Using MPI at ACCRE

ACCRE Overview

ACCRE OVERVIEW

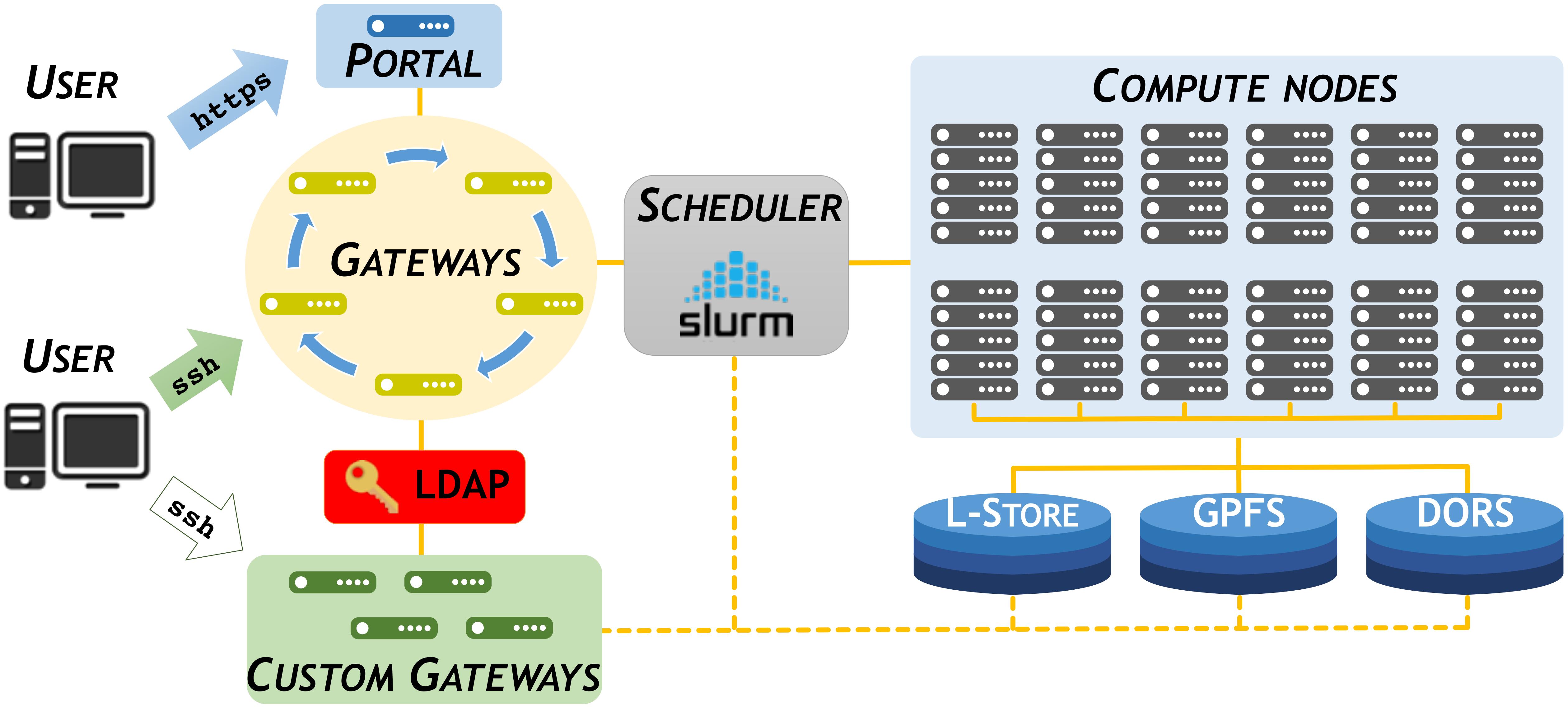
ACCRE Overview

- ACCRE - Advanced Computing Center For Research and Education
- Centralized computing infrastructure for Vanderbilt researchers
- Operates as a co-op in which researchers share hardware
- ~10k CPU cores
- ~200 GPUs
- ~10PB disk storage + tape backups
- Optimized Scientific Software Stack
- Batch Job Scheduler
- Interactive resources (Jupyter, etc.)
- Staff of ~10

