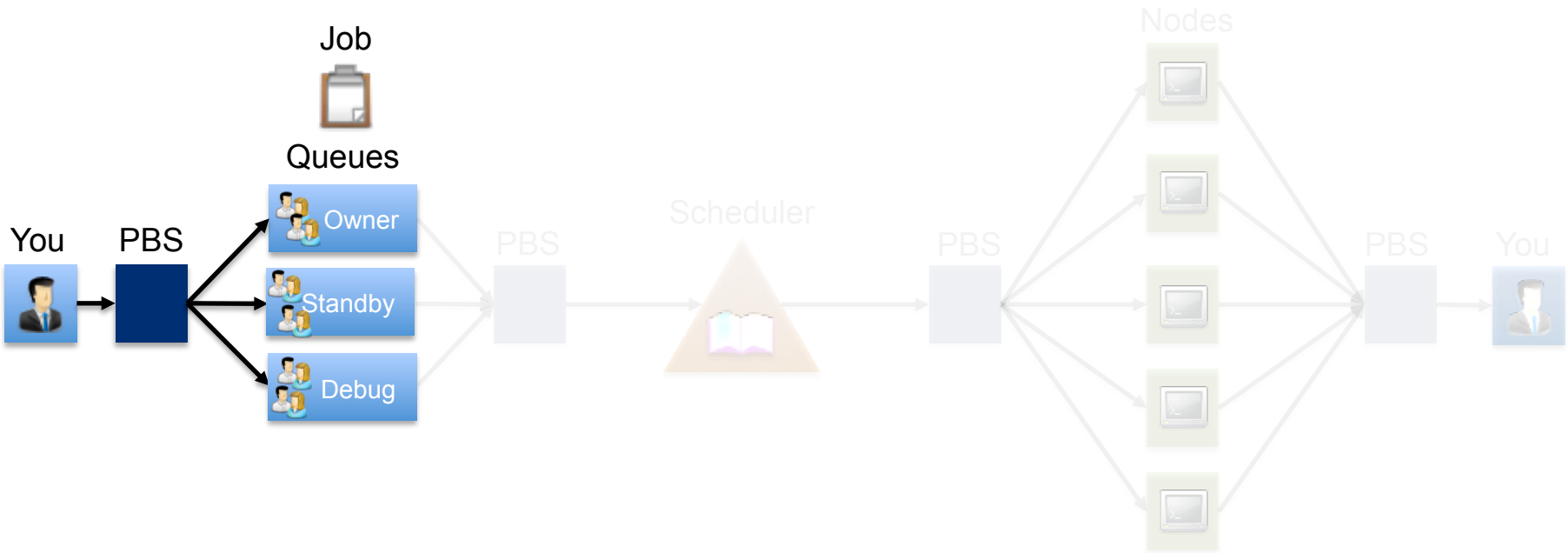
- You prepare a job submission script



# JOBS, QUEUES, AND PBS

## LIFE AS A JOB

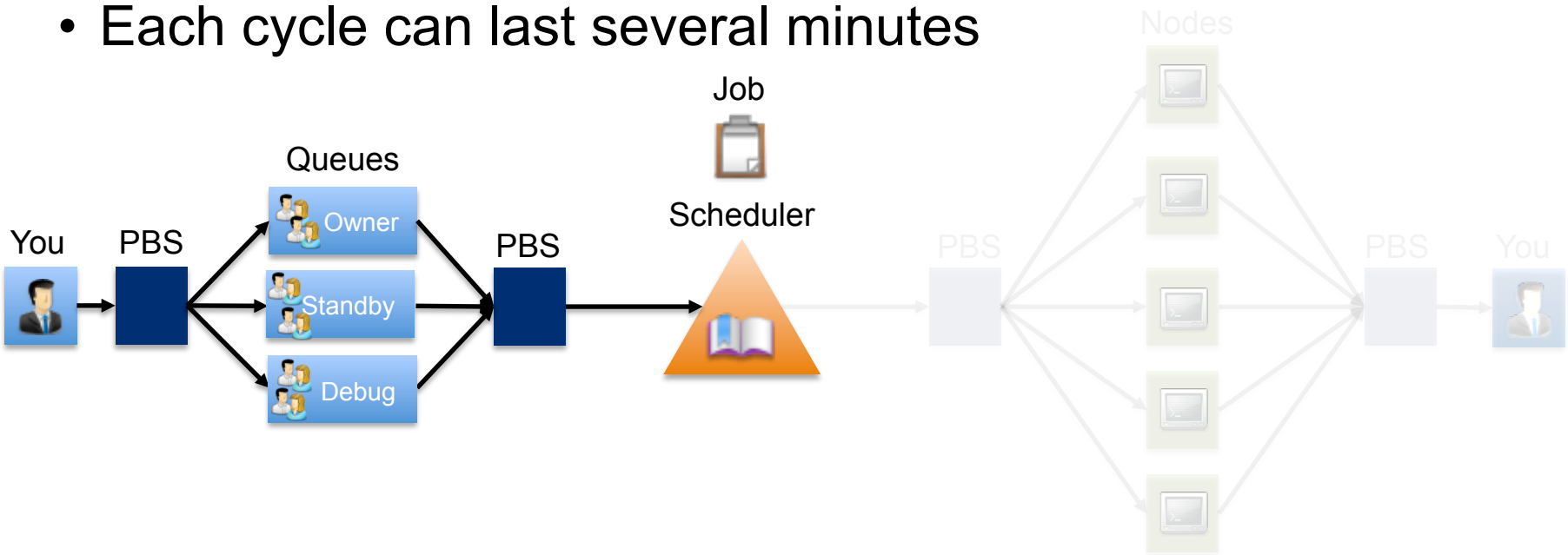
- You submit job script into PBS
- Job is placed in queue



# JOBS, QUEUES, AND PBS

## LIFE AS A JOB

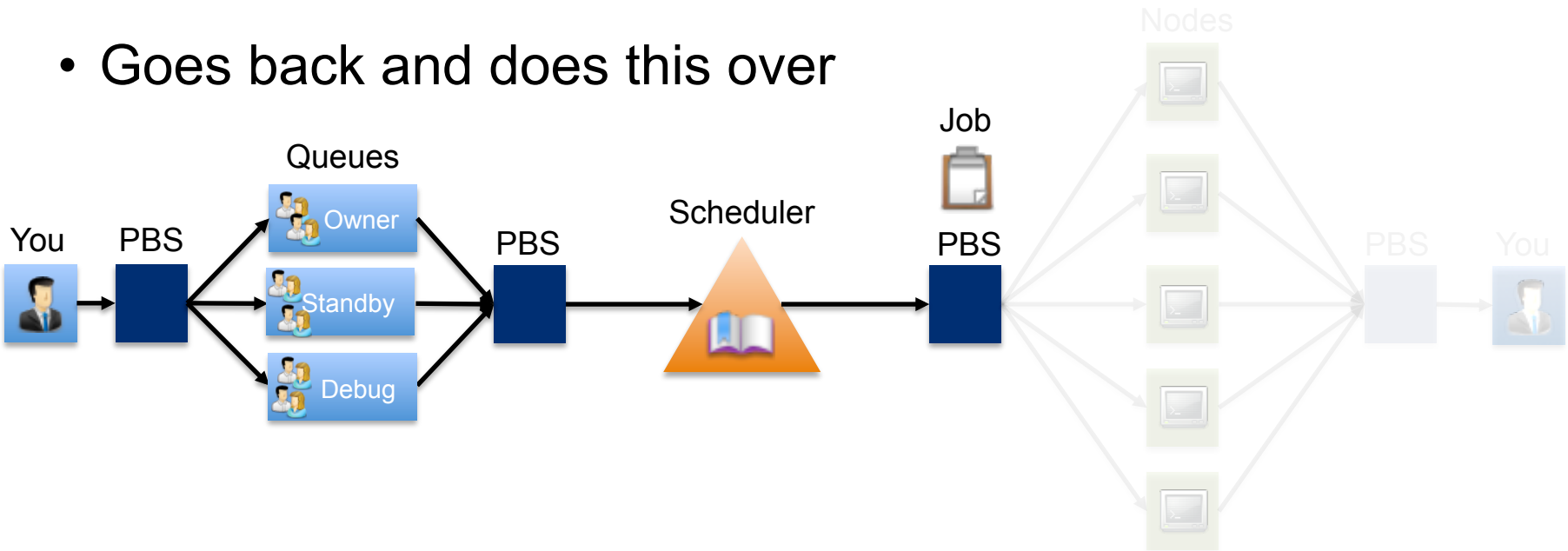
- Scheduler iteratively asks PBS for new jobs, status of old jobs, and status of nodes
- Each cycle can last several minutes



