The University HPC Cluster: *Your Playground for Understanding Parallel Computer Architectures*

Kim F. Wong
Center for Research Computing

ECE2166_2019S

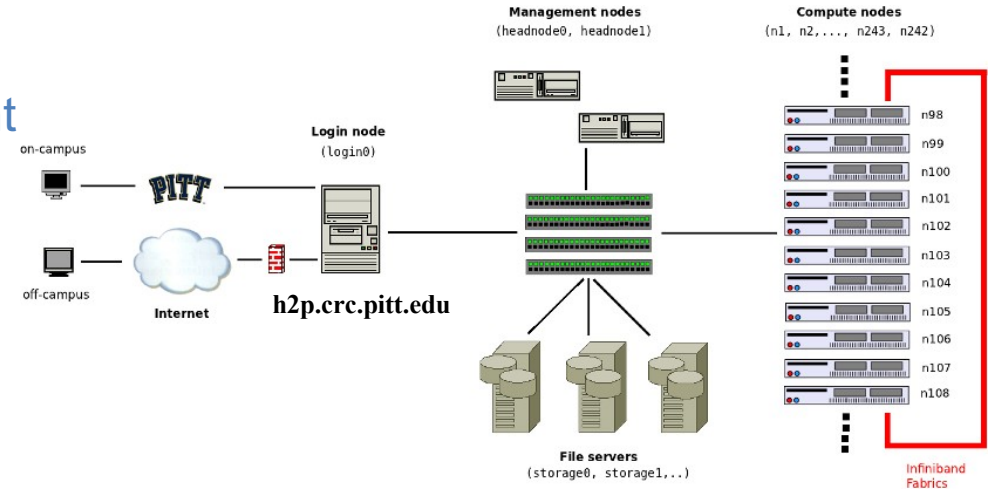
What is High Performance Computing?

Here are some characteristics of a HPC cluster

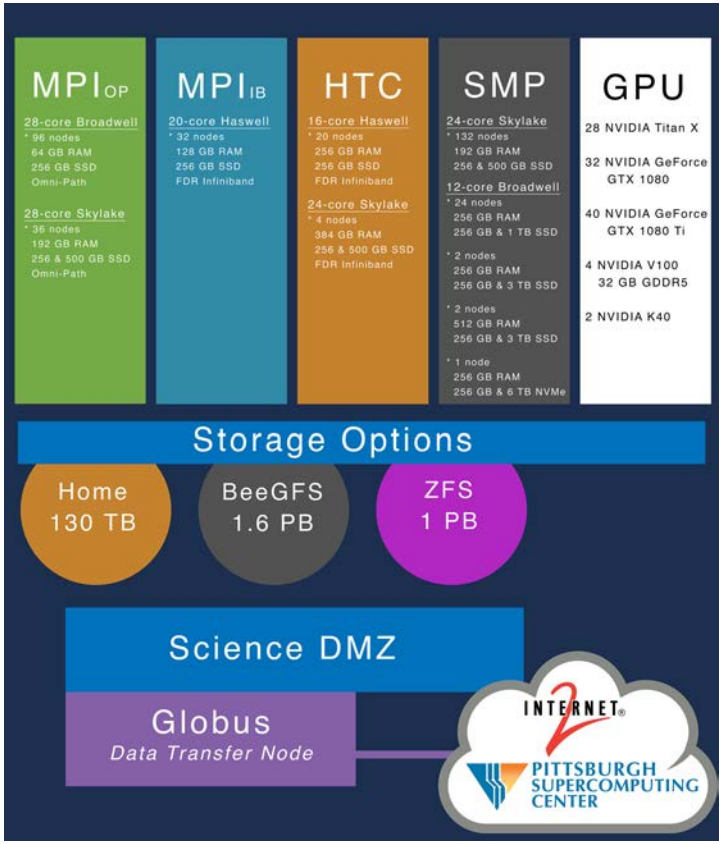
- Comprised of a network of computers that can be harnessed simultaneously to tackle a problem requiring large processing power
- Compute noes within a cluster are connected by a specialized high-speed interconnect or a standard GigE network
- HPC resources are shared among a community of users
- Users request resources through some kind of management software

Motivations for building a HPC cluster

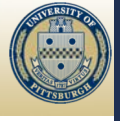
- Necessary for solving grand challenge problems … problems that require an enormous amount of computation or memory
- Provides more cost-effective usage of shared computational resources towards common objectives



What do we have at Pitt?



Pitt has racks of compute nodes and disk storage housed inside an environmentally controlled data center. The various clusters were configured specifically to support different work flows from different research communities.



MPI$_{OP}$

28-core Broadwell
* 96 nodes
  64 GB RAM
  256 GB SSD
  Omni-Path

28-core Skylake
* 36 nodes
  192 GB RAM
  256 & 500 GB SSD
  Omni-Path

MPI$_{IB}$

20-core Haswell
* 32 nodes
  128 GB RAM
  256 GB SSD
  FDR Infiniband

HTC

16-core Haswell
* 20 nodes
  256 GB RAM
  256 GB SSD
  FDR Infiniband

24-core Skylake
* 4 nodes
  384 GB RAM
  256 & 500 GB SSD
  FDR Infiniband

SMP

24-core Skylake
* 132 nodes
  192 GB RAM
  256 & 500 GB SSD

12-core Broadwell
* 24 nodes
  256 GB RAM
  256 GB & 1 TB SSD

* 2 nodes
  256 GB RAM
  256 GB & 3 TB SSD

* 2 nodes
  512 GB RAM
  256 GB & 3 TB SSD

* 1 node
  256 GB RAM
  256 GB & 6 TB NVMe

GPU

28 NVIDIA Titan X

32 NVIDIA GeForce GTX 1080

40 NVIDIA GeForce GTX 1080 Ti

4 NVIDIA V100 32 GB GDDR5

2 NVIDIA K40

Storage Options

Home
130 TB

BeeGFS
1.6 PB

ZFS
1 PB

Science DMZ

How does Pitt compare to national centers?

