HPC User Environment 2

Feng Chen
HPC User Services
LSU HPC & LONI
sys-help@loni.org

Louisiana State University
Baton Rouge
February 2, 2022

Outline

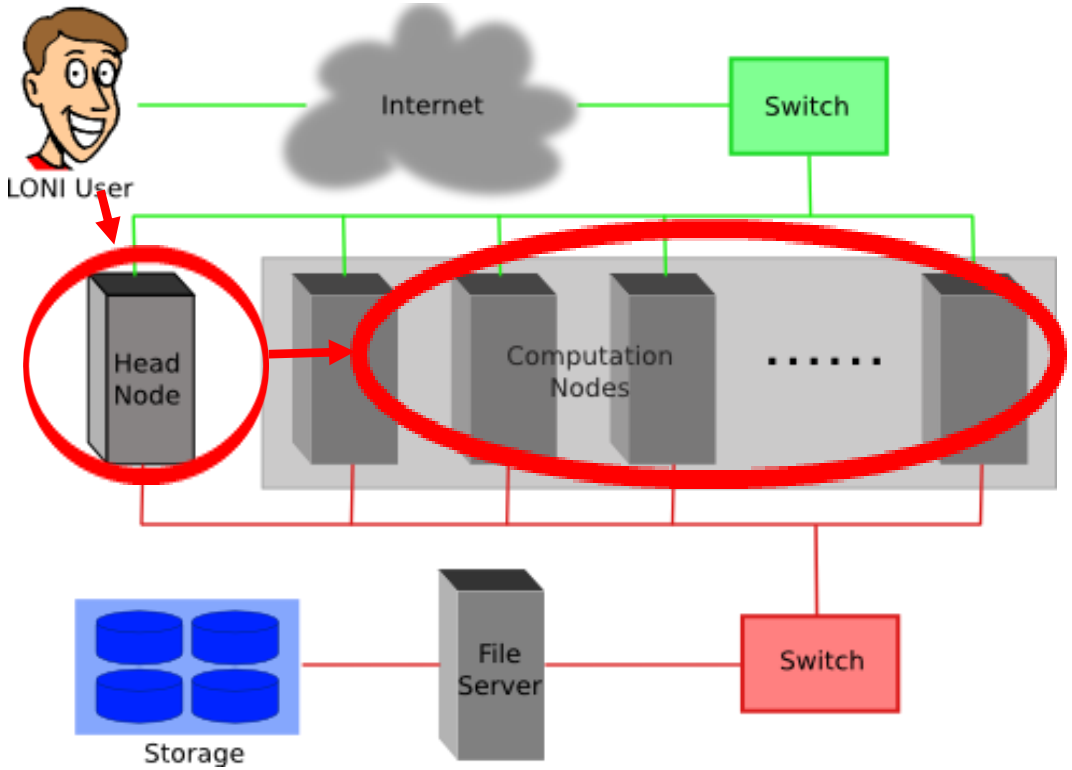
- **Review HPC User Environment 1 topics**
  - Cluster architecture
  - Connect to clusters
  - Software management using module
- **Things to be covered in this training**
  - Job management
    - Job queue basics
    - Interactive vs Batch jobs
    - Submit and monitor your jobs
  - ~~Understanding Job scheduling~~
    - ~~Job priority~~
    - ~~Backfill~~
  - ~~Compiling and analyze codes on cluster~~
    - ~~Serial program~~
    - ~~Parallel program~~