Using ACCRE vs Using Your Own Hardware

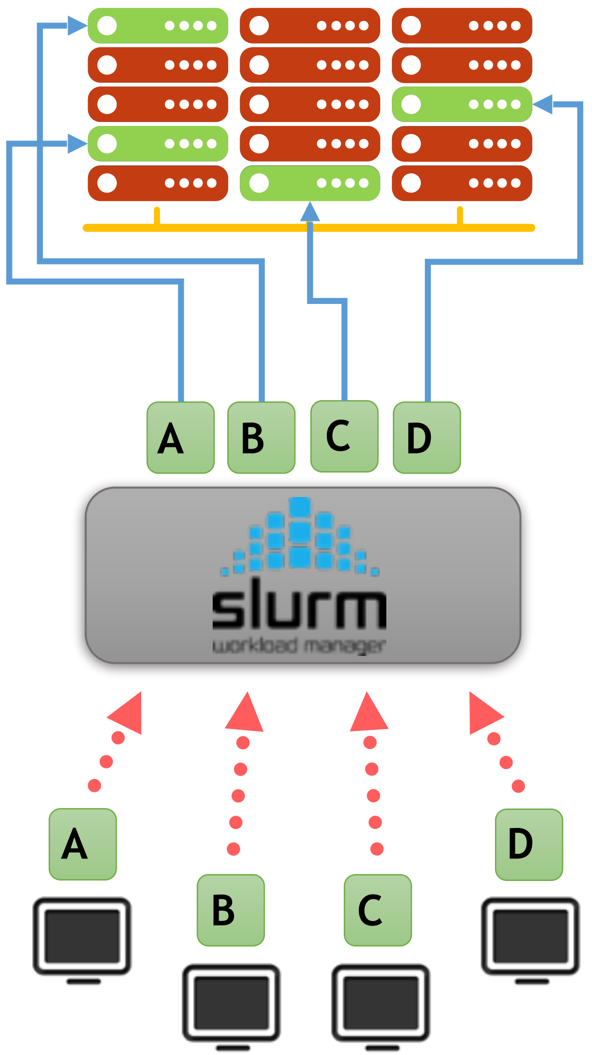
- Using your own hardware:
  - can use all resources immediately
  - have to set up software, system, and networking yourself
  - full administrative access (root)
- Using ACCRE:
  - must schedule resource requirements
  - can "burst" to use more resources than you own
  - dedicated staff maintain system and software stack
  - no administrative access (regular user)

ACCRE ARCHITECTURE



The scheduler

1. Execute user's workloads in the right priority order
2. Provide requested resources on compute nodes
3. Optimize cluster utilization



ACCRE is a Heterogeneous Cluster

- Different Memory Configurations and CPU Core Counts
  - Nodes with 64GB, 128GB, 192GB, 256GB, and 384GB
  - Between 8 and 32 CPU-cores per node
- Different Intel CPU Architecture Families
  - Variable clock speed, L1/2/3 Cache Memory
  - Additional Instruction Sets on Newer CPUs
- Specialized Accelerated Nodes
  - Nvidia 4x GPU Nodes (Maxwell, Pascal, Turing)

ACCRE is a Heterogeneous Cluster

- Heterogeneity is fine for "embarrassingly" parallel tasks
  - Don't care if jobs start at the same time
  - Don't care if some jobs are slower than others
- Not so great for distributed simulations
  - Need to synchronize between iterations
  - Limited to speed of slowest nodes
- ACCRE cluster users can choose to run on all architectures or specific architectures

Intel CPU Families

Intel CPU Families (on ACCRE)

- Westmere (~2010)
  - 64-bit extensions, MMX, SSE, SSE2, SSE3, SSSE3, SSE4.1, SSE4.2, POPCNT, AES and PCLMUL
- Sandy Bridge (~2011)
  - 64-bit extensions, MMX, SSE, SSE2, SSE3, SSSE3, SSE4.1, SSE4.2, POPCNT, **AVX**, AES and PCLMUL
- Haswell (~2014)
  - 64-bit extensions, **MOVBE**, MMX, SSE, SSE2, SSE3, SSSE3, SSE4.1, SSE4.2, POPCNT, AVX, **AVX2**, AES, PCLMUL, **FSGSBASE**, **RDRND**, **FMA**, **BMI**, **BMI2** and **F16C**
- Skylake (~2015)
  - 64-bit extensions, MOVBE, MMX, SSE, SSE2, SSE3, SSSE3, SSE4.1, SSE4.2, POPCNT, **PKU**, AVX, AVX2, AES, PCLMUL, FSGSBASE, RDRND, FMA, BMI, BMI2, F16C, **RDSEED**, **ADCX**, **PREFETCHW**, **CLFLUSHOPT**, **XSAVEC**, **XSAVES**, **AVX512F**, **CLWB**, **AVX512VL**, **AVX512BW**, **AVX512DQ** and **AVX512CD**
- Cascade Lake (~2019)
  - 64-bit extensions, MOVBE, MMX, SSE, SSE2, SSE3, SSSE3, SSE4.1, SSE4.2, POPCNT, PKU, AVX, AVX2, AES, PCLMUL, FSGSBASE, RDRND, FMA, BMI, BMI2, F16C, RDSEED, ADCX, PREFETCHW, CLFLUSHOPT, XSAVEC, XSAVES, AVX512F, CLWB, AVX512VL, AVX512BW, AVX512DQ, AVX512CD and **AVX512VNNI**