# JOBS, QUEUES, AND PBS

## LIFE AS A JOB

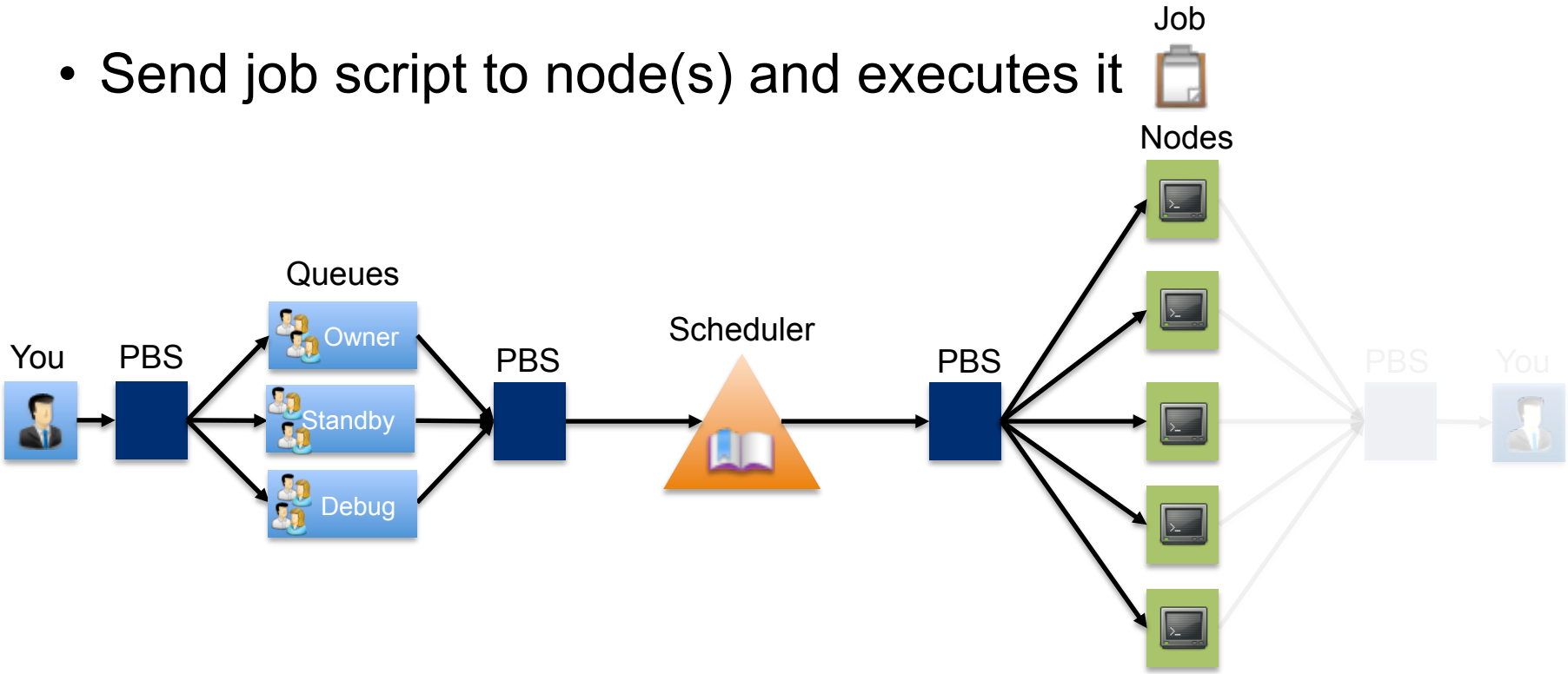
- Scheduler considers jobs, acts as a handler
- Tells PBS to start jobs one at a time
- Goes back and does this over



# JOBS, QUEUES, AND PBS

## LIFE AS A JOB

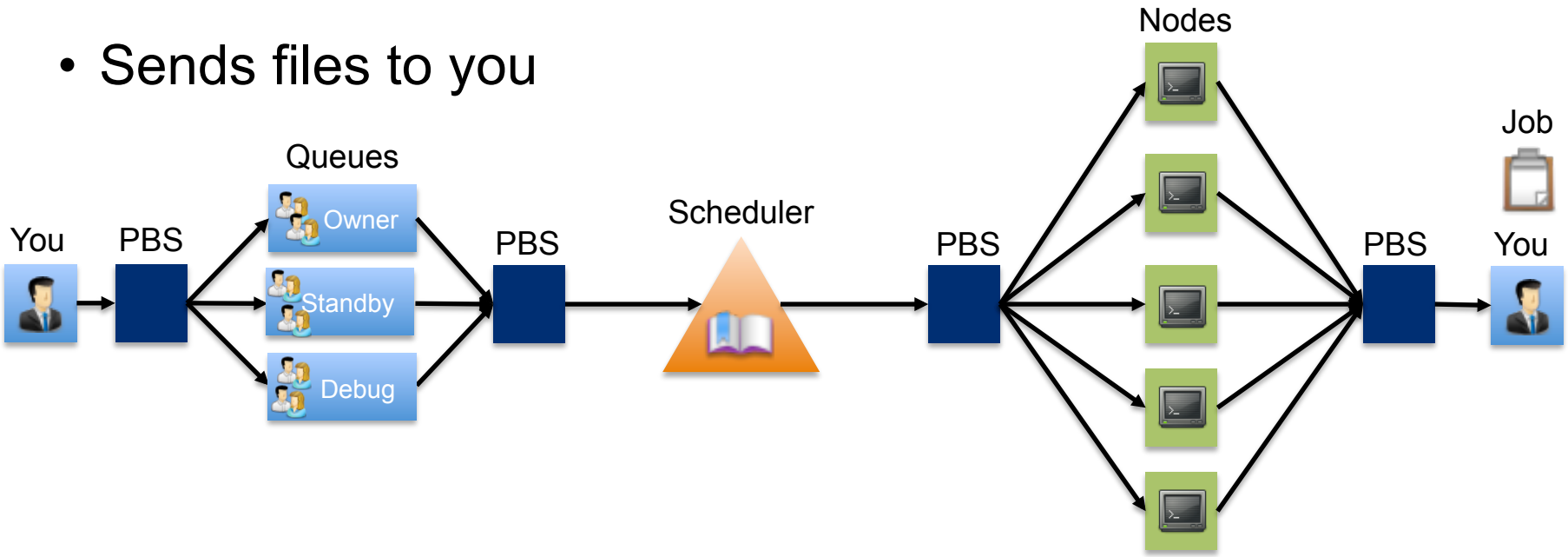
- PBS checks health of the node
- Send job script to node(s) and executes it



# JOBS, QUEUES, AND PBS

## LIFE AS A JOB

- PBS watches for job script to complete on nodes
- Collects job output files from nodes
- Sends files to you



# HOW TO CREATE A SUBMISSION SCRIPT

# HOW TO CREATE A SUBMISSION SCRIPT

## COMPONENTS OF A SUBMISSION FILE

- PBS directives
  - Specify resources needed such as number of nodes, cores
- Module load
  - Set up paths, libraries
- PBS environment variables
  - Set by PBS, can be used in your submission script
- Customized commands
  - Your job to run

# HOW TO CREATE A SUBMISSION SCRIPT

## DIRECTIVES TO PBS

- A way to set PBS job attributes
- Appear at the top of your submission file
- Common PBS job attributes include:

| PBS directives | Description |
|---|---|
| `#PBS -l nodes=2:ppn=8` | Number of nodes and cores required |
| `#PBS -q workshop` | The destination queue of the job |
| `#PBS -N test` | The name of the job |
| `#PBS -l walltime=00:10:00` | The estimated maximum walltime of the job, the job will be killed beyond this walltime |

# HOW TO CREATE A SUBMISSION SCRIPT

## LOAD MODULES

- Unload all currently loaded modules that may be accidentally loaded from sourced files such as `~/.bashrc`
- Load on the compute node(s) any relevant modules

```
module purge
module load matlab/R2014b
module load impi/5.1.1.109
```