HPC User Environment 2

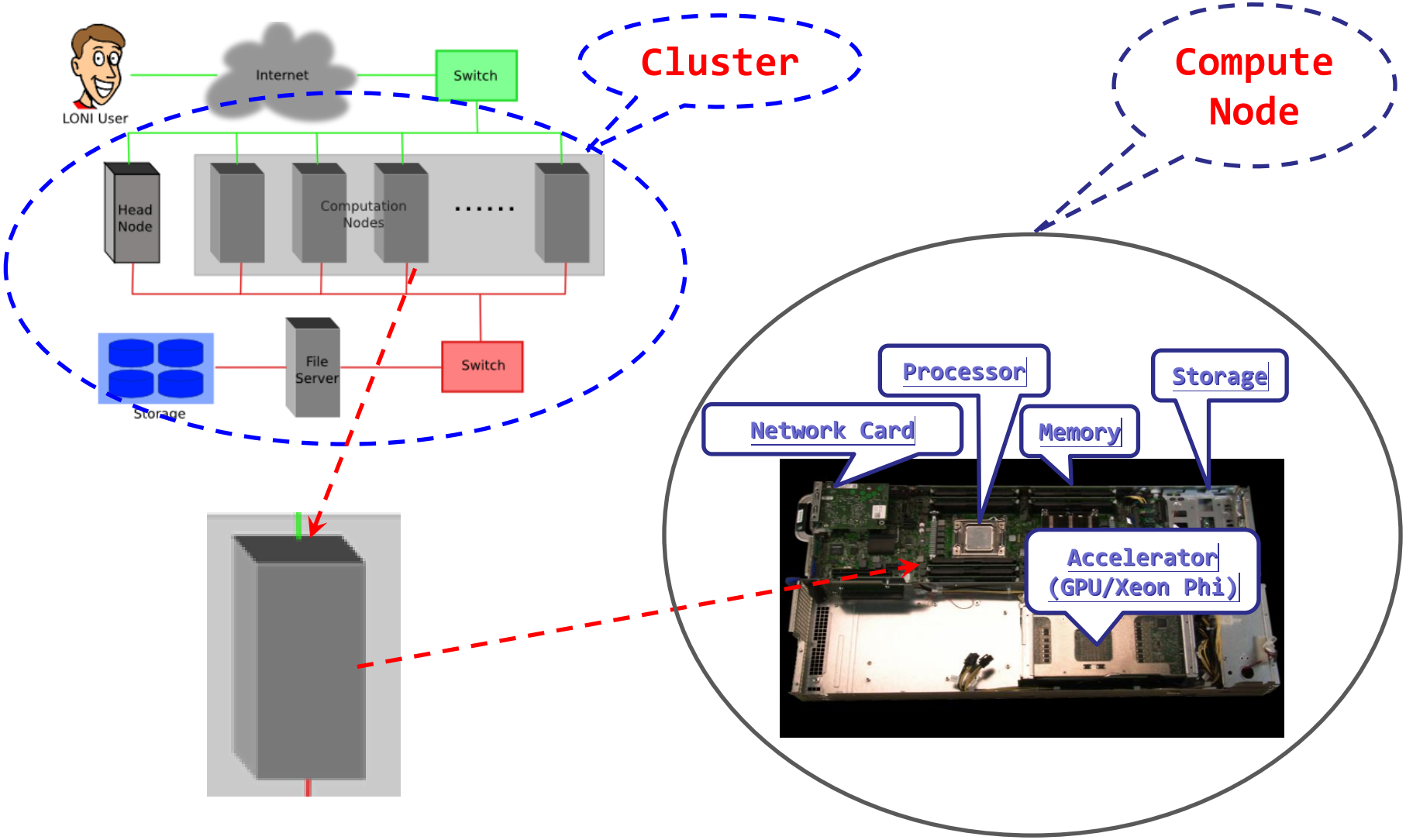
Review of HPC User Environment 1

Cluster Environment

- **Multiple compute nodes**
- **Multiple users**
- **Each user may have multiple jobs running simultaneously**
- **Multiple users may share the same node**

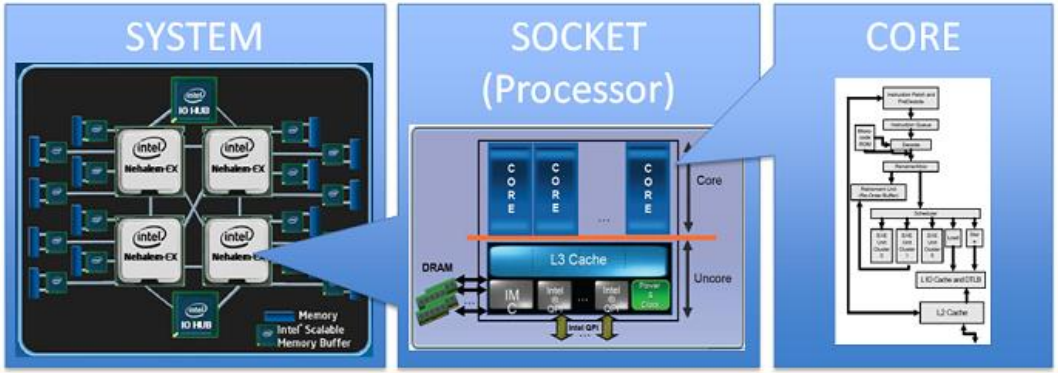


Conceptual Relationship



Cluster Nomenclature

Term	Definition
Cluster	The top-level organizational unit of an HPC cluster, comprising a set of nodes, a queue, and jobs.
Node	A single, named host machine in the cluster.
Core	The basic computation unit of the CPU. For example, a quad-core processor is considered 4 cores.
Job	A user's request to use a certain amount of resources for a certain amount of time on cluster for his work.



Accessing Cluster Using SSH (Secure Shell)

- **On Unix and Mac**
  - use ssh on a terminal to connect
- **Windows box (ssh client):**
  - MobaXterm (http://mobaxterm.mobatek.net/)
  - Putty, Cygwin (http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html)
- `ssh username@smic.hpc.lsu.edu`
- **Host name**
  - LONI: <cluster_name>.loni.org
    - <cluster_name> can be:
      - `qb.loni.org (QB2)`
      - `qbc.loni.org (QB3)`
  - LSU HPC: <cluster_name>.hpc.lsu.edu
    - <cluster_name> can be:
      - `smic.hpc.lsu.edu (SuperMIC)`
      - `db1.hpc.lsu.edu (DeepBayou)`

Software Management with Environment Modules

- **To list all available or part of packages is:** `module av`
  `module av <package name>`
- **To see what packages are currently loaded into a user's environment, the command is:** `module list`
- **The command for loading a package into a user's environment is:** `module load <package name>`. **If a specific version of a package is desired, the command can be expanded to:** `module load <package name>/<package version>`.
- **On HPC and LONI clusters, Modules can be loaded automatically on login by adding the appropriate module load commands to a user's** `~/.bashrc` **or** `~/.modules (recommended)` **file**

HPC User Environment 2

Review Questions for Section 1

Access to Cluster

- **Which supercomputer cluster can you use?**
  a) SuperMike3 (upcoming)
  b) SuperMIC (SMIC)
  c) DeepBayou
  d) QueenBee2 (QB2)
  e) QueenBee3 (QB3)

LSU HPC: a, b, c
LONI HPC: d, e

Access to Cluster

- **How do I connect to HPC/LONI cluster?**
  a) By logging onto HPC webpage at www.hpc.lsu.edu
  b) Using an ssh (secure shell) client such as MobaXterm/Putty ☺
  c) Go to the machine room in ISB in downtown Baton Rouge and connect my laptop to the nodes using a cable

- **Login onto SuperMIC or Queenbee2**

```
ssh username@smic.hpc.lsu.edu
ssh username@qb.loni.org
```

- **Windows box (ssh client):**
  - MobaXterm (recommended)
  - Putty

- **Do you need help login to the supercomputer?**

Allocation

- **To run job on the cluster, you must**
  a) Send your credit card information to the HPC staff
  b) Make sure your advisor has enough funding for you
  c) Have an activate allocation ☺
  d) All of the above

- **List active allocation balance:** balance

```
[ychen64@smic1 ~]$ balance
====================== Allocation information for ychen64 ======================
   Proj. Name|        Alloc|  Balance| Deposited|    %Used| Days Left|       End
--------------------------------------------------------------------------------
hpc_hpcadmin6|hpc_hpcadmin6 on @smic2|994513.02|1000000.00|     0.55|       287|2020-06-30
hpc_train_2019|hpc_train_2019 on @smic2| 49500.99|  50000.00|     1.00|       196|2020-03-31

Note: Balance and Deposit are measured in CPU-hours
```

Software Management

- **How do we manage the software installed on HPC/LONI clusters?**
  a) Using Environment Modules ☺
  b) Using a drop down menu on the www.hpc.lsu.edu webpage