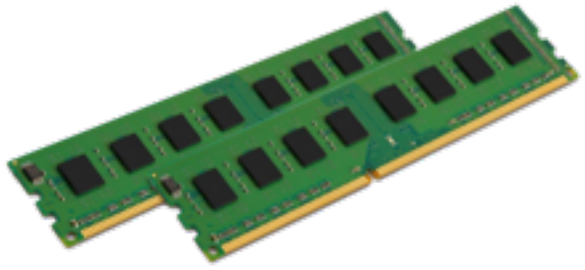
ACCRE Cluster compute nodes

ACCRE
Advanced Computing Center for Research & Education

Regular nodes

Dual multicore CPUs

Random Access Memory



Newer
Older

Family	*No. of cores*	*RAM* / GB	*No. of nodes*
Skylake	16	*256*	41
	24	*128*	52
Haswell	12	*128*	41
	16	*128*	120
		256	50
Sandy Bridge	12	*64*	31
		96	2
		128	193
		256	4
	16	*128*	3
Westmere	8	*128*	22
	12	*48*	16
Total	8,292	82,432	575

Using ACCRE

USING ACCRE

Shell Access to the Cluster

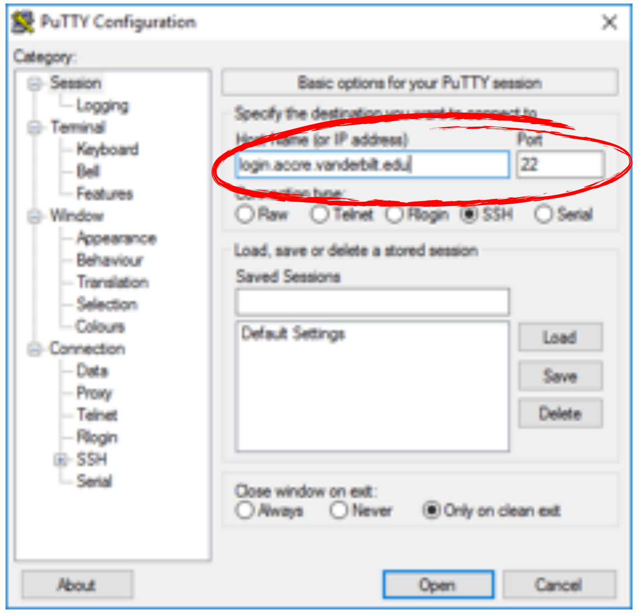
ACCRE
Advanced Computing Center for Research & Education

Users can log into the cluster with a Secure Shell (**ssh**) client.

Mac OS

From a terminal: `ssh vunetid@login.accre.vanderbilt.edu`

PuTTY


PuTTY Configuration

Category:
- Session
  - Logging
- Terminal
  - Keyboard
  - Bell
  - Features
- Window
  - Appearance
  - Behaviour
  - Translation
  - Selection
  - Colours
- Connection
  - Data
  - Proxy
  - Telnet
  - Rlogin
  - SSH
  - Serial

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): login.accre.vanderbilt.edu

Port: 22

Connection type: ( ) Raw ( ) Telnet ( ) Rlogin [✓ SSH] ( ) Serial

Load, save or delete a stored session

Saved Sessions

Default Settings

Load | Save | Delete

Close window on exit: ( ) Always ( ) Never [✓ Only on clean exit]