# HOW TO CREATE A SUBMISSION SCRIPT

## CUSTOMIZED COMMANDS

- Specify everything needed for your job to run
- A simple job to print the hostname of the compute node on which the job is running:

```
/bin/hostname
```

- A slightly more complicated job to run a R script in your submission file:

```
# my_script.r is the R script that you want to run,
it takes two arguments: arg1 and arg2

Rscript my_script.r arg1 arg2
```

# HOW TO CREATE A SUBMISSION SCRIPT

## PBS RELATED ENVIRONMENT VARIABLES

- Can be used within your job submission files
- Common PBS environment variables include:

| Name | Description |
|---|---|
| `PBS_O_WORKDIR` | Absolute path of the current working directory when you submitted this job |
| `PBS_JOBID` | Job ID number assigned to this job by PBS |
| `PBS_JOBNAME` | Job name supplied by the user |
| `PBS_NODEFILE` | File containing the list of nodes assigned to this job |

# HOW TO CREATE A SUBMISSION SCRIPT

## PBS_O_WORKDIR

```
#!/bin/sh -l
# FILENAME: subjob.sub

# Change to the directory from which you originally
# submitted this job
cd $PBS_O_WORKDIR

# Print out the current working directory path
pwd
```

# HOW TO CREATE A SUBMISSION SCRIPT

## PBS_JOBID

```
#!/bin/sh -l
# FILENAME: subjob.sub

# Change to the directory from which you originally
# submitted this job
cd $PBS_O_WORKDIR

# Make a directory for each job to store the job
# output and error file
mkdir $PBS_JOBID
myprogram 1> $PBS_JOBID/myprogram.out 2> \
   $PBS_JOBID/myprogram.err
```

# HOW TO CREATE A SUBMISSION SCRIPT

## PBS_JOBNAME

```
#!/bin/sh -l
# FILENAME: subjob.sub
#PBS -N input_800

# Change to the directory from which you originally
# submitted this job
cd $PBS_O_WORKDIR

# I have an input file called input_800 (the
# PBS_JOBNAME)
./hello $PBS_JOBNAME
```

# HOW TO CREATE A SUBMISSION SCRIPT

## PBS_NODEFILE

```
#!/bin/sh -l
# FILENAME: subjob.sub

# Change to the directory from which you originally
# submitted this job
cd $PBS_O_WORKDIR

# Run a MPI program
mpiexec -n 40 -machinefile $PBS_NODEFILE./mpi_hello
```

# HOW TO CREATE A SUBMISSION SCRIPT

## Tip: BACKGROUND TASKS

- Option 1: one process per PBS job
  - Drawbacks: overhead!
- Option 2: pack lots of processes into a loop inside a single PBS jobs
  - Drawbacks:
    - need to be careful not to overwhelm the node at once,
    - slightly more complicated submission script.
  - Huge gain in efficiency.
  - NB: if you start something with a '`&`' in background, remember a '`wait`' at the end!

```
#PBS ……
command1 &
command2 &
…
command16 &
wait
```

- Scripted PBS submissions… use some `sleep` between `qsub`'s! (0.2-0.3s)

# JOBS MONITORING COMMANDS

# EXAMPLE JOB SUBMISSION SCRIPT

## subjob4cluster101.sub

```
#!/bin/sh -l
#PBS -N blastx_nr
#PBS -q workshop
#PBS -l nodes=1:ppn=8
#PBS -l walltime=03:00:00

module purge
module load bioinfo
module load blast
module list

cd $PBS_O_WORKDIR
pwd

cat subjob4cluster101.sub
```

```
date +"%d %B %Y %H:%M:%S"

# this is a test example
echo " "

blastx \
-query sequences.fasta \
-out seq.blastx_nr.fmt6 \
-db nr -num_threads 8 \
-outfmt 6 \
-evalue 1E-06 \
-max_target_seqs 5

echo " "
date +"%d %B %Y %H:%M:%S"
```

# HOW TO CHECK FOR AVAILABLE RESOURCES

## qlist

List all queues I can use and their current status and limits.

```
$ qlist

                    Current Number of Cores

 Queue            Total   Queue   Run     Free      Max Walltime

 ================  ==================================   ===============
 workq             352     900     334      18          336:00:00
 workshop           8       0       0       8            4:00:00
```

# HOW TO SUBMIT AND MONITOR YOUR JOB

## qsub

Once you have a job submission script, you may submit this script to PBS using the qsub command. PBS will find an available compute node or set of compute nodes and run your job there, or leave your job in a queue until some become available.

```
$ qsub subjob4cluster101.sub
787233.radon-adm.rcac.purdue.edu

# here the number indicates the job id that PBS
# assigns to the job, this number can be used to track
# your job.
```

# HOW TO MONITOR THE STATUS OF YOUR JOB

## qstat –u myusername

List all current jobs from the user *myusername*, where *myusername* is your Purdue login.

```
$ qstat -u gandino
 radon-adm.rcac.purdue.edu:

                                                          Req'd    Req'd       Elap
Job ID          Username    Queue     Jobname   SessID NDS TSK Memory    Time    S  Time
----------- --------    -----     -------   ------ --- --- ------   ------   -  -----
787233.radon-
adm.rcac.       gandino  workshop blastx_nr    --     1   8    --    03:00:00 Q    --
```

# HOW TO SUBMIT AND MONITOR YOUR JOB

## checkjob

This is useful when your job is not running and you want to know why

```
$ checkjob 787233
job 787233
Aname: blastx_nr
State: Idle …
BecameEligible: Mon Oct 19 17:15:04
SubmitTime:Mon Oct 19 17:13:34
...
NOTE: job req cannot run in partition radon-adm (available
procs do not meet requirements: 0 of 8 procs found)

idle procs: 8  feasible procs: 0 …
```

# HOW TO SUBMIT AND MONITOR YOUR JOB

## showstart

Will give you an estimate of when your job should start

```
$ showstart 787233

       ---------------------------------------------------
NOTE: The following information has been cached by the remote
server and may be slightly out of date
       ---------------------------------------------------

job 787233 requires 8 procs for 3:00:00
*Estimated Rsv based start in 17:14:47 on Tue Oct 20 10:39:31
Estimated Rsv based completion in 20:14:47 on Tue Oct 20 13:39:31
```

* The estimated time will vary depending on which queue your are running the job

# HOW TO SUBMIT AND MONITOR YOUR JOB

## qdel

Stop and delete the job ID *myjobid*

```
$ qstat -u gandino
 radon-adm.rcac.purdue.edu:

 Job ID                    Username    Queue    Jobname  SessID NDS  TSK   Memory   Time   S   Time
  ---------------------  ---------  --------  ---------  -----  ---  ----  ------  ------  -  -------

 787233.radon-adm.rcac.  gandino    workshop  blastx_nr  12208    1     8    --    3:00:00 R 01:00:11
$ qdel  787233
$ qstat -u gandino
 radon-adm.rcac.purdue.edu:

                                                                  Req'd   Req'd       Elap

 Job ID                    Username    Queue    Jobname  SessID NDS  TSK   Memory   Time   S   Time
  --------------------  ---------  --------  ---------  -----  ---  ----  ------  ------  -  -------

 787233.radon-adm.rcac.  gandino    workshop  blastx_nr  12208    1     8    --    3:00:00 C    --
```

# GRAB THE LAB FILES

- Login to Radon
  ```
  $ ssh <username>@radon.rcac.purdue.edu
  ```
- Change to your working space `$RCAC_SCRATCH`
  ```
  $ cd $RCAC_SCRATCH
  ```
- Copy the directory lab_cluster101 into your directory:
  ```
  $ cp -r /depot/itap/cluster101-2016/sub_job ./
  ```
- Move into the newly created lab_cluster101 directory:
  ```
  $ cd sub_job
  $ pwd
  $ ls
  ```

# LAB 3 - RUN A BATCH JOB ( QSUB )

- Open the batch script in an editor to see if you need to change it:

```
$ nano subjob4cluster101.sub  # or vi, or emacs, or just cat
subjob4cluster101.sub
```

- Launch the batch job

```
$ qsub subjob4cluster101.sub
```

- Monitor the job's status

```
$ qstat -a -u <username>
$ checkjob -v <jobid>
$ qstat -a | more    # hit space bar to advance
```