- **Check your software environment**

```
[ychen64@smic2 ~]$ module list
Currently Loaded Modulefiles:
  1) intel/18.0.0         2) INTEL/18.0.0          3) mvapich2/2.2/INTEL-18.0.0

[ychen64@qb2 ~]$ module list
Currently Loaded Modulefiles:
  1) intel/14.0.2         2) INTEL/14.0.2          3) mvapich2/2.0/INTEL-14.0.2
```

Outline

- **Review HPC User Environment 1 topics**
  - Cluster architecture
  - Connect to clusters
  - Software management using softenv and module
- **Things to be covered in this training**
  - Job management
    - Job queue basics
    - Interactive vs Batch jobs
    - Submit and monitor your jobs

HPC User Environment 2

Job Queue Basics

Job Submission Basics

1. **Find appropriate queue**
   - Understand the queuing system and your requirements and
   - Queue Querying

2. **Submit job**
   - PBS: SuperMIC and QueenBee2
   - SLURM: DeepBayou and QueenBee2/QueenBee3, SuperMike3

3. **Monitor jobs during execution**

Job Queues

- **Nodes are organized into queues.**
- **Each job queue differs in**
  - Number of available nodes
  - Max run time
  - Max running jobs per user
  - Nodes may have special characteristics: GPU, large memory, etc.
- **Jobs need to specify resource requirements**
  - Nodes, time, queue
- **It is called a queue for a reason, but jobs don't run on a "First Come First Served" policy.**
  - This will be detailed in later slides

Queue Characteristics – LONI Clusters

Machine	Queue	Max Runtime	ppn	Max running jobs	Max nodes per job	Use
QB2	workq	3 days	20	64	128	Unpreemptable
	checkpt		20		256	Preemptable
	bigmem		48		1	Big memory
	single	7 days	1,2,4,6,8		1	Single node jobs
QB3	workq	3 days	48	64	128	Unpreemptable
	checkpt		48		256	Preemptable
	gpu		48		8	Preemptable
	bigmem		48		1	Big memory
	single	7 days	1-47		1	Single node jobs

Unpreemptable vs Preemptable

http://www.adaptivecomputing.com/blog-hpc/understanding-moab-scheduling-part-iii/

Queue Characteristics – LSU Linux clusters

Machine	Queue	Max Runtime	ppn	Max running jobs	Max nodes per job	Use
SuperMIC	workq	3 days	20	34	128	Unpreemptable
	checkpt		20		200	Preemptable
	gpu		20		20	Job using GPU
	single		1,2,4,6,8		1	Single node jobs
DeepBayou	checkpt	3 days	48	34	8	Preemptable
	nvlink		48		1	Job using GPU
	single	7days	1,2,4,6,8		1	Single node jobs
SuperMike3						
		SuperMike3 is upcoming.				

Queue Characteristics

- **"qstat -q" will give you more info on the queues**

```
[fchen14@smic2 ~]$ qstat -q

server: smic3

Queue            Memory CPU Time Walltime Node  Run Que Lm  State
---------------- ------ -------- -------- ----  --- --- --  -----
workq              --      --    72:00:00   128  31   6 --   E R
mwfa               --      --    72:00:00     8   3   0 --   E R
bigmem             --      --    48:00:00     1   0   0 --   E R
lasigma            --      --    72:00:00    28  28   7 --   E R
bigmemtb           --      --    48:00:00     1   0   0 --   E R
priority           --      --    168:00:0   128   0   0 --   E R
single             --      --    72:00:00     1  62   0 --   E R
gpu                --      --    24:00:00    16   1   0 --   E R
preempt            --      --    72:00:00   --    0   0 --   E R
checkpt            --      --    72:00:00   128  31 137 --   E R
admin              --      --    24:00:00   --    0   0 --   E R
scalemp            --      --    24:00:00     1   0   0 --   E R
                                               ----- -----
                                                 156   150
```

Queue Querying – Linux Clusters

- **Displays information about active, eligible, blocked, and/or recently completed jobs: `showq` command**

```
$ showq
active jobs------------------------
JOBID              USERNAME      STATE PROCS   REMAINING            STARTTIME
236875              ebeigi3    Running    16     1:44:29  Mon Sep 15 20:00:22
236934                 mwu3    Running    16    00:03:27  Mon Sep 15 19:04:20
...
eligible jobs----------------------
JOBID              USERNAME      STATE PROCS     WCLIMIT            QUEUETIME
236795              dmarce1       Idle  1456    00:15:00  Mon Sep 15 16:38:45
236753               rsmith       Idle  2000     4:00:00  Mon Sep 15 14:44:52
236862              dlamas1       Idle   576     2:00:00  Mon Sep 15 17:28:57
...
121 eligible jobs
blocked jobs-----------------------
JOBID              USERNAME      STATE PROCS     WCLIMIT            QUEUETIME
232741              myagho1       Idle  2000  1:00:00:00  Mon Sep  8 07:22:12
235545              tanping       Idle     1  2:21:10:00  Fri Sep 12 16:50:49
235546              tanping       Idle     1  2:21:10:00  Fri Sep 12 16:50:50
...
```

Queue Querying – Free Nodes

- **Query free nodes: `qfree` command**

```
$ qfree
PBS total nodes: 506,  free: 215,  busy: 290 *33,  down: 1,  use: 57%
PBS workq nodes: 476,  free: 190,  busy: 162,  queued: 163
PBS checkpt nodes: 476,  free: 190,  busy: 124,  queued: 284
PBS single nodes: 18,  free: 15 *258,  busy: 13,  queued: 0
PBS k40 nodes: 4,  free: 3,  busy: 1,  queued: 0
(Highest priority job 660266 on queue checkpt will start in 2:27:00)
...
```