About | Open | Cancel

Bash on Windows

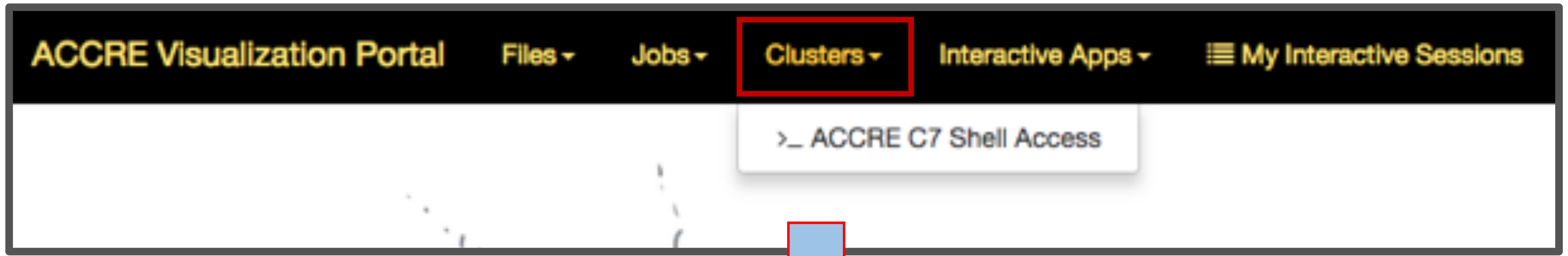
Full Ubuntu-based Bash shell

To install:

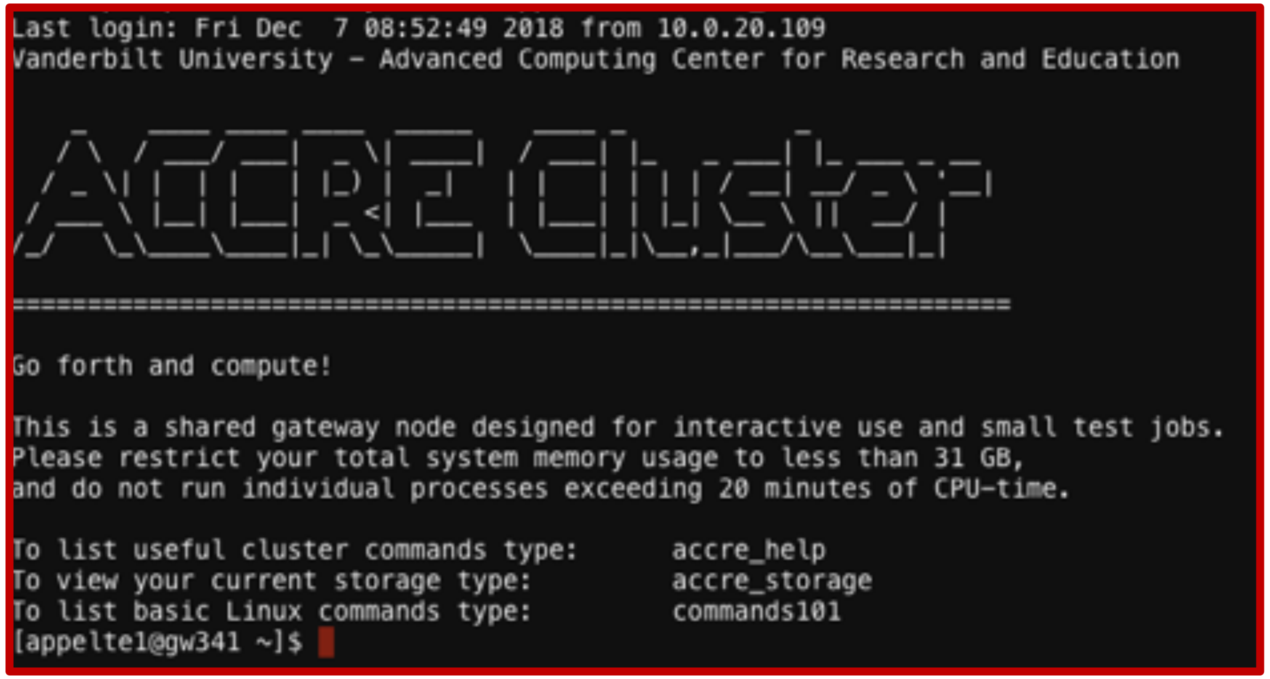
www *https://goo.gl/tAsj8U*

Windows 10 only

Shell Access to the Cluster - Portal



Opens in a new browser tab



```
Last login: Fri Dec  7 08:52:49 2018 from 10.0.20.109
Vanderbilt University - Advanced Computing Center for Research and Education

=====================================================================

Go forth and compute!

This is a shared gateway node designed for interactive use and small test jobs.
Please restrict your total system memory usage to less than 31 GB,
and do not run individual processes exceeding 20 minutes of CPU-time.

To list useful cluster commands type:      accre_help
To view your current storage type:         accre_storage
To list basic Linux commands type:         commands101
[appelte1@gw341 ~]$
```

Cluster storage

GPFS and **DORS** are distributed parallel filesystems that allow users to get access to the same set of directories on <u>all nodes</u> and <u>all gateways</u> on the cluster.