- When job completes, take a look at results:

```
$ ls                          # Note presence/names of output files
$ more subjob4cluster101.sub.oxxxxx  # "xxxxx" is your job's jobid
$ more subjob4cluster101.sub.exxxxx  # "xxxxx" is your job's jobid
```

# INTERACTIVE JOBS AND THINLINC

# INTERACTIVE JOBS

## INTERACTIVE JOB

Jobs are typically ran without user interaction, but some do need interaction.

```
$ qsub -I
qsub: waiting for job 7867033.radon-adm.rcac.purdue.edu
to start
```

The job will be queued like any other job, and may take some time to start.

```
qsub: job 7867033.radon-adm.rcac.purdue.edu ready
$ hostname
radon-a044
```

# INTERACTIVE JOBS

## WHY RUN INTERACTIVELY ON A COMPUTE NODE?

- Dedicated compute node (vs a shared frontend)
- Test code without impacting others
- Quicker develop / test / debug cycle
- Run GUI apps as a job
  - Matlab
  - IGV
  - Fluent

# INTERACTIVE JOBS

## DISPLAY FORWARDING (X11)

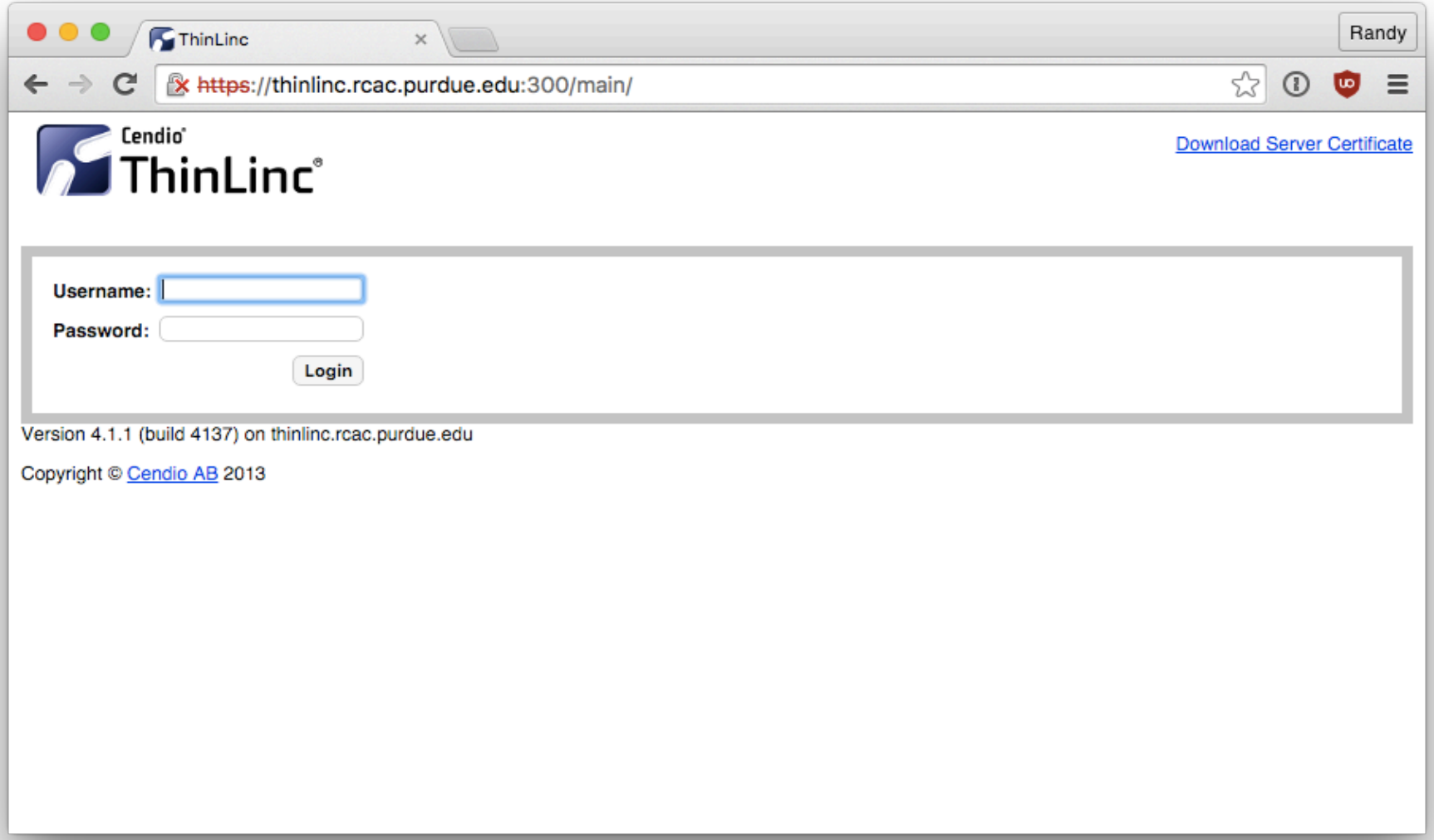
To forward your display, use "`-Y`" option to `ssh`

```
$ ssh -Y radon.rcac.purdue.edu
```

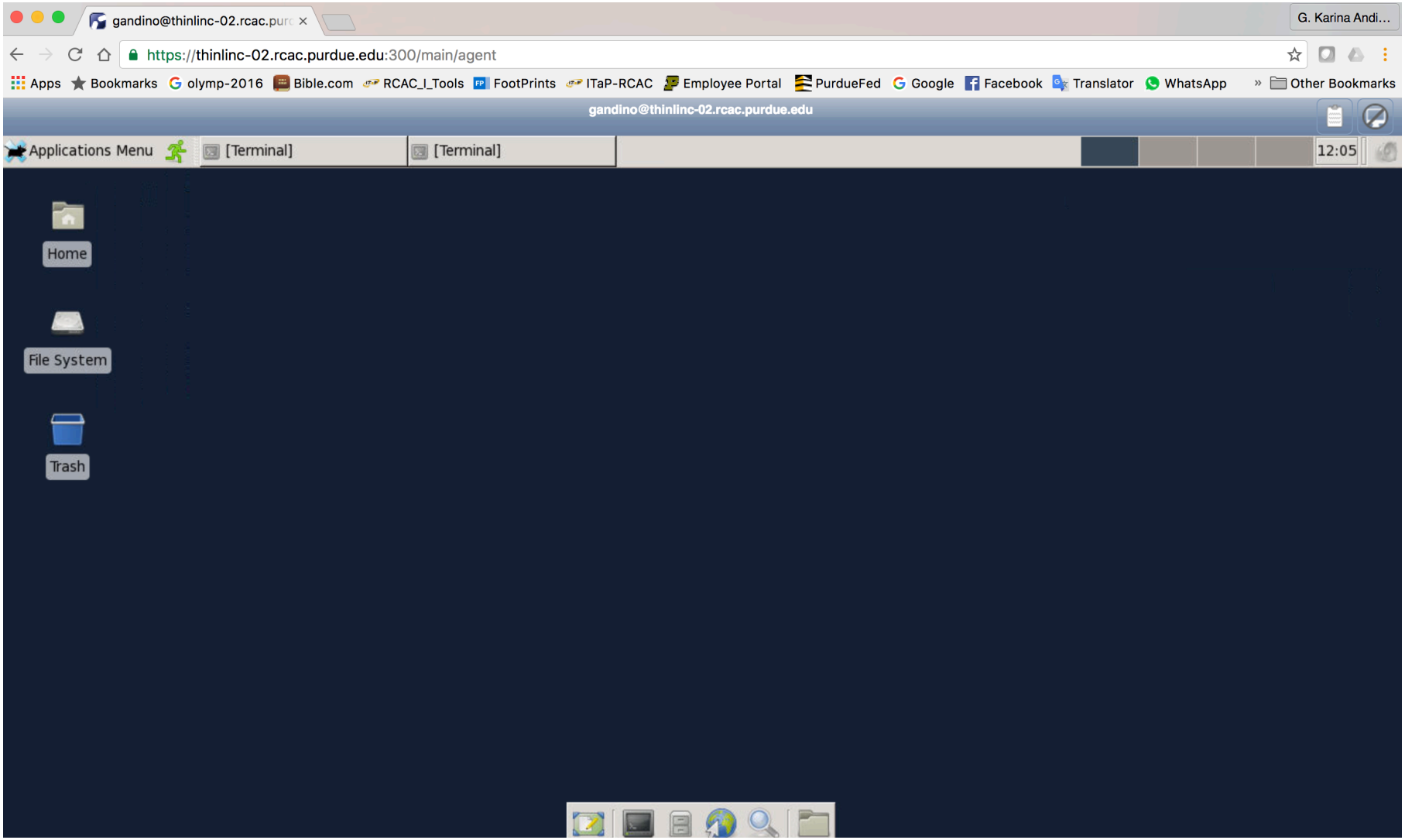
We recommend using thinlinc.rcac.purdue.edu.