Queue Characteristics

- **"sinfo" will give you more info on the queues (DeepBayou and QB3)**

```
[fchen14@qbc1 ~]$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
single*      up 7-00:00:00      4  drain qbc[114-115,119-120]
single*      up 7-00:00:00    119  alloc qbc[001-002,006-018,021-024,026,031-039,041-057,062-066,069-076,079-086,088-093,095-113,116-117,121-126,148-151,154-163,166,186-189]
single*      up 7-00:00:00     69   idle qbc[003-005,019-020,025,027-030,040,058-061,067-068,077-078,087,094,118,127-147,152-153,164-165,167-185,190-192]
checkpt      up 3-00:00:00      4  drain qbc[114-115,119-120]
checkpt      up 3-00:00:00    119  alloc qbc[001-002,006-018,021-024,026,031-039,041-057,062-066,069-076,079-086,088-093,095-113,116-117,121-126,148-151,154-163,166,186-189]
checkpt      up 3-00:00:00     69   idle qbc[003-005,019-020,025,027-030,040,058-061,067-068,077-078,087,094,118,127-147,152-153,164-165,167-185,190-192]
workq        up 3-00:00:00      4  drain qbc[114-115,119-120]
workq        up 3-00:00:00    119  alloc qbc[001-002,006-018,021-024,026,031-039,041-057,062-066,069-076,079-086,088-093,095-113,116-117,121-126,148-151,154-163,166,186-189]
workq        up 3-00:00:00     69   idle qbc[003-005,019-020,025,027-030,040,058-061,067-068,077-078,087,094,118,127-147,152-153,164-165,167-185,190-192]
gpu          up 3-00:00:00      8   idle qbc[193-200]
bigmem       up 3-00:00:00      2   idle qbc[201-202]
```

HPC User Environment 2

Submit and Monitor Your Jobs

Two Job Types

- **Interactive job**
  - Set up an interactive environment on compute nodes for users
    - Advantage: can run programs interactively
    - Disadvantage: must be present when the job starts
  - Purpose: testing and debugging, compiling
    - **NEVER RUN COMPUTATIONALLY INTENSIVE CODE ON THE HEAD NODE (Login Node)**
    - Try not to run interactive jobs with large core count, which is a waste of resources
- **Batch job**
  - Executed without user intervention using a job script
    - Advantage: the system takes care of everything
    - Disadvantage: can only execute one sequence of commands which cannot changed after submission
  - Purpose: production run

Check Your Available Allocations

```
[fchen14@smic1 ~]$ showquota
Hard disk quotas for user fchen14 (uid 32584):
     Filesystem       MB used       quota       files      fquota
     /home               3463        5000       26880           0
     /work            2678135           0     2405526     4000000
     /project         2678135           0     2405526     4000000

CPU Allocation SUs remaining:
    hpc_hpcadmin8:        1938592.91  2000000.00  2022-07-01
    hpc_train_2021:         27129.33    50000.00  2022-07-01
```

Submitting Jobs on Linux Clusters

- **Interactive job example:**
  - PBS for SuperMIC and QueenBee2

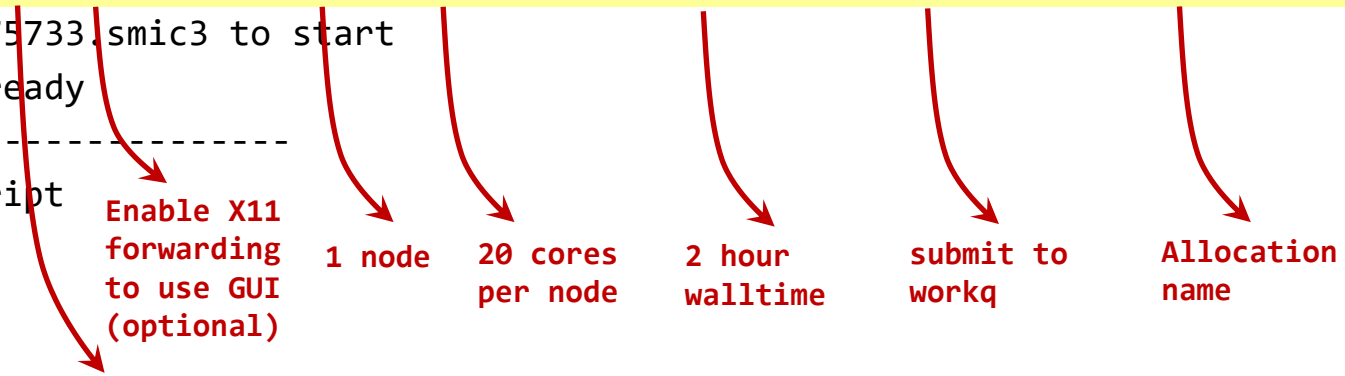
```
qsub -I \
     -l walltime=<hh:mm:ss>,nodes=<num_nodes>:ppn=<num_cores> \
     -A <Allocation> \
     -q <queue name> \
     -X to enable X11 forwarding
```

  - SLRUM for DeepBayou and QueenBee3

```
srun -t hh:mm:ss  \
     -N short for --nodes, number of nodes \
     -n short for --ntasks, number of tasks to run job on \
     -c short for --ncpus-per-task, number of threads per process \
     -A <Allocation> \
     -p  <queue name> \
     --x11 enable X11 forwarding \
      --pty bash
```

Submit a PBS Interactive Job on SuperMIC

```
[fchen14@smic1 ~]$ qsub -I -X -l nodes=1:ppn=20,walltime=2:00:00 -q workq -A hpc_train_2022
qsub: waiting for job 675733.smic3 to start
qsub: job 675733.smic3 ready
----------------------------------------
Running PBS prologue script
...
Job ID:    675733.smic3
Username:  fchen14
Group:     Admins
Date:      13-Jun-2017 15:34
Node:      smic044 (62703)
----------------------------------------
PBS has allocated the following nodes:
smic044
A total of 16 processors on 1 nodes allocated
----------------------------------------
...
Concluding PBS prologue script - 13-Jun-2017 15:34:19
----------------------------------------
[fchen14@smic044 ~]$
```



Submit a PBS Interactive Job on SuperMIC

```
[fchen14@smic1 ~]$ qsub -I -X -l nodes=1:ppn=20,walltime=2:00:00 -q workq -A hpc_train_2022
qsub: waiting for job 675733.smic3 to start
qsub: job 675733.smic3 ready
----------------------------------------
Running PBS prologue script
...
Job ID:    675733.smic3
Username:  fchen14
Group:     Admins
Date:      13-Jun-2017 15:34
Node:      smic044 (62703)
----------------------------------------
PBS has allocated the following nodes:
smic044
A total of 16 processors on 1 nodes allocated
----------------------------------------
...
Concluding PBS prologue script - 13-Jun-2017 15:34:19
----------------------------------------
[fchen14@smic044 ~]$
```