	Nightly backup	*Included with account*	*For purchase*
/home	✓	✓	✗
/scratch	✗	✓	✓
/data	✓	✗	✓

/dors

Managed by ***Center for Structural Biology***, supported by ACCRE.

Provides easy access to data from both desktops and cluster.

GPFS STORAGE QUOTA

QUOTA: When exceeded the user receives a warning message.
Usage has to return below the quota within the **GRACE PERIOD**.

LIMIT: Cannot be exceeded.
Automatically set to the actual quota usage when grace period expires.

	Data size		*Number of files*		GRACE PERIOD
	QUOTA	LIMIT	QUOTA	LIMIT	
/home	15 GB	20 GB	200,000	300,000	7 days
/scratch	50 GB	200 GB	200,000	1,000,000	14 days
/data	Can be purchased at 1 TB increments.				

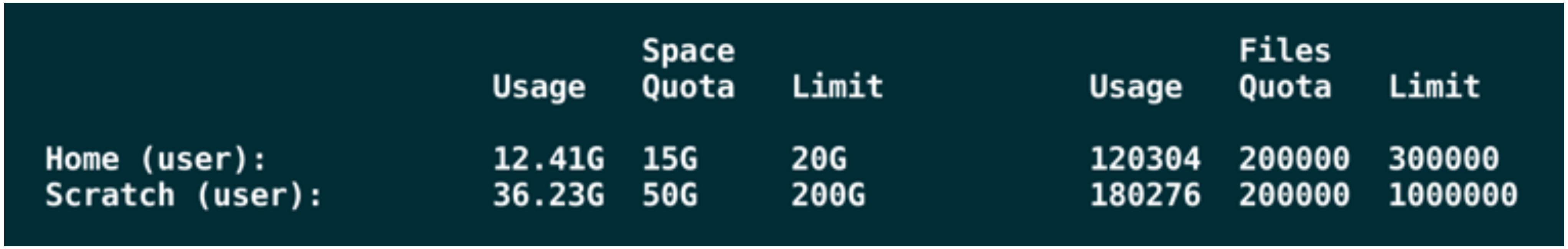
Note that groups may also purchase additional scratch quota in 1 TB increments.

GPFS STORAGE QUOTA

How can I check my current quota usage?

`accre_storage`

- Shows the current usage for all quotas associated with the user.