Oak Ridge Leadership Computing Facility

PEAK PERFORMANCE
20+
PETAFLOPS

299,008
OPTERON CORES

COMPUTE NODES
18,688

32GB + 6GB
Memory Per Node

NVIDIA TESLA
K20 GPU ACCELERATORS
18,688
GPUs

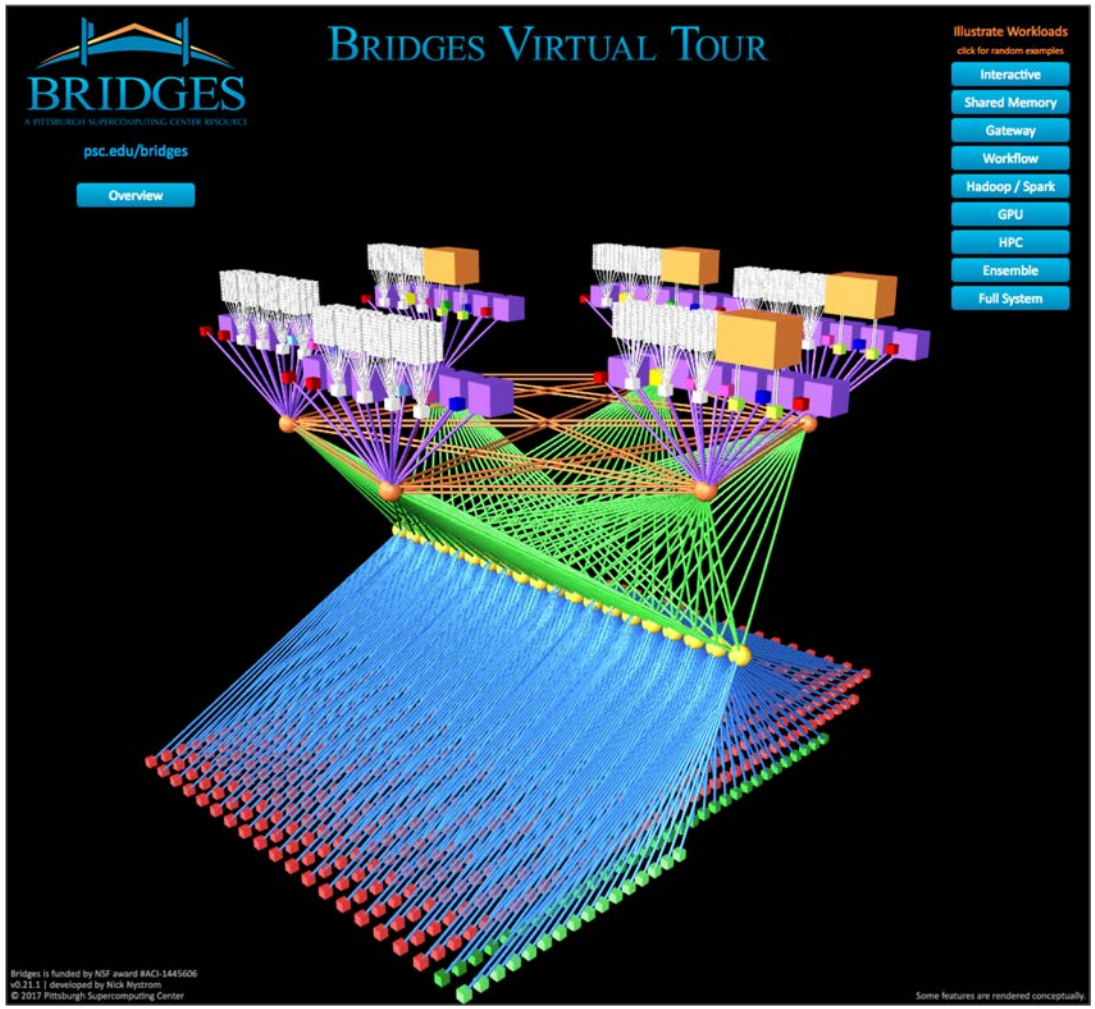
TOTAL SYSTEM MEMORY
710
TERABYTES

GEMINI
INTERCONNECT

4,352 sqft
FLOOR SPACE

Pitt has similar bleeding-edge technologies but on a smaller scale. The knowledge you learn here transfers to other potential places of employment when you graduate.

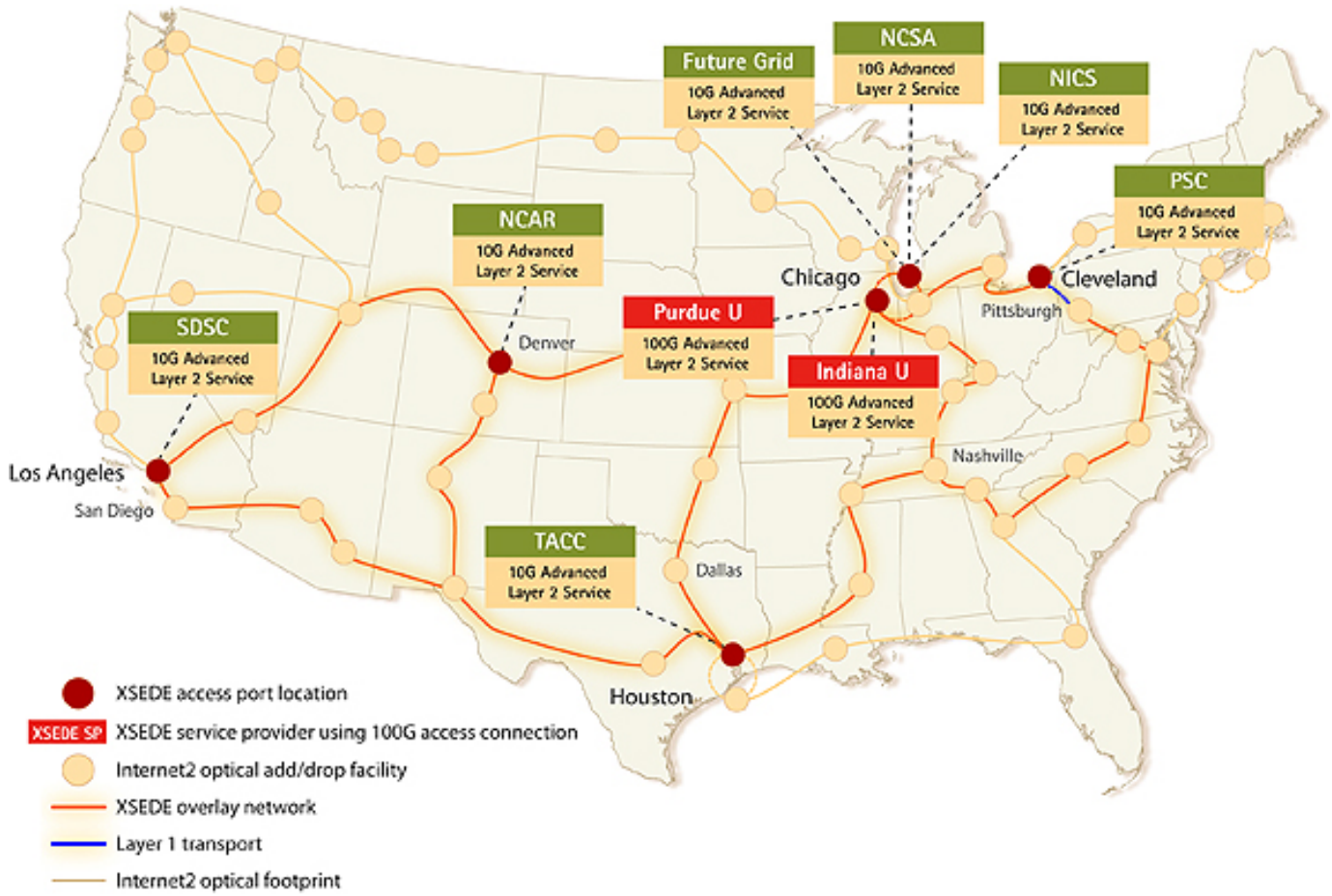
Pittsburgh Supercomputing Center



Bridges

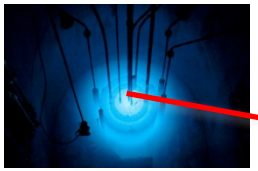
- Funded by $17.2M from NSF
- An XSEDE resource
- 4 nodes with 12 TB RAM
- 42 nodes with 3TB RAM
- 800 nodes with 128 GB RAM
- Nodes with NVIDIA K80 GPUs
- Nodes with NVIDIA V100 GPUs
- Nodes connected by Omni-Path fabric