- **Note the digit change.**

Submit a PBS Interactive Job on QB2 and SMIC

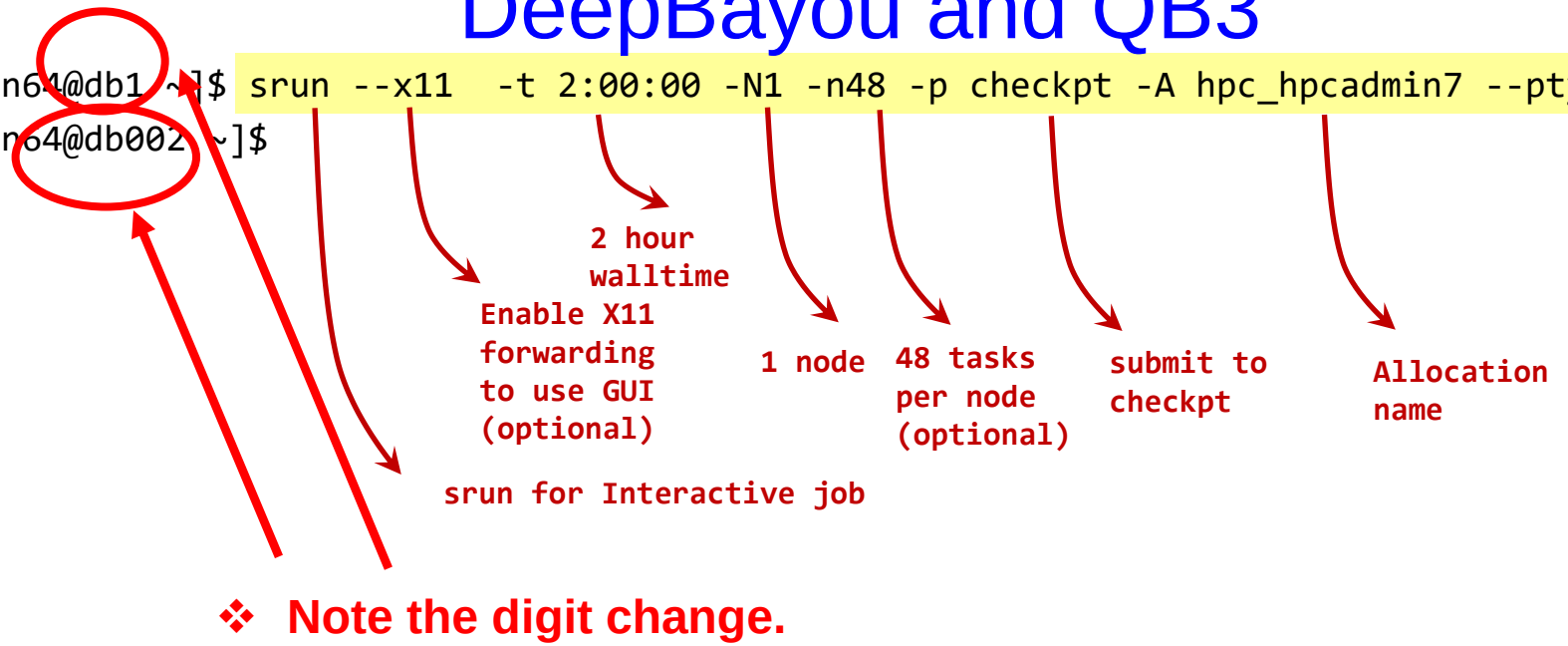
```
[ychen64@qb2 ~]$ qsub -I -X -l nodes=1:ppn=20,walltime=02:00:00, -q workq -A loni_train_2022
qsub: waiting for job 505851.qb3 to start
qsub: job 505851.qb3 ready
----------------------------------------
Running PBS prologue script
----------------------------------------
User and Job Data:
----------------------------------------
Job ID:    505851.qb3
Username:  ychen64
Group:     loniadmin
Date:      13-Jun-2018 01:27
Node:      qb061 (4497)
----------------------------------------
PBS has allocated the following nodes:
…
------------------------------------------------------
Concluding PBS prologue script - 13-Jun-2018 01:27:39
------------------------------------------------------
[ychen64@qb061 ~]$
```

20 cores per node

Allocation name

Submit a SLURM Interactive Job on DeepBayou and QB3

```
[ychen64@db1 ~]$ srun --x11  -t 2:00:00 -N1 -n48 -p checkpt -A hpc_hpcadmin7 --pty bash
[ychen64@db002 ~]$
```



- **Note the digit change.**

Exercise

- **Start an interactive job session for 1 hour (or a time with 30-min increment)**
  - Find out your allocation name if you don't remember
  - Decide how many node and which queue to use
  - Use "`qsub -I`" or "`srun`" , including all necessary options
  - Once job started, verify that you are NOT on the head node

- **Why 30 min for the interactive jobs?**
  - The requested time for the interactive jobs should have 30-min increment
  - A 30-min job is the easiest job to be fit into the job queue.
  - Based on the actual test needs, longer time can be requested, max 12 hours.

Exercise (Continue)

- **Computing an approximate value for PI**
  - cd to your work directory

    ```
    $ cd /work/$USER
    ```
  - Download the tarball from HPC website to the home directory

    ```
    $ wget http://www.hpc.lsu.edu/training/weekly-materials/Downloads/pi.tar.gz
    ```
  - Untar it

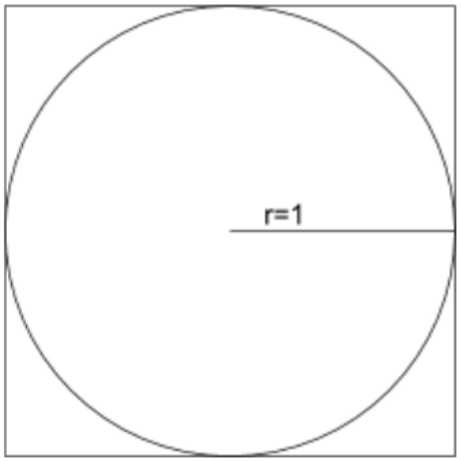
    ```
    $  tar –xvzf pi.tar.gz
    ```
  - cd to the directory “pi”

    ```
    $ cd pi
    ```
  - Use “module list” to make sure the mvapich2 is loaded.
  - Execute serial or mpi version

    ```
    $ serialpi.out #serial version, if no argument given, default value 1000000000
    # MPI version:
    # QueenBee2 or SuperMIC:
    $ mpirun -np 20 ./mpi_pi.out 100000000000  # default 1000000000000
    # DeepBayou or QueenBee3
    $ srun ./ mpi_pi.out 100000000000  # default 1000000000000
    ```

Appendix
Computing an approximate value for PI

- The executables in this training calculate the value for PI based on the math which is actually quite simple: Imagine a square dartboard with circle inscribed within it such that the diameter of the circle is the length of a side of the square.