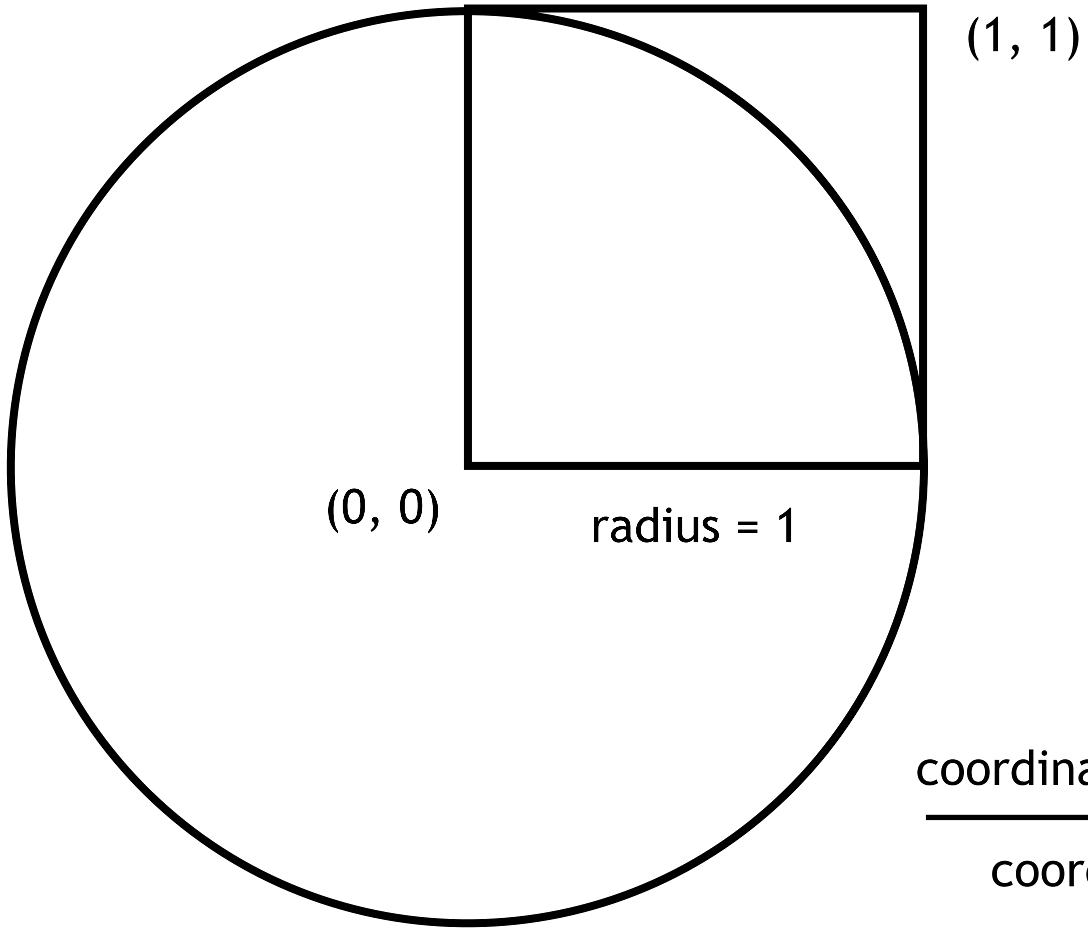
```
                                 Space                              Files
                       Usage     Quota     Limit           Usage     Quota     Limit

Home (user):           12.41G    15G       20G             120304    200000    300000
Scratch (user):        36.23G    50G       200G            180276    200000    1000000
```