XSEDE: National Supercomputing Centers



Extreme Science and Engineering Discovery Environment

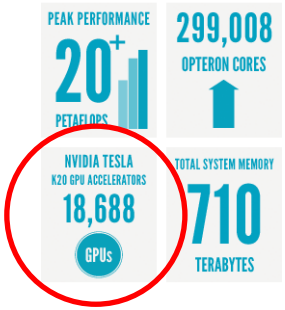
Two eras in computing



The Ever-changing Face of HPC

Big data analytics, machine learning, and AI are reshaping HPC architectures

ORNL Titan Configuration



PSC Bridges Configuration

- 16 nodes with 2 Tesla K80 GPUs
- 32 nodes with 2 Tesla P100 GPUs
- 9 nodes with 8 V100 GPUs
- NVIDIA DGX-2 with 16 V100 GPUs

NVIDIA V100 GPU Architecture

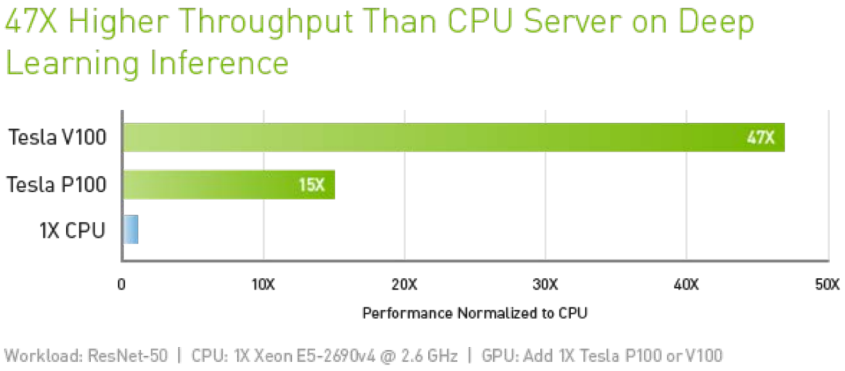
- Each GPU has 80 SM (Streaming Multiprocessor)
- Each SM has 64 FP32 cores
- → Each GPU has 5120 cores! These are single precision cores. Double precision will be half of this but that is like 2560 brains!

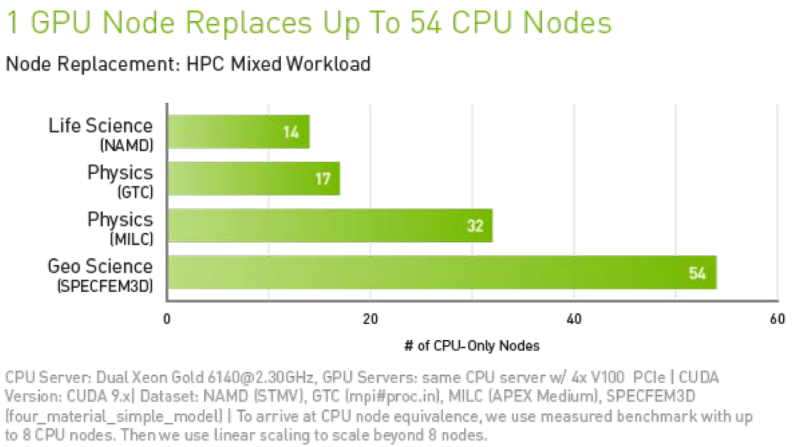
But you need to write code to leverage all cores

Again there is no free lunch. However, new tools have made the programming easier.

- NVIDIA CUDA Toolkit
- OpenACC directive-based programming for accelerators
- OpenMP v4 directive-based programming for accelerators

Leveraging GPU acceleration, your code can see speedup of 10X.

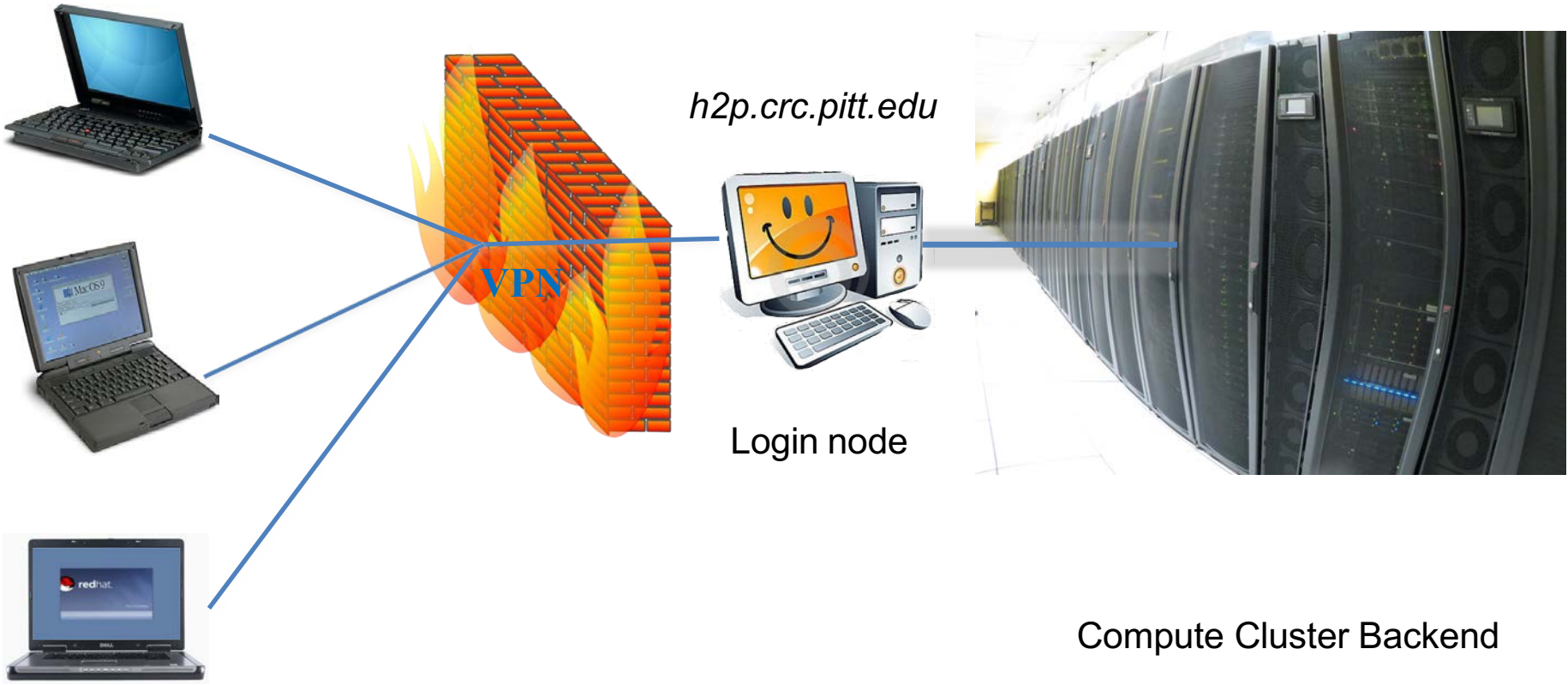




How do you access the resources remotely?