- We can observe that the ratio of the area of the circle to the area of the square is equal to some constant, π/4 (since the square's area is 2*2 = 4 and area_circle = π*r^2 = π). If we randomly place many points (darts) inside the square, we can count how many are also inside the circle (satisfy x^2+y^2 <= 1) vs the total number of points and compute an estimate for the value of π. (Problem description is from Jared Baker, UW; Ben Matthews, NCAR)

During the break…

- **Finish the exercise run.**
- **If you are not familiar with the Linux commands used in the exercise, review the Linux commands cheat sheet in the next slide.**

Cheat Sheet of Commands in Linux

- history
- mkdir (name of file) # makes a folder
- ls # list
  -a list all files including hidden
  -l shows files with a long listing format
- cd # change directory
- pwd # shows location
- cp # copy
- rm # Remove files (careful)
- Up arrow (↑) # moves back in history
- Tab -> fills in unique file name
- Tab Tab -> press tab twice, shows all available file names

Submit a Batch Job

- **PBS batch Job example:**

```
[ychen64@qb2 pi]$ qsub qsub.submit
```

- **SLURM batch Job example:**

```
[ychen64@qbc1 pi]$ sbatch sbatch.submit
```

- **Batch job cannot be submitted when you are on the compute node**

```
[ychen64@qb023 pi]$ qsub qsub.submit
qsub: Bad UID for job execution MSG=ruserok failed validating
ychen64/ychen64 from qb023
```

- **Carefully prepare the PBS/SLURM job script**
  - examples in the next few slides

PBS Job Script – Parallel Job

```
#!/bin/bash
#PBS -l nodes=2:ppn=20          #Number of nodes and processors per node
#PBS -l walltime=24:00:00       #Maximum wall time
#PBS -N myjob                   #Job name
#PBS -o <file name>             #File name for standard output
#PBS -e <file name>             #File name for standard error
#PBS -q checkpt                 #Queue name
#PBS -A <allocation_if_needed>  #Allocation name
#PBS -m e                       #Send mail when job ends
#PBS -M <email address>         #Send mail to this address
```

Tells the scheduler how much resource you need.

```
<shell commands>
mpirun -machinefile $PBS_NODEFILE -np 40 <path_to_executable> <options>
<shell commands>
```

How will you use the resources?

- Note: don't let your <path_to_executable> <options> be the EOF
  - EOF can be <shell commands>, comments or a blank line.

SLURM Job Script – Parallel Job

```
#!/bin/bash
#SBATCH -N 2                          #number of nodes
#SBATCH -n 96                         #total number of MPI processes
#SBATCH -t hh:mm:ss                   #short for --time
#SBATCH -o <file name>                #File name for standard output
#SBATCH -e <file name>                #File name for standard error
#SBATCH -p checkpt                    #Queue name
#SBATCH -A <allocation_if_needed>     #Allocation name
#SBATCH --mail-type END               #Send mail when job ends
#SBATCH --mail-user <email>           #Send mail to this address
```

Tells the scheduler how much resource you need.

```
<shell commands>
srun <path_to_executable> <options>
<shell commands>
```

How will you use the resources?

- Note: don't let your <path_to_executable> <options> be the EOF
  - EOF can be <shell commands>, comments or a blank line.

True or False?

- **I have the below job script on QB2, since I used nodes=2:ppn=20, my script will run in parallel using 2 nodes with 40 cores.**
  a) True
  b) False

```
#!/bin/bash
#PBS -l nodes=2:ppn=20
#PBS -l walltime=24:00:00
#PBS -N myjob
#PBS -j oe
#PBS -q checkpt
#PBS -A my_allocation

./my_executable.out
```

Job Monitoring - PBS

- **Check details on your job using `qstat`**

```
$ qstat -n -u $USER  : For quick look at nodes assigned to you
```

- **Delete job using `qdel`**

```
$ qdel <jobid>
```

- **Check details of your job using `checkjob`**

```
$ checkjob <jobid>
```

- **Check health of your job using `qshow`**

```
$ qshow <jobid>
```

- **Please pay close attention to the CPU load and the memory consumed by your job!**

Job Monitoring - SLURM

- **Check details on your job using `squeue`**

  ```
  $ squeue -u $USER  : For quick look at nodes assigned to you
  ```

- **Delete job using `scancel`**

  ```
  $ scancel -c <job-id>
  ```

- **Check details of your job using `scontrol`**

  ```
  $ scontrol show job <job-id>
  ```

- **Check health of your job using `qshow`**

  ```
  $ qshow <jobid>
  ```

- **Please pay close attention to the CPU and the memory consumed by your job!**

Using the “top” command

- The Linux `top` program provides a dynamic real-time view of a running system.
- Should be used on the compute node assigned to you (ssh to it first)

```
top - 19:39:56 up 89 days,  4:13,  1 user,  load average: 0.63, 0.18, 0.06
Tasks: 489 total,   2 running, 487 sleeping,   0 stopped,   0 zombie
Cpu(s):  6.3%us,  0.0%sy,  0.0%ni, 93.7%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:  65909356k total,  3389616k used, 62519740k free,   151460k buffers
Swap: 207618040k total,     5608k used, 207612432k free,   947716k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
39595 fchen14   20   0  266m 257m  592 R 99.9  0.4   0:06.94 a.out
39589 fchen14   20   0 17376 1612  980 R  0.3  0.0   0:00.05 top
38479 fchen14   20   0  108m 2156 1348 S  0.0  0.0   0:00.03 bash
39253 fchen14   20   0  103m 1340 1076 S  0.0  0.0   0:00.00 236297.smic3.SC
39254 fchen14   20   0  103m 1324 1060 S  0.0  0.0   0:00.00 bm_laplace.sh
39264 fchen14   20   0 99836 1908  992 S  0.0  0.0   0:00.00 sshd
39265 fchen14   20   0  108m 3056 1496 S  0.0  0.0   0:00.03 bash
```