- Thinlinc keeps session alive after disconnect
- Faster than X11 forwarding
- Available off-campus, but runs as on-campus (useful for licenses)
- Standalone client is also available

# THINLINC


ThinLinc
Randy
https://thinlinc.rcac.purdue.edu:300/main/
Cendio
ThinLinc
Download Server Certificate
Username:
Password:
Login
Version 4.1.1 (build 4137) on thinlinc.rcac.purdue.edu
Copyright © Cendio AB 2013

# THINLINC


gandino@thinlinc-02.rcac.purc
G. Karina Andi...
https://thinlinc-02.rcac.purdue.edu:300/main/agent
Apps
Bookmarks
olymp-2016
Bible.com
RCAC_I_Tools
FootPrints
ITaP-RCAC
Employee Portal
PurdueFed
Google
Facebook
Translator
WhatsApp
Other Bookmarks
gandino@thinlinc-02.rcac.purdue.edu
Applications Menu
[Terminal]
[Terminal]
12:05
Home
File System
Trash

# THINLINC

gandino@thinlinc-02.rcac.purd

https://thinlinc-02.rcac.purdue.edu:300/main/agent

gandino@thinlinc-02.rcac.purdue.edu

Applications Menu

Terminal

```
-bash: module: command not found
-bash: module: command not found
-bash: module: command not found

checking files that may be purged

No $RCAC_SCRATCH environment variable defined.
No $RCAC_SCRATCH environment variable defined.
(thinlinc-02) ssh -Y gandino@radon.rcac.purdue.edu
```

# THINLINC

gandino@thinlinc-02.rcac.purc

https://thinlinc-02.rcac.purdue.edu:300/main/agent

gandino@thinlinc-02.rcac.purdue.edu

Applications Menu [Terminal] Terminal 12:05

Terminal

File Edit View Terminal Go Help

```
Please send email to rcac-help@purdue.edu to report problems with this cluster.
==============================================================================
==                    Welcome to the Radon Cluster                          ==
==  Radon is composed of hyperthreaded 4-core Intel Xeon E3-1284 CPUs and   ==
==  10 Gbit Ethernet. Your scratch is mounted at the path specified in       ==
==  the environment variable $RCAC_SCRATCH.                                  ==
==                                                                           ==
==  New mailing list for researchers interested in high performance          ==
==  computing for their work.                                                ==
==  Info: https://lists.purdue.edu/mailman/listinfo/hpc                      ==
==  Subscribe with email to 'hpc-subscribe@lists.purdue.edu'                 ==
==                                                                           ==
==============================================================================
Latest module updates:
molpro 2010.1.24 was updated on Tue Oct 18 09:05
lammps 15May15_impi-5.0.2.044 was removed on Tue Oct 18 09:05
r 3.2.3 was removed on Tue Oct 18 09:05
espresso 5.2.1_impi-5.0.3.048_intel-15.0.3.187 was removed on Tue Oct 18 13:04

checking files that may be purged

Scratch filesystem not specified or no purge data found.
Scratch filesystem not specified or no purge data found.
(radon-fe00)
```

# THINLINC

gandino@thinlinc-02.rcac.purc ×

G. Karina Andi...

https://thinlinc-02.rcac.purdue.edu:300/main/agent

Apps | Bookmarks | olymp-2016 | Bible.com | RCAC_I_Tools | FootPrints | ITaP-RCAC | Employee Portal | PurdueFed | Google | Facebook | Translator | WhatsApp | » Other Bookmarks

gandino@thinlinc-02.rcac.purdue.edu

Applications Menu | [Terminal] | Terminal | 12:09

Home

File System

Trash

Terminal

File Edit View Terminal Go Help

```
==  Radon is composed of hyperthreaded 4-core Intel Xeon E3-1284 CPUs and  ==
==  10 Gbit Ethernet. Your scratch is mounted at the path specified in     ==
==  the environment variable $RCAC_SCRATCH                                 ==
==                                                                         ==
==  New mailing list for res                                               ==
==  computing for their work                                               ==
==  Info: https://lists.purdu                                              ==
==  Subscribe with email to                                                ==
==                                                                         ==
=============================================================================
Latest module updates:
molpro 2010.1.24 was updated
lammps 15May15_impi-5.0.2.04
r 3.2.3 was removed on Tue O
espresso 5.2.1_impi-5.0.3.04                                    18 13:04

checking files that may be pu

Scratch filesystem not speci
Scratch filesystem not speci
(radon-fe00) module load mat
(radon-fe00) matlab
MATLAB is selecting SOFTWARE
```

R2016a (9.0.0.341360)
64-bit (glnxa64)
February 11, 2016
License Number: 819994

MATLAB®

Academic License - for use in teaching, academic research, and meeting course requirements at degree granting institutions only. Not for government, commercial, or other organizational use.



MathWorks® R2016a



ITaP

PURDUE UNIVERSITY

# THINLINC

gandino@thinlinc-02.rcac.purc

https://thinlinc-02.rcac.purdue.edu:300/main/agent

Apps Bookmarks olymp-2016 Bible.com RCAC_I_Tools FootPrints ITaP-RCAC Employee Portal PurdueFed Google Facebook Translator WhatsApp Other Bookmarks

gandino@thinlinc-02.rcac.purdue.edu

Applications Menu | MATLAB R2016a - acad... | Terminal | Terminal | 12:10

MATLAB R2016a - academic use

HOME PLOTS APPS

New Script, New, Open, Find Files, Compare, Import Data, Save Workspace, New Variable, Open Variable, Clear Workspace, Analyze Code, Run and Time, Clear Commands, Simulink, Layout, Preferences, Set Path, Parallel, Add-Ons, Help, Community, Request Support

FILE VARIABLE CODE SIMULINK ENVIRONMENT RESOURCES

Search Documentation

/ home gandino

Current Folder

Name

90percen_virus
_BioPerl-1.6.1
BACKUP
blast2go
class_project
Desktop
Dropbox
Dropbox_oct_20_14
EMAN2DB
files_4_testing
gc_working_file

Details

Workspace

Name Value

Command Window

New to MATLAB? See resources for Getting Started.

New MATLAB Graphics System

MATLAB R2014b introduces a new MATLAB graphics system, with new default colors, fonts, and styles, and many new features. Some existing code may need to be revised to work in this version of MATLAB.

Learn more

Initializing...