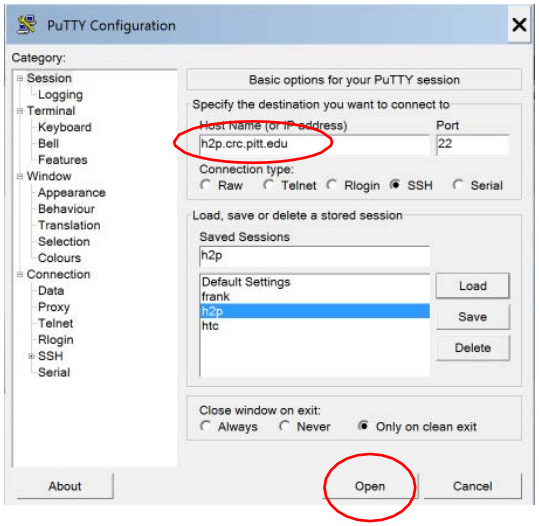
Since h2p.crc.pitt.edu is firewalled, you will need to establish a Virtual Private Network (VPN) before connecting to the cluster. Pitt also uses Multifactor Authentication. Setup instructions are provided in these links:

- http://technology.pitt.edu/help-desk/how-to-documents/secure-remote-access-connect-pulse-secure-client
- http://technology.pitt.edu/services/multifactor-authentication-pitt



Windows: Accessing the HPC Cluster

Access to h2p is enabled via a secure shell (SSH) connection to the cluster.



A SSH client called PuTTY is available for Windows

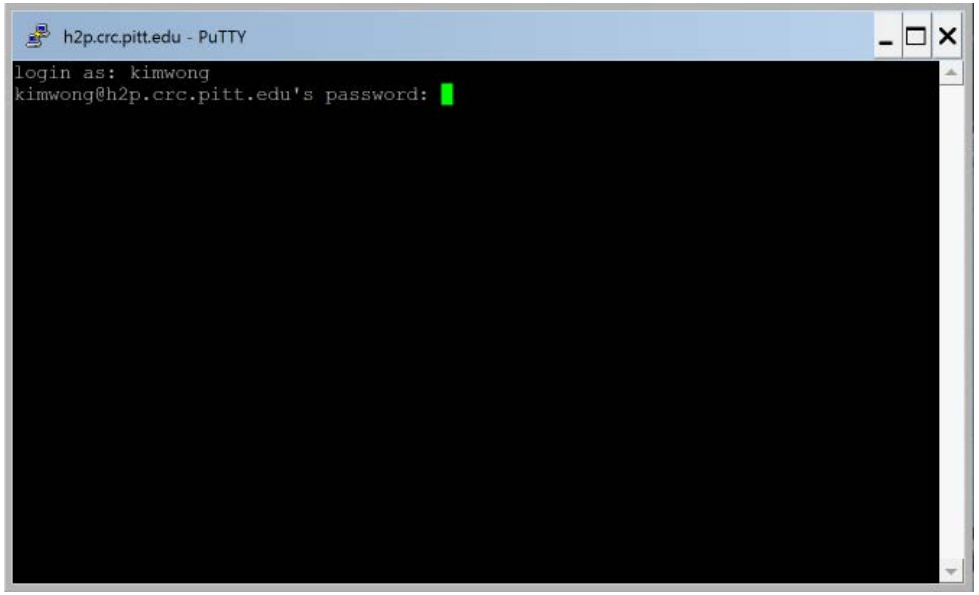
Download from here, https://www.putty.org/

Specify these connection properties:

- Hostname: h2p.crc.pitt.edu
- Port: 22
- Connection type: SSH

Clicking the Open button will *open a SSH terminal*

- login as: <Pitt Username>
- password: <my.pitt password>



Mac: Accessing the HPC Cluster

macOS has a built in terminal client that can be used to ssh to the cluster. In the finder toolbar, click on Go → Utilities → Terminal.

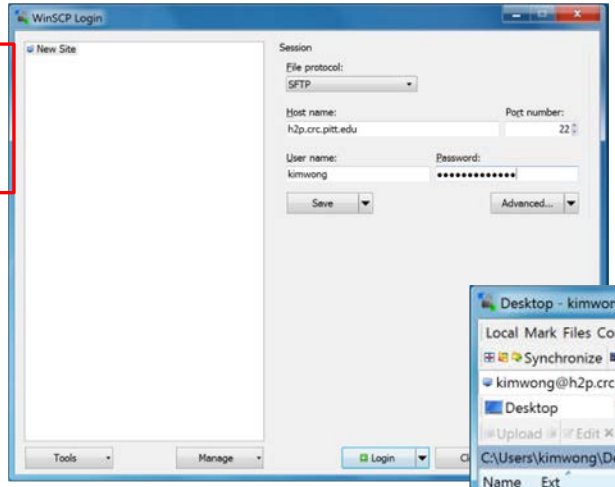
Type `ssh -X <username>@h2p.crc.pitt.edu` within a terminal

Windows: transferring files?

If transferring from off-campus, a VPN session is required.

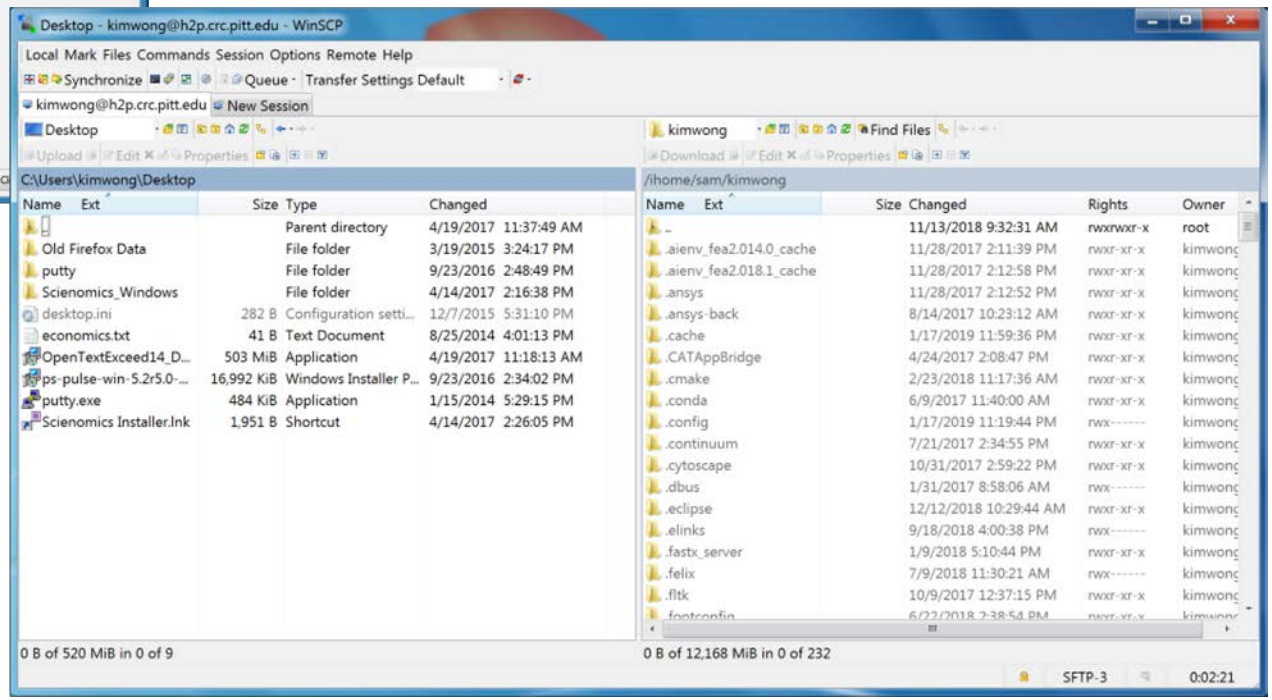
For Windows, use WinSCP https://sourceforge.net/projects/winscp/.
Login in to h2p using your Pitt credentials.