Using the “free” command

- The Linux `free` displays the total amount of free and used physical and swap memory in the system
- Should be used on the compute node assigned to you (ssh to it first)

```
$ free -h
             total       used       free     shared    buffers     cached
Mem:           62G       3.1G        59G       177M        31M       1.3G
-/+ buffers/cache:       1.7G        61G
Swap:         127G         0B       127G
```

PBS Environmental Variables

```
[fchen14@smic315 ~]$ echo $PBS_
```

$PBS_ENVIRONMENT	$PBS_MOMPORT	$PBS_NUM_PPN	$PBS_O_MAIL
$PBS_QUEUE	$PBS_WALLTIME	$PBS_GPUFILE	**$PBS_NODEFILE**
$PBS_O_HOME	$PBS_O_PATH	$PBS_SERVER	$PBS_JOBCOOKIE
$PBS_NODENUM	$PBS_O_HOST	$PBS_O_QUEUE	$PBS_TASKNUM
$PBS_JOBID	$PBS_NP	$PBS_O_LANG	$PBS_O_SHELL
$PBS_VERSION	$PBS_JOBNAME	$PBS_NUM_NODES	$PBS_O_LOGNAME
$PBS_O_WORKDIR	$PBS_VNODENUM		

SLURM Environmental Variables

```
[ychen64@qbc025 ~]$ echo $SLURM_
```

$SLURM_CLUSTER_NAME	$SLURM_JOB_UID	$SLURM_STEPID
$SLURM_CPU_BIND	$SLURM_JOB_USER	$SLURM_STEP_ID
$SLURM_CPU_BIND_LIST	$SLURM_LAUNCH_NODE_IPADDR	$SLURM_STEP_LAUNCHER_PORT
$SLURM_CPU_BIND_TYPE	$SLURM_LOCALID	**$SLURM_STEP_NODELIST**
$SLURM_CPU_BIND_VERBOSE	$SLURM_MPI_TYPE	$SLURM_STEP_NUM_NODES
$SLURM_CPUS_ON_NODE	$SLURM_NNODES	$SLURM_STEP_NUM_TASKS
$SLURM_GTIDS	$SLURM_NODEID	
$SLURM_STEP_TASKS_PER_NODE		
$SLURM_JOB_ACCOUNT	$SLURM_NODELIST	**$SLURM_SUBMIT_DIR**
$SLURM_JOB_CPUS_PER_NODE	$SLURM_NPROCS	$SLURM_SUBMIT_HOST
$SLURM_JOB_GID	$SLURM_NTASKS	$SLURM_TASK_PID
$SLURM_JOBID	$SLURM_PRIO_PROCESS	$SLURM_TASKS_PER_NODE
$SLURM_JOB_ID	$SLURM_PROCID	$SLURM_TOPOLOGY_ADDR
$SLURM_JOB_NAME	$SLURM_PTY_PORT	
$SLURM_TOPOLOGY_ADDR_PATTERN		
$SLURM_JOB_NODELIST	$SLURM_PTY_WIN_COL	$SLURM_UMASK
$SLURM_JOB_NUM_NODES	$SLURM_PTY_WIN_ROW	$SLURM_WORKING_CLUSTER
$SLURM_JOB_PARTITION	$SLURM_SRUN_COMM_HOST	
$SLURM_JOB_QOS	$SLURM_SRUN_COMM_PORT	

Exercise

- **Submit a batch job**
  - cd to the directory “pi”

    ```
    $ cd pi
    ```
  - edit qsub.submit (change allocation name, email, ppn=, mpirun etc.)

    ```
    $ vi qsub.submit
    ```
  - submit job

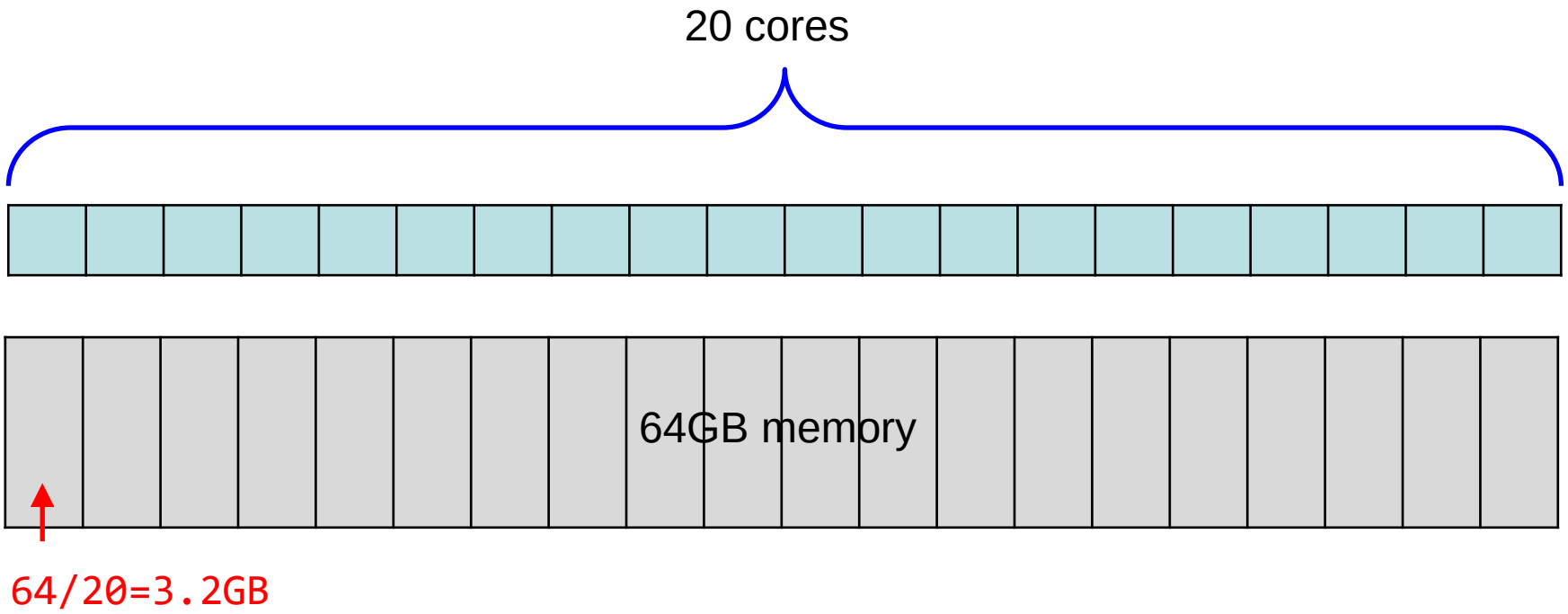
    ```
    $ qsub qsub.submit
    ```
- **Check details on your job using `qstat` or `squeue`**

  ```
  $ qstat -n -u $USER   #PBS
  $ squeue -l -u $USER  #SLURM
  ```
- **Monitor the job**
  - **`qshow` or `scontrol`**
  - **`top`** (must ssh to the compute node assigned to your job)
  - **`free`** (must ssh to the compute node assigned to your job)