ITaP

PURDUE UNIVERSITY

# INTERACTIVE JOBS AND THINLINC

## BUT WAIT, THIS JUST RAN ON THE FRONTEND!

```
$ ssh -Y radon.rcac.purdue.edu
$ echo $DISPLAY
radon-fe01.rcac.purdue.edu:27.0
$ qsub -I -q workshop -l nodes=1:ppn=8,walltime=03:00:00 -v
DISPLAY
qsub: waiting for job 903193.radon-adm.rcac.purdue.edu to start
qsub: job 903193.radon-adm.rcac.purdue.edu ready
$ echo $DISPLAY
radon-fe00.rcac.purdue.edu:27.0
```

# INTERACTIVE JOBS AND THINLINC

rherban@thinlinc-02.rcac.p

Randy

https://128.211.140.106:300/main/agent

rherban@thinlinc-02.rcac.purdue.edu

Applications Menu | Terminal - rherban@rado... | 08:21

Terminal - rherban@radon-e000:~

File Edit View Terminal Go Help

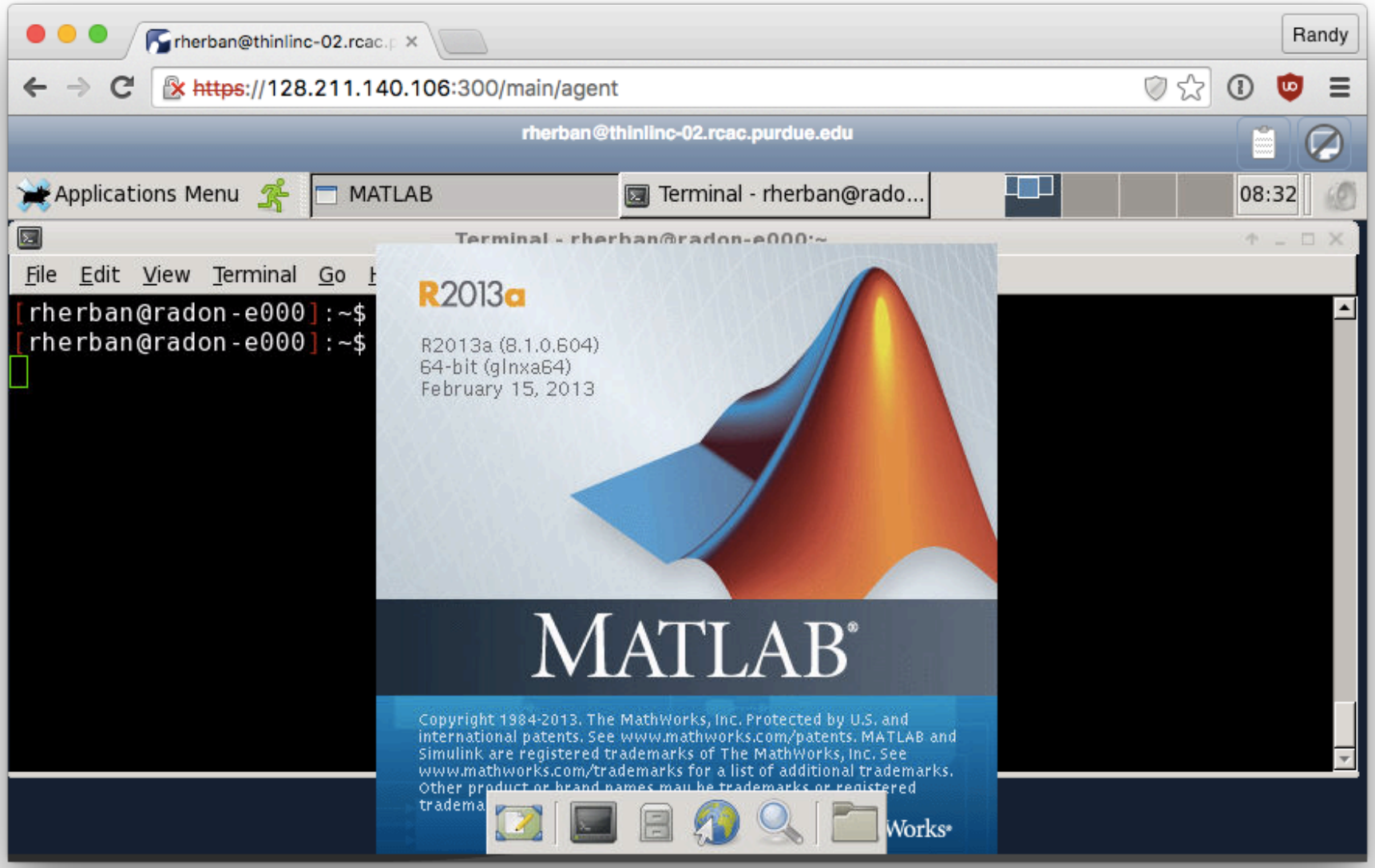
```
[rherban@radon-e000]:~$ module load matlab
[rherban@radon-e000]:~$ matlab
```



ITaP

PURDUE UNIVERSITY

# INTERACTIVE JOBS AND THINLINC



# WORKING SPACE AND STORAGE

## HIERARCHY

- Types of storage and when to use them

- $HOME vs $RCAC_SCRATCH
- Local disk (/tmp) on each individual node
- Research Data Depot
- Fortress archive

# WORKING SPACE AND STORAGE

## SUMMARY

| | $HOME | $RCAC_SCRATCH | /tmp | /depot/… | Fortress (HPSS) |
|---|---|---|---|---|---|
| **Capacity** | 25 GB | Varies by cluster… 1 to 100 TB | 150-400 GB | 100 GB and up | unlimited |
| **Resilience to hardware failures** | yes | yes | no | yes | yes |
| **Resilience to human errors** | yes (snapshots) | no | no | yes (snapshots) | no |
| **Subject to purging** | no | yes | yes | no | no |
| **Performance** | medium | high | medium | medium | slow to very slow |
| **Designed for HPC (running jobs off it)** | no | yes | no | in moderation | -No (as main I/O) -Yes (for staging and archiving) |
| **Common access within cluster** | yes | yes | no | yes | yes |
| **Common access across clusters** | yes | no | no | yes | yes |
| **Globus endpoint** | yes | yes | no | yes | yes |

# WORKING SPACE AND STORAGE

## DATA DEPOT, FORTRESS AND GLOBUS

- Data Depot - a high-capacity, fast, reliable and secure data storage service.
  - Basically, a “home directory for labs”
  - Redundant, fast, flexible, accessible from anywhere
  - Don’t have to own nodes
  - Data Depot space is mounted and accessible from all nodes and clusters

- To learn more: https://www.rcac.purdue.edu/storage/depot/

# WORKING SPACE AND STORAGE

## DATA DEPOT, FORTRESS AND GLOBUS

- Fortress - a large, long-term storage system
  - Does not support direct FTP, SFTP or SCP transfers
  - HSI - FTP-style interface without requiring any special user knowledge.
  - HTAR - a utility to aggregate a set of files into a single tar archive directly into Fortress.

- To learn more: https://www.rcac.purdue.edu/storage/fortress/

# WORKING SPACE AND STORAGE

## DATA DEPOT, FORTRESS AND GLOBUS

- Globus – a powerful and easy to use file transfer service
  - Transfer large amounts of research data to and from campus, or share data with collaborators around the country or around the globe.

- Purdue Globus Portal: https://transfer.rcac.purdue.edu

# WORKING SPACE AND STORAGE

## TIPS FROM US

- Types of storage and when to use them:
  "capacity, speed, longevity – pick any two"

- ITaP Research Computing offerings are designed around these scenarios:
  - Personal codes, executables, scripts – develop and store in $HOME
  - Lab-wide codes, executables, scripts, settings – in /depot
  - Data and results (input/output):
    - produce/analyze – in $RCAC_SCRATCH
    - permanently store – in Fortress archive and/or in /depot

- myquota command
- Use variables instead of explicit paths
- Snapshots are not a substitute for backups!