1



2

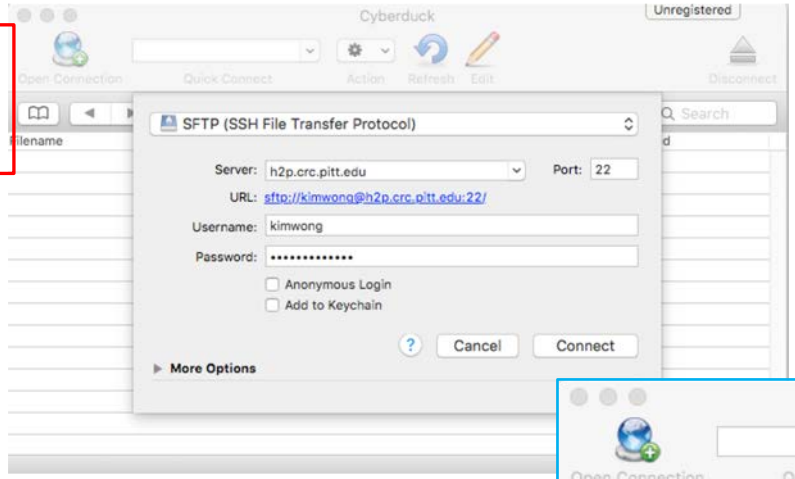
drag and drop
between panels



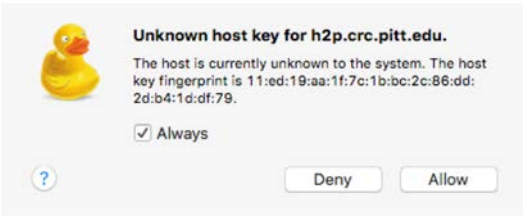
macOS: transferring files

For Macs, I have heard that Cyberduck works well: https://cyberduck.io/. Select SFTP (SSH File Transfer Protocol).

1

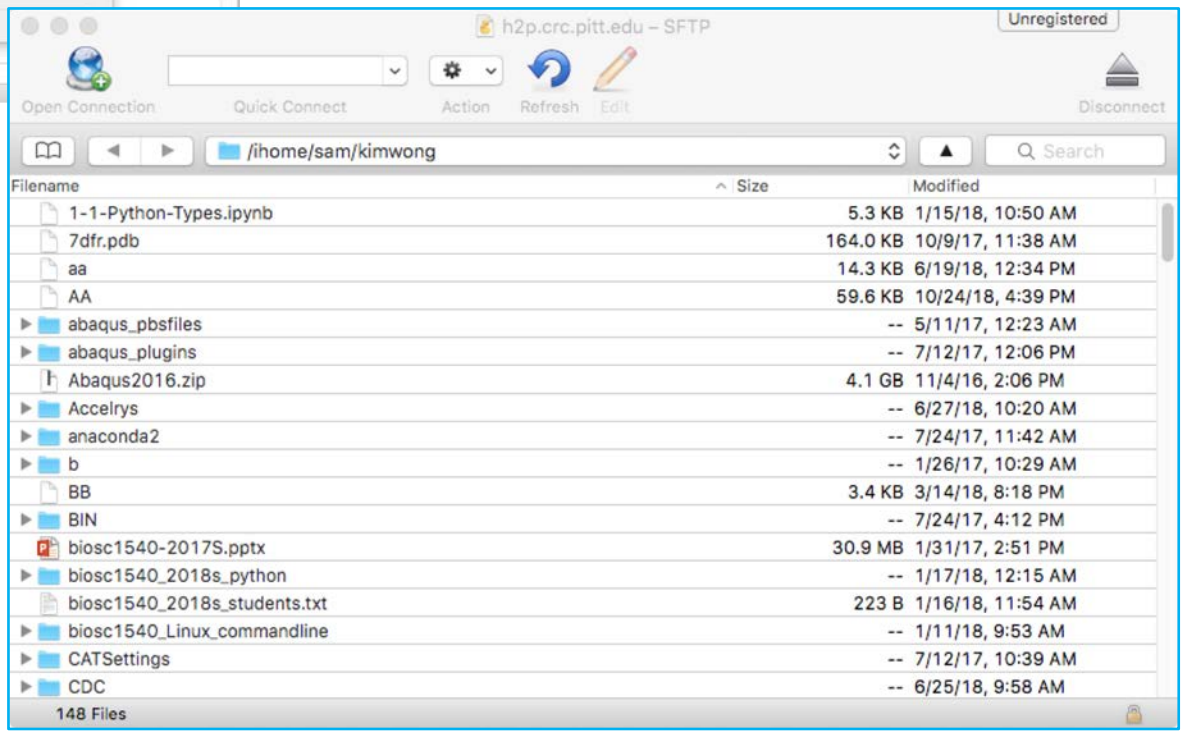


2



3

After authentication, a new window shows up. Drag and drop between that window and your local desktop/folders.



Cluster User Support

Main documentation

https://crc.pitt.edu/h2p

Requesting help via online ticketing system (need to log in using Pitt credentials)