Pay attention to single queue usage

- **Single queue - Used for jobs that will only execute on a single node, i.e. `nodes=1:ppn=1/2/4/6/8`.**
- **Jobs in the single queue should not use:**
  - More than 3.2GB memory per core for QB2 and SuperMIC (64G/20).
  - More than 4.0GB memory per core for QB3 (192G/48).
- **If applications require more memory, scale the number of cores (ppn) to the amount of memory required: i.e. max memory available for jobs in single queue is 8GB for -n2 on QB3.**
- **Typical type of warning:**
  - E124 - Exceeded memory allocation. This Job XXXX appears to be using more memory (GB) than allocated (9 > 3).
  - E123 - Exceeded ppn/core allocation. This Job XXXX appears to be using more cores than allocated (6 > 1). Please allocate the number of cores that the job will use, (ppn=6). This Job has 1 core(s) allocated (ppn=1).

Core and Memory in Single queue



Question:

On QB2, if my job needs 7GB memory, what ppn value should I use?

On QB3, if my job needs 7GB memory, what -n value should I use?

PBS Job Script – Serial Job

```
#!/bin/bash
#PBS -l nodes=1:ppn=1        # Number of nodes and processor
#PBS -l walltime=24:00:00    # Maximum wall time
#PBS -N myjob                # Job name
#PBS -o <file name>          # File name for standard output
#PBS -e <file name>          # File name for standard error
#PBS -q single               # The queue for serial jobs
#PBS -A <loni_allocation>    # Allocation name
#PBS -m e                    # Send mail when job ends
#PBS -M <email address>      # Send mail to this address
```

Tells the job scheduler how much resource you need.

```
<shell commands>
<path_to_executable> <options>
<shell commands>
```

How will you use the resources?

- Note: don't let your <path_to_executable> <options> be the EOF
  - EOF can be <shell commands>, comments or a blank line.

More things to be noticed

- The purpose of bigmem queue is for jobs costing big (larger than 64 GB) memory not for jobs using more number of cores.
- GPU is available in workq or checkpt queues on QB-2.
- Users are encouraged to use accelerators (GPU) whenever possible. Application for allocation involving with usage of accelerators will be easier to be approved.

Job Submission Quiz

- **How to suspend your account? (cont'd)**
  - Use more memory than allowed. (e.g. use 5GB memory on SuperMIC with ppn=1)
  - Seriously underutilize node resources (e.g. allocate 32 nodes but just use 1 core)
  - Submit job to the big memory queue but use only few MB of memory
  - Repeatedly running intensive jobs on the headnode (login node)

- **How to monitor core and memory usage?**

Summary

- **Review of HPC User Environment 1 topics**
- **Understand job queues**
- **How to submit jobs**
  - Interactive vs batch job
  - How to submit both jobs
  - How to monitor jobs

Future Training

- **1. February 2,2022: HPC User Environment 2**
- ***2. February 9,2022: Basic Shell Scripting***

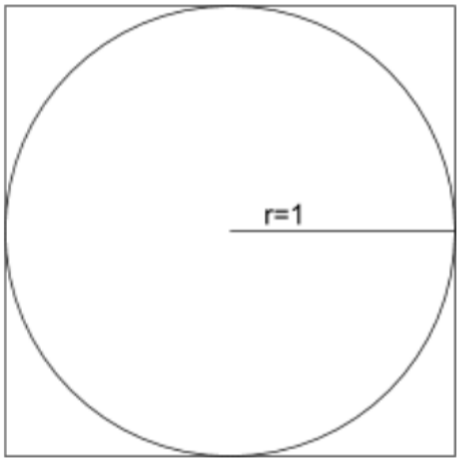
- **Keep an eye on:**
  - http://www.hpc.lsu.edu/training/tutorials.php#upcoming

HPC@LSU User Services

- **Hardware resources**
  - Currently manages 5 clusters
- **Software stack**
  - Communication software
  - Programming support: compilers and libraries
  - Application software
- **Contact user services**
  - Email Help Ticket: ***sys-help@loni.org***
  - Telephone Help Desk: +1 (225) 578-0900

Appendix
Computing an approximate value for PI

- The executables in this training calculate the value for PI based on the math which is actually quite simple: Imagine a square dartboard with circle inscribed within it such that the diameter of the circle is the length of a side of the square.



- We can observe that the ratio of the area of the circle to the area of the square is equal to some constant, π/4 (since the square's area is 2*2 = 4 and area_circle = π*r^2 = π). If we randomly place many points (darts) inside the square, we can count how many are also inside the circle (satisfy x^2+y^2 <= 1) vs the total number of points and compute an estimate for the value of π. (Problem description is from Jared Baker, UW; Ben Matthews, NCAR)