Demo Time

- Demo Cluster Login
- Demo Portal Usage
- Demo ACCRE Storage

A Simple Example Program with Lmod

An Inefficient Way to Calculate Pi

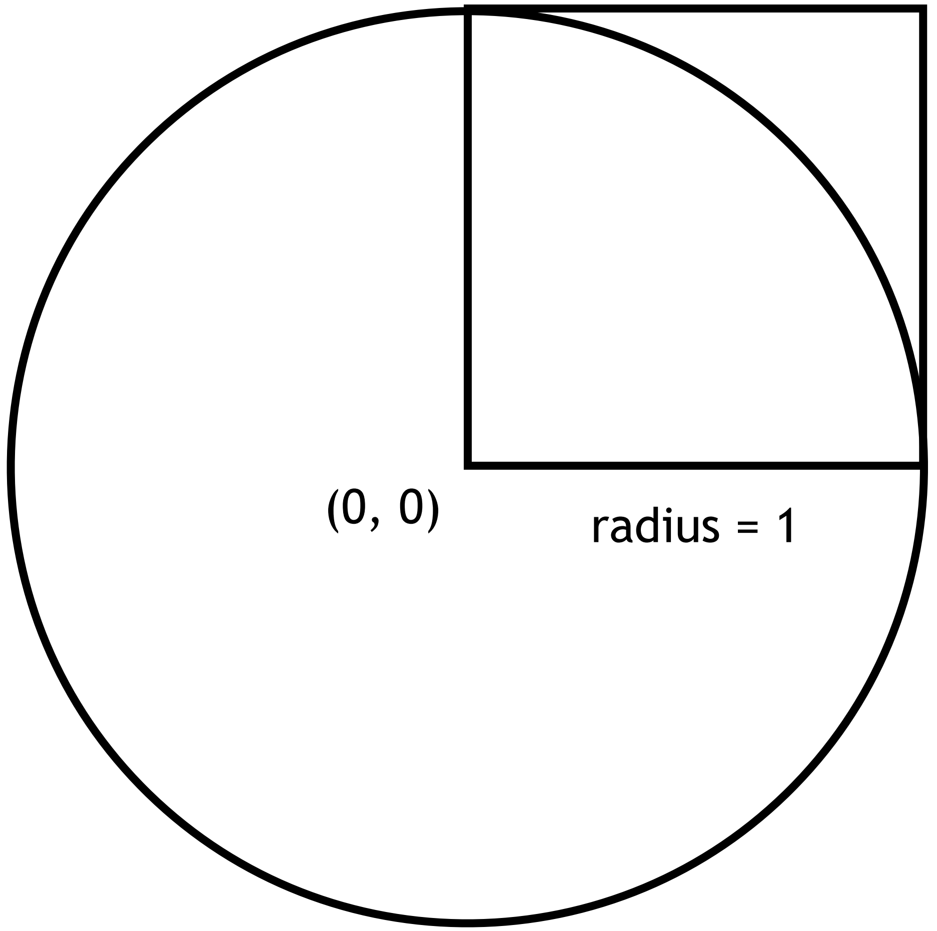


$$\frac{\text{Area Quarter Circle}}{\text{Area Square}} = \frac{\text{pi / 4}}{1} = \text{pi / 4}$$

Pick lots of random coordinates from (0, 0) to (1, 1)

$$\frac{\text{coordinates in the quarter circle}}{\text{coordinates in the square}} \approx \frac{\text{Area Quarter Circle}}{\text{Area Square}} = \text{pi / 4}$$

An Inefficient Way to Calculate Pi



$$\frac{\text{coordinates in the quarter circle}}{\text{coordinates in the square}} \approx \text{pi} / 4$$

```
attempts = 0
hits = 0

for i = 0; i < N; i++
    x = random(0, 1)
    y = random(0, 1)
    attempts++
    if x*x + y*y < 1
        hits++

return 4 * hits / attempts
```

The software stack

Researchers should focus on the **science** behind the software they use.

- Easily search available software
- Effortless access to selected software
- Seamless shell initialization
- Incompatible software conflict resolution

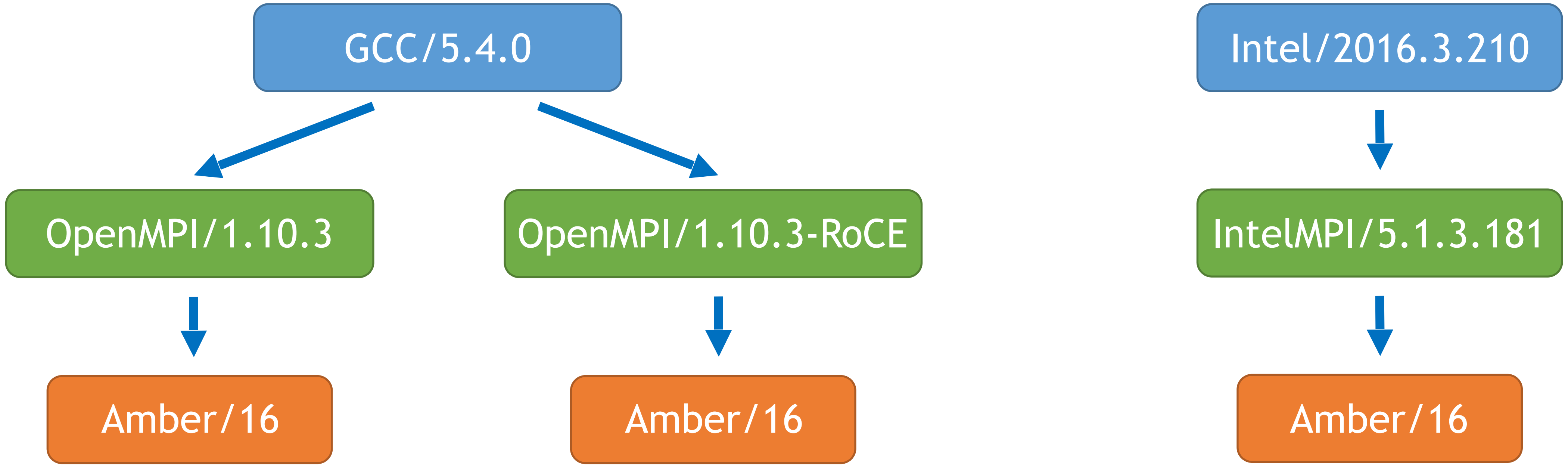
Lmod

Lua-based framework that provides a convenient way to customize user's environment through software modules.

Supports all shells available on the cluster.

SOFTWARE HIERARCHY

Software is organized in a tree structure and displayed accordingly to the loaded dependencies.