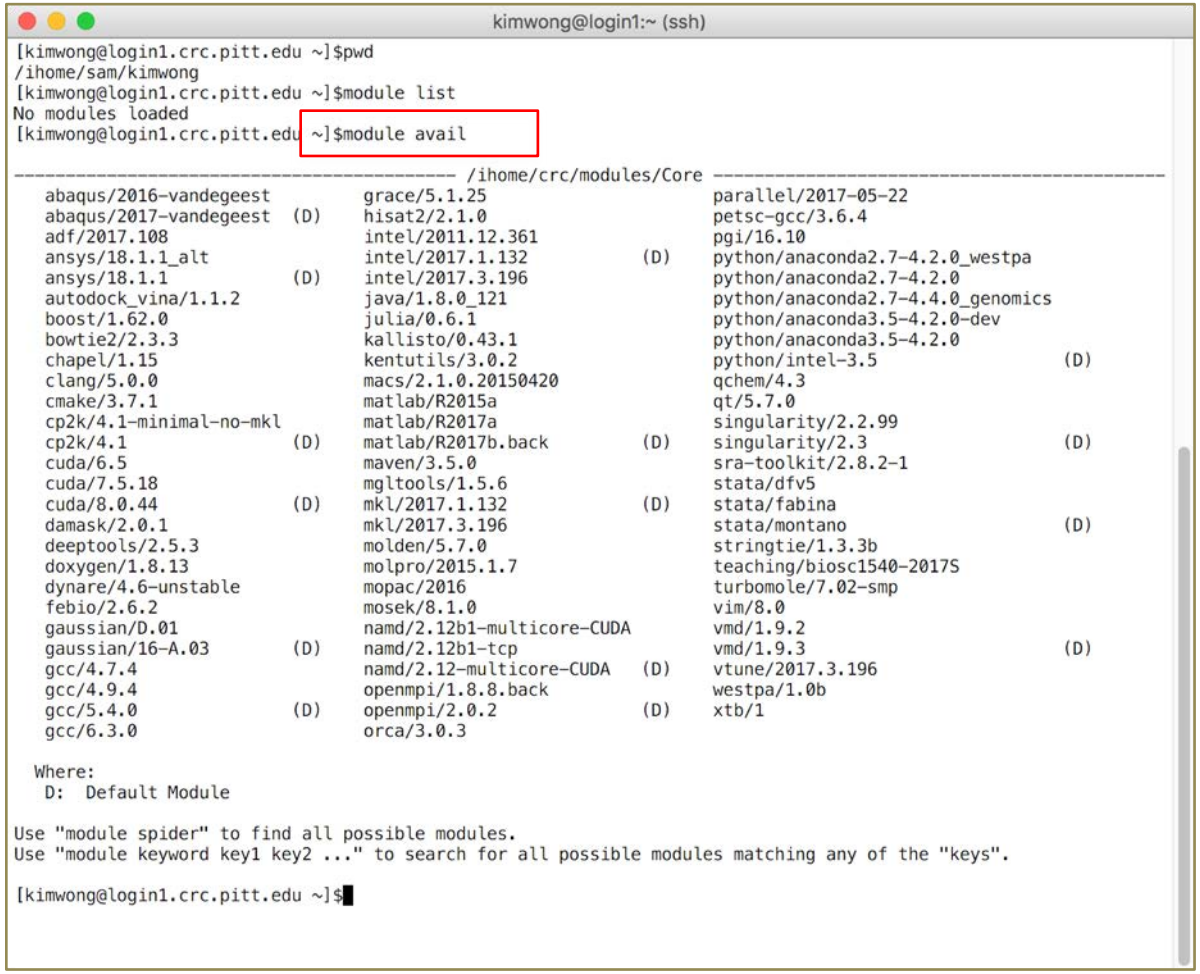
https://crc.pitt.edu/tickets

Physical office location

312 Schenley Place
4420 Bayard Street

LMOD Environment Modules (1/5)

`module avail` shows the core packages.



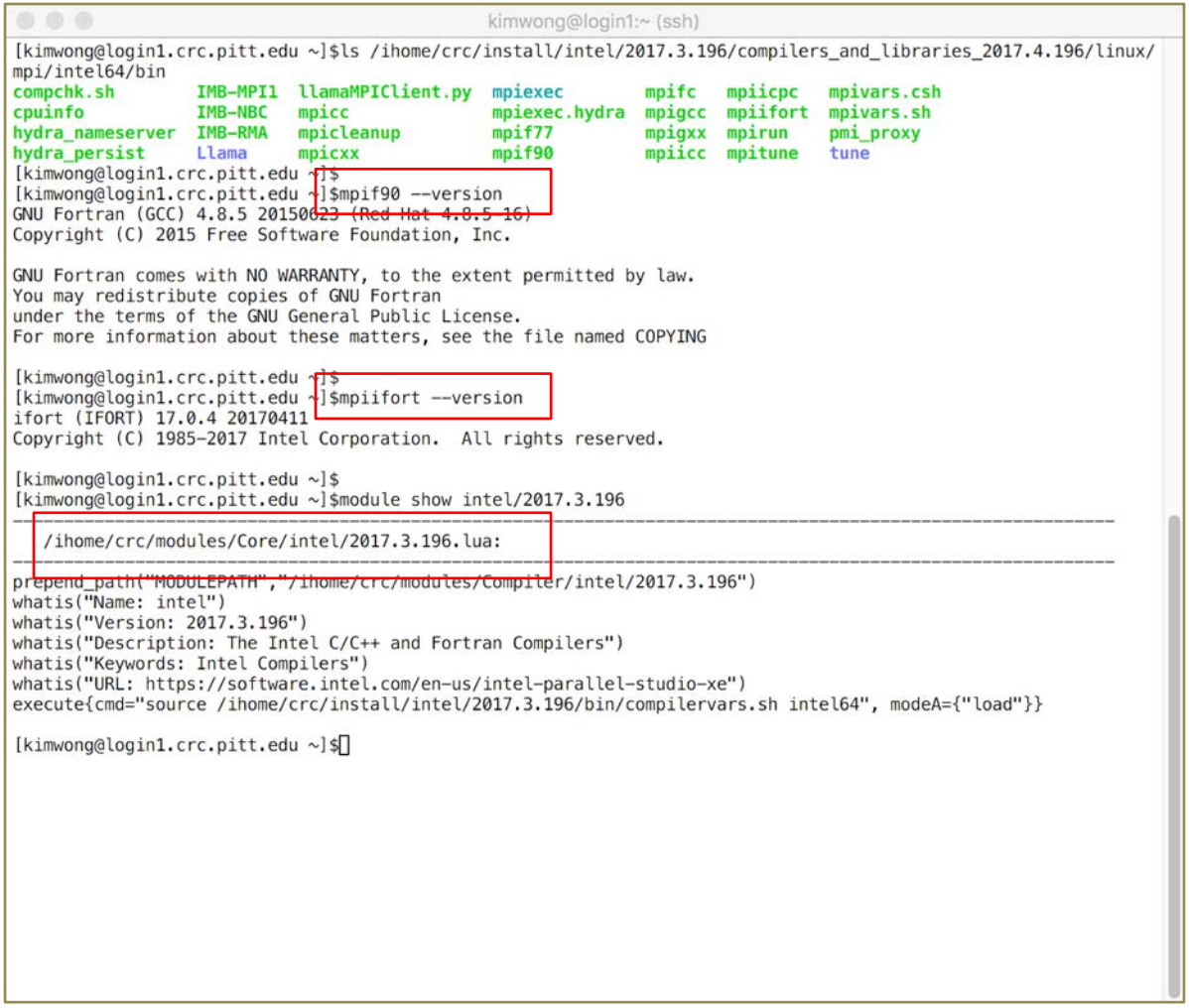
LMOD Environment Modules (2/5)