# FILE SYSTEM CITIZENSHIP

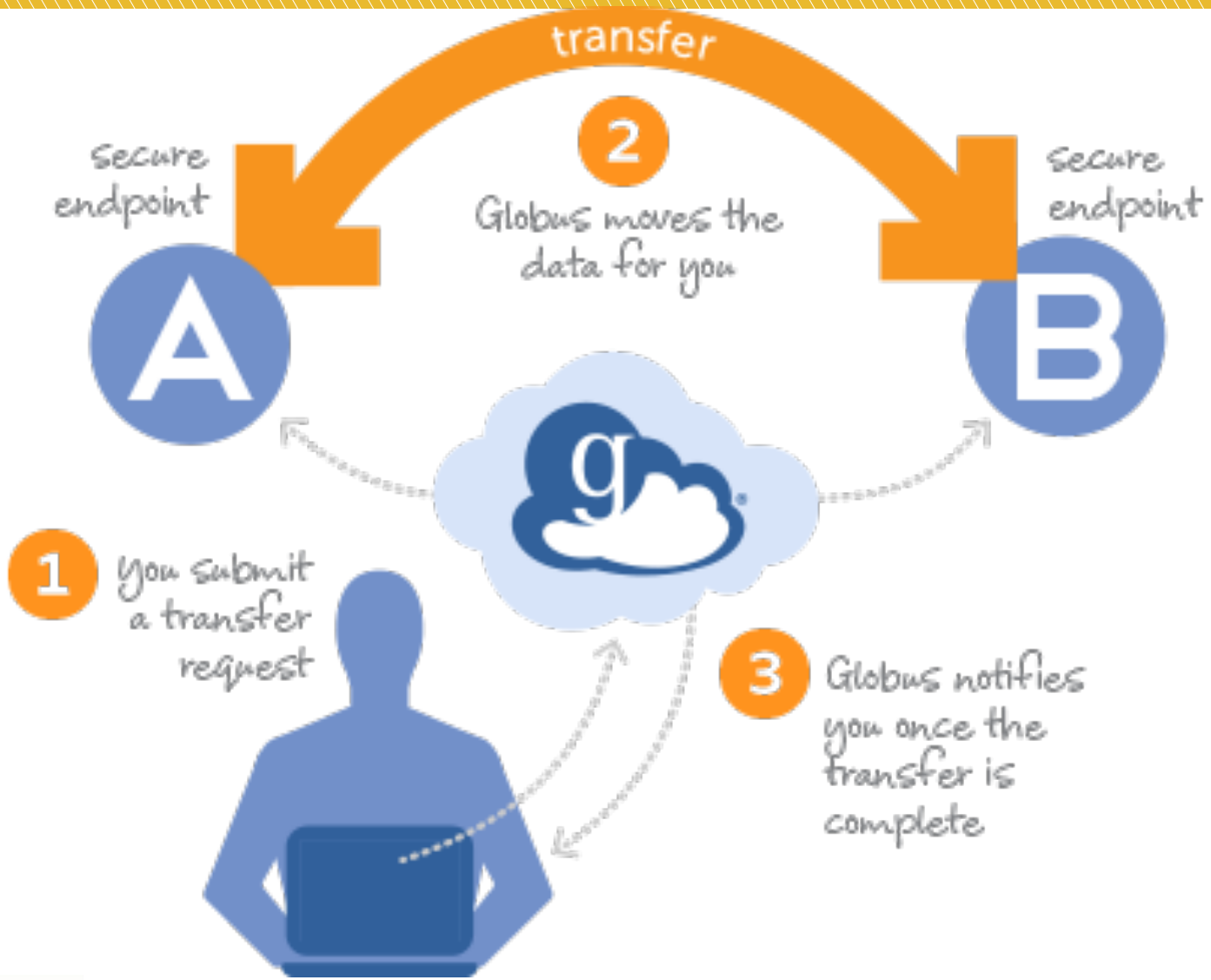
- Avoid running jobs from $HOME
- Run demanding jobs from $RCAC_SCRATCH
- Avoid frequent I/O when possible
- Minimize simultaneous I/O from many processes
- Learn to recognize/avoid other stressors
  - e.g. under-the-table stat (du, default ls) on big dirs
- Know when it's time to learn/use parallel I/O

# GLOBUS

# GLOBUS

- Globus enables file transfers between two sites (“endpoints”)
  - Like hiring movers to pack your house and get it to new location
  - You don’t have to be involved
  - Get an update when it completes / fails
  - Goes directly between the two fast sites
    - Doesn’t use your bandwidth!
- Endpoints on all of our clusters, many other Universities/ labs and even on your own laptop!

# GLOBUS




ITaP

PURDUE UNIVERSITY

# GLOBUS

- Exercise:

- Log into Globus:
- transfer.rcac.purdue.edu
- globus.org

# GLOBUS

PURDUE UNIVERSITY

globus Globus Account Log In

## Log in to use Purdue Globus Web App

### Use your existing organizational login

e.g. university, national lab, facility, project, Google or Globus ID

(Your Globus username and password used prior to February 13, 2016 is now Globus ID)

Purdue University Main Campus

Continue

Globus uses CILogon to enable you to Log In from this organization. By clicking Continue, you agree to the CILogon privacy policy and you agree to share your username, email address, and affiliation with CILogon and Globus. You also agree for CILogon to issue a certificate that allows Globus to act on your behalf.

Didn't find your organization? Then use Globus ID to sign up.

# GLOBUS

https://transfer.rcac.purdue.edu

PURDUE UNIVERSITY

Information Technology at Purdue
Research Computing (RCAC)

Manage Data | Groups | Support | Account

Transfer Files | Activity | Endpoints | Bookmarks | Publish | Console

## Transfer Files

RECENT ACTIVITY 0 0 0

Endpoint: Start here...
Path: Go

Endpoint:
Path: Go

Start by selecting an endpoint.

Start by selecting an endpoint.



ITaP

PURDUE UNIVERSITY

# GLOBUS ENDPOINTS

Endpoint purdue Cancel

**Purdue Structural Biology - Saturn**
owner: purduestbio@globusid.org
Purdue Cryo-EM Facility

**Purdue Research Computing - Home Directories**
owner: purdue@globusid.org
This endpoint provides access to home directories storage on all Purdue clusters - /home

**Purdue Research Computing - Data Depot**
owner: purdue@globusid.org
This endpoint provides access to the Purdue Research Data Depot storage - /depot

**Purdue Radon Cluster**
owner: purdue@globusid.org
This endpoint provides access to scratch storage on the Purdue Radon cluster - /scratch/radon

**Purdue Peregrine1 cluster**
owner: purdue@globusid.org
This endpoint provides access to scratch storage on the Purdue Peregrine1 cluster - /scratch/peregrine1

**Purdue Hathi Cluster**
owner: purdue@globusid.org
This endpoint provides access to scratch storage on the Purdue Hathi cluster - /hadoop/mnt/user

**Purdue Hammer Cluster**
owner: purdue@globusid.org
This endpoint provides access to scratch storage on the Purdue Hammer cluster - /scratch/hammer

# GLOBUS TRANSFER

Transfer Files | Activity | Endpoints | Bookmarks | Publish | Console

## Transfer Files

RECENT ACTIVITY 0 0 0

Endpoint: Purdue Research Computing - Home Dire

Path: /~/ Go

select all | up one folder | refresh list

unix101-2016 *Folder*

Endpoint:

Path: Go

Start by selecting an endpoint.

ITaP

PURDUE UNIVERSITY

# GLOBUS TRANSFER

Endpoint purdue

Cancel

Purdue Structural Biology - Saturn
owner: purduestbio@globusid.org
Purdue Cryo-EM Facility

Purdue Research Computing - Home Directories
owner: purdue@globusid.org
This endpoint provides access to home directories storage on all Purdue clusters - /home

Purdue Research Computing - Data Depot
owner: purdue@globusid.org
This endpoint provides access to the Purdue Research Data Depot storage - /depot

Purdue Radon Cluster
owner: purdue@globusid.org
This endpoint provides access to scratch storage on the Purdue Radon cluster - /scratch/radon

Purdue Peregrine1 cluster
owner: purdue@globusid.org
This endpoint provides access to scratch storage on the Purdue Peregrine1 cluster - /scratch/peregrine1

Purdue Hathi Cluster
owner: purdue@globusid.org
This endpoint provides access to scratch storage on the Purdue Hathi cluster - /hadoop/mnt/user

Purdue Hammer Cluster
owner: purdue@globusid.org
This endpoint provides access to scratch storage on the Purdue Hammer cluster - /scratch/hammer