LMOD - THE ESSENTIALS

```
module avail <mod>
```

- If no module is passed, print a list of all modules that are available to be loaded.
- If a module is specified, show all available modules with that name.

```
module load mod1 mod2 …
```

- Load the specified modules.

```
module unload mod1 mod2 …
```

- Unload the specified modules.

```
module list
```

- Show all modules loaded in the current environment.

```
module purge
```

- Remove all loaded modules from the environment.

Demo Time

- Show Program Code in editor
- Use Lmod to load GCC and compile
- Compile and run single-cpu program

Submitting Jobs

Submitting Batch Jobs

Creating a Batch Job Script

A **batch job** consists of a sequence of commands listed in a file with the purpose of being interpreted as shell commands.

Shebang

- Specify the script interpreter (Bash)
- Must be the first line!

SLURM directives

- Start with "#SBATCH":
  Parsed by Slurm but ignored by Bash.
- Can be separated by spaces.
- Comments between and after directives are allowed.
- Must be before actual commands!

Script commands

- Commands you want to execute on the compute nodes.

myjob.slurm

```
#!/bin/bash

#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --mem=1G
#SBATCH --time=1-06:30:00
#SBATCH --job-name=myjob
#SBATCH --output=myjob.out


# Just a comment
module load GCC Python
python myscript.py
```

CREATE A BATCH JOB SCRIPT

`--nodes=N`

- Request *N* nodes to be allocated. (*Default*: *N*=1)

`--ntasks=N`

- Request *N* tasks to be allocated. (*Default*: *N*=1)
- Unless otherwise specified, one task maps to one CPU core.

`--mem=NG`

- Request *N* gigabytes of memory per node. (*Default*: *N*=1)

`--time=d-hh:mm:ss`

- Request *d* days, *hh* hours, *mm* minutes and *ss* seconds. (*Default*: 00:15:00)

`--output=<file_name>`

- Write the batch script's standard output in the specified file.
- If not specified the output will be saved in the file: `slurm-<jobid>.out`

Demo Time

- Show batch script for single-cpu job
- Schedule job
- Monitor job and look at results

Choosing your architecture

Choosing Your Architecture

```
--constraint=<arch>
```

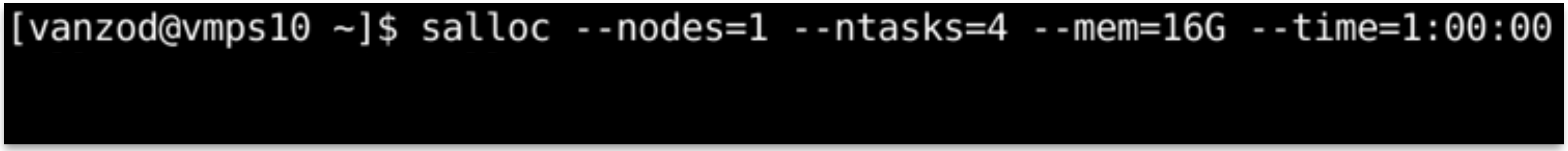
- Currently westmere, sandybridge, haswell, and skylake are available

Compiling on a Compute node

```
salloc options
```

- Obtain job allocation with shell access.
- Accepts all the same *options* previously seen for `sbatch`.

Gateway



```
[vanzod@vmps10 ~]$ salloc --nodes=1 --ntasks=4 --mem=16G --time=1:00:00
```

Recommended for compiling, debugging and benchmarking sessions.

Demo Time

- Use Salloc to run on a skylake node
- Compile program for skylake architecture
- Try enabling avx512 instructions
- Compare results to un-optimized

Using MPI

Demo Time

- Show mpi version of program in editor
- Load OpenMPI with LMod
- Run MPI program with 2, 4, 8 tasks interactively
- Compare results to single-cpu job

CREATE A BATCH JOB SCRIPT FOR MPI

ACCRE
Advanced Computing Center for Research & Education

`--nodes=`*`N`*

- Request *N* nodes to be allocated. (*Default*: *N*=1)

`--ntasks=`*`N`*

- Request *N* tasks to be allocated. (*Default*: *N*=1)
- Unless otherwise specified, one task maps to one CPU core.

`–tasks-per-node=`*`N`*

- Request specifically *N* tasks to run on each node.

`–cpus-per-task=`*`N`*

- Request *N* cpu cores to be allocated to each task. (*Default*: *N*=1)

Demo Time

- Show mpi version slurm batch script
- Launch a scheduled MPI job with 48+ tasks
- Launch a scheduled MPI job with 8 tasks on each node
- Compare results