`module spider <package>` describes how to load the package, including dependencies.

```
kimwong@login1:~ (ssh)
[kimwong@login1.crc.pitt.edu ~]$module spider intel

---------------------------------------------------------------------------------------------------------------
  intel:
---------------------------------------------------------------------------------------------------------------
    Description:
      The Intel C/C++ and Fortran Compilers

     Versions:
        intel/2011.12.361
        intel/2017.1.132
        intel/2017.3.196

     Other possible modules matches:
        intel-mpi

---------------------------------------------------------------------------------------------------------------
  To find other possible module matches do:
      module -r spider '.*intel.*'

---------------------------------------------------------------------------------------------------------------
  For detailed information about a specific "intel" module (including how to load the modules) use the module's
full name.
  For example:

     $ module spider intel/2017.3.196
---------------------------------------------------------------------------------------------------------------

[kimwong@login1.crc.pitt.edu ~]$which ifort
/usr/bin/which: no ifort in (/ihome/sam/kimwong/TEMP18/miniconda2/bin:/ihome/sam/kimwong/TEMP17/miniconda2/bin:.
:/home/sam/kimwong/BIN/grace/bin:/home/sam/kimwong/BIN:/usr/lib64/qt-3.3/bin:/usr/local/bin:/usr/bin:/usr/local/
sbin:/usr/sbin:/ihome/crc/wrappers:/ihome/sam/kimwong/bin)
[kimwong@login1.crc.pitt.edu ~]$
[kimwong@login1.crc.pitt.edu ~]$module load intel/2017.3.196
[kimwong@login1.crc.pitt.edu ~]$which ifort
/ihome/crc/install/intel/2017.3.196/compilers_and_libraries_2017.4.196/linux/bin/intel64/ifort
[kimwong@login1.crc.pitt.edu ~]$
[kimwong@login1.crc.pitt.edu ~]$which mpif90
/ihome/crc/install/intel/2017.3.196/compilers_and_libraries_2017.4.196/linux/mpi/intel64/bin/mpif90
[kimwong@login1.crc.pitt.edu ~]$
[kimwong@login1.crc.pitt.edu ~]$ls /ihome/crc/install/intel/2017.3.196/compilers_and_libraries_2017.4.196/linux/
mpi/intel64/bin
compchk.sh        IMB-MPI1  llamaMPIClient.py  mpiexec        mpifc   mpiicpc   mpivars.csh
cpuinfo           IMB-NBC   mpicc              mpiexec.hydra  mpigcc  mpiifort  mpivars.sh
hydra_nameserver  IMB-RMA   mpicleanup         mpif77         mpigxx  mpirun    pmi_proxy
hydra_persist     Llama     mpicxx             mpif90         mpiicc  mpitune   tune
[kimwong@login1.crc.pitt.edu ~]$
```

LMOD Environment Modules (3/5)

In the newer versions of the Intel Parallel Studio, the `mpif90` wrapper scripts points to GNU fortran. Use `mpiifort`.



LMOD Environment Modules (4/5)

After loading a package, use `module show <package>` to locate the lua file that contains info about how the environment variables are defined.

```
kimwong@login1:~ (ssh)
[kimwong@login1.crc.pitt.edu ~]$module show intel/2017.3.196
-------------------------------------------------------------------------------------------------------------------
   /ihome/crc/modules/Core/intel/2017.3.196.lua:
-------------------------------------------------------------------------------------------------------------------
prepend_path("MODULEPATH","/ihome/crc/modules/Compiler/intel/2017.3.196")
whatis("Name: intel")
whatis("Version: 2017.3.196")
whatis("Description: The Intel C/C++ and Fortran Compilers")
whatis("Keywords: Intel Compilers")
whatis("URL: https://software.intel.com/en-us/intel-parallel-studio-xe")
execute{cmd="source /ihome/crc/install/intel/2017.3.196/bin/compilervars.sh intel64", modeA={"load"}}

[kimwong@login1.crc.pitt.edu ~]$
[kimwong@login1.crc.pitt.edu ~]$cat /ihome/crc/modules/Core/intel/2017.3.196.lua
-- Setup module path for packages built with this compiler
local mdir = "/ihome/crc/modules/Compiler/intel/2017.3.196"
prepend_path("MODULEPATH", mdir)

-- Description
whatis("Name: "..myModuleName())
whatis("Version: 2017.3.196")
whatis("Description: The Intel C/C++ and Fortran Compilers")
whatis("Keywords: Intel Compilers")
whatis("URL: https://software.intel.com/en-us/intel-parallel-studio-xe")

-- On load, run compilervars.sh script
--if (os.getenv("INTEL_PYTHONHOME") == nil) then
    execute{cmd='source /ihome/crc/install/intel/2017.3.196/bin/compilervars.sh intel64', modeA={"load"}}
--end

-- Licenses for new intel distributions need to be in:
-- /ihome/crc/install/intel/2017.1.132/compilers_and_libraries_2017.1.132/linux/licenses
[kimwong@login1.crc.pitt.edu ~]$
[kimwong@login1.crc.pitt.edu ~]$
```

LMOD Environment Modules (5/5)

Here is an example of a package that depends on another package.

```
[kimwong@login1.crc.pitt.edu ~]$module spider intel-mpi

----------------------------------------------------------------------------------------------------------------
  intel-mpi:
----------------------------------------------------------------------------------------------------------------
    Description:
      The Intel Message Passing Interface Library

     Versions:
        intel-mpi/2017.1.132
        intel-mpi/2017.3.196

----------------------------------------------------------------------------------------------------------------
  For detailed information about a specific "intel-mpi" module (including how to load the modules) use the modul
e's full name.
  For example:

     $ module spider intel-mpi/2017.3.196
----------------------------------------------------------------------------------------------------------------

[kimwong@login1.crc.pitt.edu ~]$module load intel-mpi/2017.3.196
Lmod has detected the following error:  These module(s) exist but cannot be loaded as requested:
"intel-mpi/2017.3.196"

   Try: "module spider intel-mpi/2017.3.196" to see how to load the module(s).



[kimwong@login1.crc.pitt.edu ~]$module spider intel-mpi/2017.3.196

----------------------------------------------------------------------------------------------------------------
  intel-mpi: intel-mpi/2017.3.196
----------------------------------------------------------------------------------------------------------------
    Description:
      The Intel Message Passing Interface Library

    You will need to load all module(s) on any one of the lines below before the "intel-mpi/2017.3.196" module i
s available to load.

      intel/2017.3.196


[kimwong@login1.crc.pitt.edu ~]$module load intel/2017.3.196 intel-mpi/2017.3.196
[kimwong@login1.crc.pitt.edu ~]$
```

SLURM Queueing System (1/5)