# GLOBUS TRANSFER

Transfer Files | Activity | Endpoints | Bookmarks | Publish | Console

## Transfer Files

RECENT ACTIVITY 0 0 0

Endpoint: Purdue Research Computing - Home Dire
Path: /~/ Go

select all | up one folder | refresh list

- unix101-2016 *Folder*

Endpoint: Purdue Radon Cluster
Path: /scratch/radon/ Go

select all | up one folder | refresh list

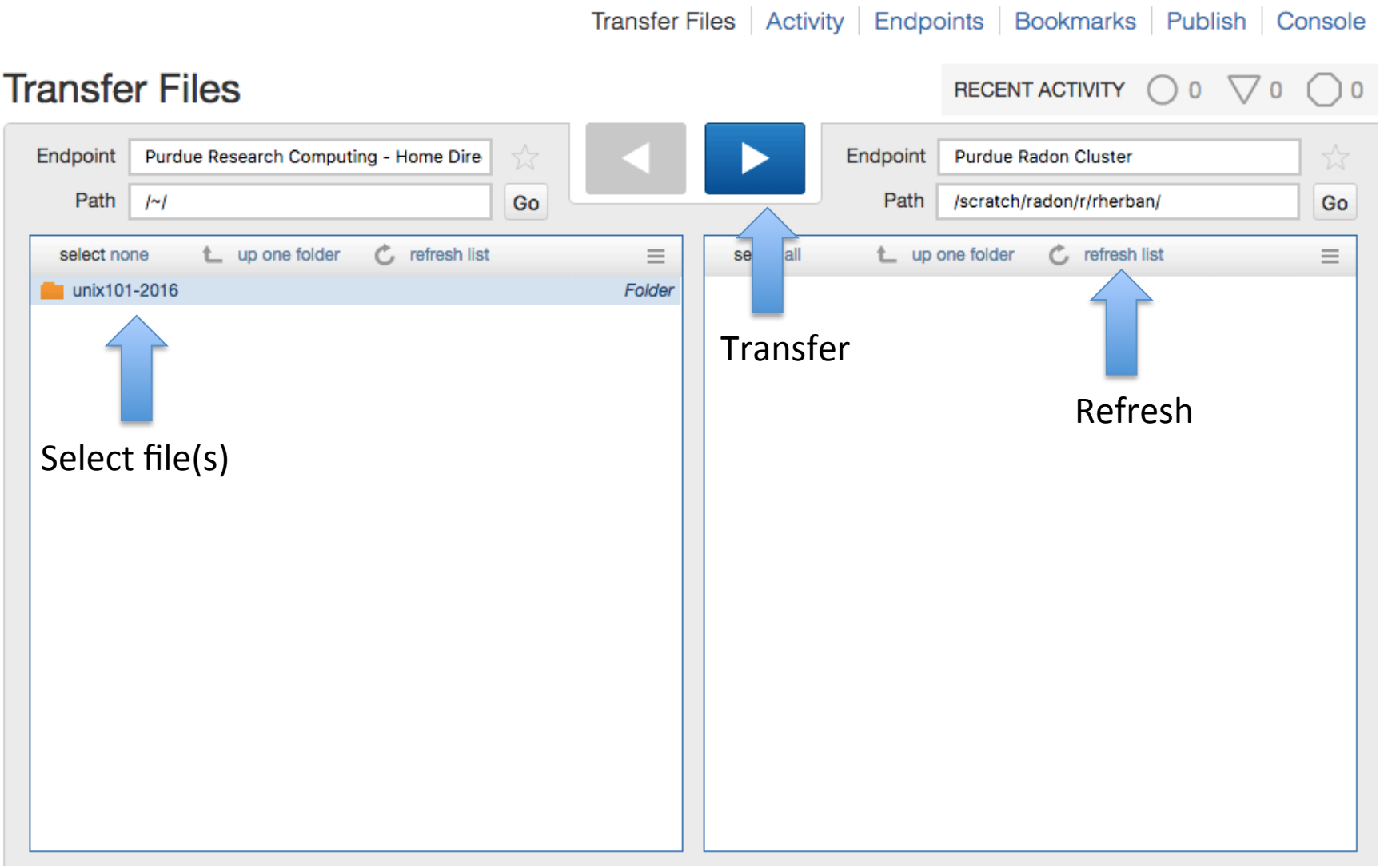
- a *Folder*
- b *Folder*
- c *Folder*
- d *Folder*
- e *Folder*
- f *Folder*
- g *Folder*
- h *Folder*
- i *Folder*
- j *Folder*
- k *Folder*
- l *Folder*
- m *Folder*
- n *Folder*
- o *Folder*
- p *Folder*
- q *Folder*
- r *Folder*
- s *Folder*
- t *Folder*

Navigate to your directory

ITaP

PURDUE UNIVERSITY

# GLOBUS TRANSFER




ITaP

PURDUE UNIVERSITY

# GLOBUS TRANSFER

Transfer Files | Activity | Endpoints | Bookmarks | Publish | Console

## Transfer Files

RECENT ACTIVITY 1 0 0

Transfer request submitted successfully. Task id: 2e22df00-961f-11e6-b0a4-22000b92c261

Endpoint: Purdue Research Computing - Home Dire
Path: /~/ Go

Endpoint: Purdue Radon Cluster
Path: /scratch/radon/r/rherban/ Go

select none | up one folder | refresh list

unix101-2016 *Folder*

select all | up one folder | refresh list

# GLOBUS

- Can share select folders with outside collaborators
  - No need to have Purdue ID
- Can set up an endpoint on your personal laptop:
  - https://www.globus.org/globus-connect-personal

# WORKFLOW

- Unless you are running a one-off simulation, you should stop and think about the workflow
- What's the optimal resource request?
  - Experimentation - scaling studies
- Do I break my workflow into multiple jobs?
- How does my data flow?
  - From original source
  - Working spaces
  - Post-processing/analysis
  - Archives

# WORKFLOW

- Example – nightly weather forecasts
- Break into multiple jobs
  - Obtain raw data (current observations)
  - Pre-process raw data and prepare
  - Run forecast
  - Copy results somewhere or post-process into viewable images

# GLOBUS & WORKFLOW

- Build your workflow
  - Grab dataset before batch of jobs

    ```
    rsync -rav /depot/group/data/experiment1 /scratch/radon/u/
    user/
    ```

  - Submit batch of jobs
    - Save to `/scratch/radon/u/user/results`
  - Copy results back to Depot or Fortress

    ```
    ssh user@cli.globusonline.org transfer -- purdue#radon:/
    scratch/radon/u/user/results purdue#depot:/depot/group/
    data/experiment1-results
    ```

    - Requires ssh key setup with Globus
    - Happens in the background; doesn't waste job time

ITaP

PURDUE UNIVERSITY

# GLOBUS & WORKFLOW

- Create GlobusID
- Go to globus.org and create Globus ID
- Log in with it and link your Purdue Credentials

# GLOBUS SSH KEY

- Can use existing ssh key

```
$ cd ~
$ ls -las .ssh
$ cat .ssh/id_rsa.pub
```

- Or generate a new key

```
$ ssh-keygen -t rsa -f .ssh/id_globus -N ""
$ cat .ssh/id_globus.pub
```

- Copy the key

# GLOBUS SSH KEY

https://transfer.rcac.purdue.edu

PURDUE UNIVERSITY

Information Technology at Purdue
Research Computing (RCAC)

Manage Data | Groups | Support | Account

## Account

Manage Your Consents | Logout

## Identities

Manage These Identities | Add Linked Identity

### Randy Herban via Globus ID

**Name:** Randy Herban
**Username:** rherban2@globusid.org
**E-mail:** rherban@purdue.edu
This is your **primary** identity.

This identity is managed by Globus ID, where you can change your password, update your profile, and manage SSH and X.509 keys.

### Randy L Herban via Purdue University Main Campus

**Name:** Randy L Herban
**Username:** rherban@purdue.edu
**E-mail:** rherban@purdue.edu
This is a **linked** identity.

ITaP

PURDUE UNIVERSITY

# GLOBUS SSH KEY

globus ID

Randy Herban (rherban2@globusid.org)
Back to Account Information on the Purdue Globus Web App
Home | Log Out

## Manage SSH and X.509 Keys

Add a New Key

You have the following key attached to your Globus ID account:

| | | | |
|---|---|---|---|
| SSH Public Key | expand | delete | rename |

ITaP

PURDUE UNIVERSITY

# GLOBUS SSH KEY

globus ID

Randy Herban (rherban2@globusid.org)
Back to Account Information on the Purdue Globus Web App
Home | Log Out

## Add a New Key

Back to Key Management

Alias: My Laptop's Key

Type
- ○ SSH Public Key
  Allows you to use the Globus command line interface via any standard `ssh` client. Some Globus-based applications may also use this SSH authorized key to provide access to their resources.
- ○ X.509 Credential
  Allows you to use the Globus command line interface via a `gsissh` client, and to activate endpoints with a delegated proxy of this X.509 credential via `gsissh` to the CLI.

Body

Paste key here

Add Key | Cancel

DUE
RSITY

# ADDITIONAL WORKSHOPS

- Software Carpentry (Oct 27 & 28) – Sign Up Now!
  - Bash Automation
  - Git
  - R / Rstudio
  - https://www.rcac.purdue.edu/news/896

- XSEDE Workshops (Monthly)
  - Big Data (Nov 1)
  - GPU Programming (Dec 6)
  - MPI
  - Summer Boot Camp