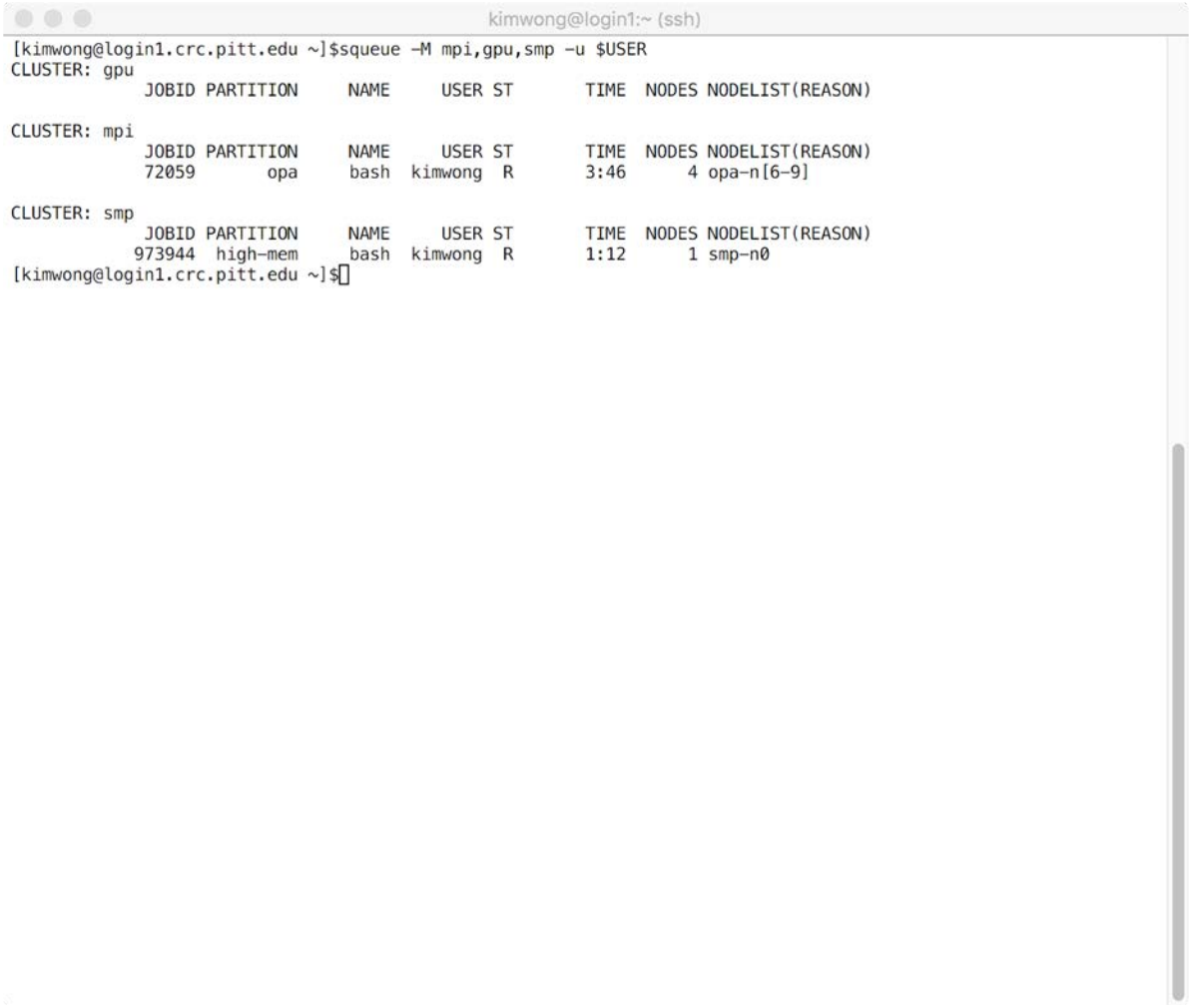
Use `sinfo` to get status of clusters.

kimwong@login1:~ (ssh)

```
[kimwong@login1.crc.pitt.edu ~]$sinfo -M mpi,gpu,smp
CLUSTER: gpu
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
gtx1080*     up   infinite      8    mix gpu-stage[08-15]
titanx       up   infinite      6    mix gpu-stage[01,03-07]
titanx       up   infinite      1  alloc gpu-stage02
k40          up   infinite      1   idle smpgpu-n0
titan        up   infinite      4  down* legacy-n[127-130]
titan        up   infinite      2   idle legacy-n[126,131]

CLUSTER: mpi
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
opa*         up   infinite      1   resv opa-n0
opa*         up   infinite     70  alloc opa-n[1,4-5,11-24,31-54,61-84,91-95]
opa*         up   infinite     25   idle opa-n[2-3,6-10,25-30,55-60,85-90]
ib           up   infinite     32  alloc ib-n[0-31]
legacy       up   infinite     20  down* legacy-n[20,26-29,33,43,57,62-63,66,68-69,71,75,77,82-85]
legacy       up   infinite      6  alloc legacy-n[44-49]
legacy       up   infinite     62   idle legacy-n[0-19,21-23,30-32,34-42,50-56,58-61,64-65,67,70,72-74,76,78-81,
86-89]
compbio      up   infinite      2  down* legacy-n[95,98]
compbio      up   infinite      1  drain legacy-n105
compbio      up   infinite      5    mix legacy-n[90,92,96-97,112]
compbio      up   infinite      9  alloc legacy-n[93-94,99-100,102-104,106-107]
compbio      up   infinite      6   idle legacy-n[109,111,113-114,116-117]

CLUSTER: smp
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
smp*         up   infinite     17    mix smp-n[30,36,38,44,60,76,80-81,84-85,102-103,105-106,116-117,120]
smp*         up   infinite     83  alloc smp-n[24-29,31-35,37,39-43,45-59,61-75,77-79,82-83,86-101,104,107-115,1
18-119,121-123]
high-mem     up   infinite      4  alloc smp-512-n[1-2],smp-n[3-4]
high-mem     up   infinite     25   idle smp-256-n[1-2],smp-n[0-2,5-23],smp-nvme-n1
[kimwong@login1.crc.pitt.edu ~]$
```

SLURM Queueing System (2/5)

Use `squeue` to get status of jobs on clusters.



SLURM Queueing System (3/5)

Use `crc-interactive.py` to get an interactive session on a cluster.

```
kimwong@opa-n25:~ (ssh)
[kimwong@login1.crc.pitt.edu ~]$crc-interactive.py -m -n 4 -c 28 -t 3
srun: job 72062 queued and waiting for resources
srun: job 72062 has been allocated resources
[kimwong@opa-n25.sam.pitt.edu ~]$
[kimwong@opa-n25.sam.pitt.edu ~]$scontrol show jobID=72062
JobId=72062 JobName=bash
   UserId=kimwong(15083) GroupId=sam(16036) MCS_label=N/A
   Priority=3059 Nice=0 Account=sam QOS=normal
   JobState=RUNNING Reason=None Dependency=(null)
   Requeue=1 Restarts=0 BatchFlag=0 Reboot=0 ExitCode=0:0
   RunTime=00:00:32 TimeLimit=03:00:00 TimeMin=N/A
   SubmitTime=2018-01-17T12:38:24 EligibleTime=2018-01-17T12:38:24
   StartTime=2018-01-17T12:38:37 EndTime=2018-01-17T15:38:37 Deadline=N/A
   PreemptTime=None SuspendTime=None SecsPreSuspend=0
   Partition=opa AllocNode:Sid=mpi-interactive:15832
   ReqNodeList=(null) ExcNodeList=(null)
   NodeList=opa-n[25-28]
   BatchHost=opa-n25
   NumNodes=4 NumCPUs=112 NumTasks=112 CPUs/Task=1 ReqB:S:C:T=0:0:*:*
   TRES=cpu=112,mem=4G,node=4
   Socks/Node=* NtasksPerN:B:S:C=28:0:*:* CoreSpec=*
   MinCPUsNode=28 MinMemoryNode=1024M MinTmpDiskNode=0
   Features=(null) Gres=(null) Reservation=(null)
   OverSubscribe=OK Contiguous=0 Licenses=(null) Network=(null)
   Command=bash
   WorkDir=/ihome/sam/kimwong
   Power=

[kimwong@opa-n25.sam.pitt.edu ~]$scontrol show node opa-n25
NodeName=opa-n25 Arch=x86_64 CoresPerSocket=14
   CPUAlloc=28 CPUErr=0 CPUTot=28 CPULoad=0.01
   AvailableFeatures=(null)
   ActiveFeatures=(null)
   Gres=(null)
   NodeAddr=opa-n25 NodeHostName=opa-n25 Version=16.05
   OS=Linux RealMemory=64308 AllocMem=1024 FreeMem=0 Sockets=2 Boards=1
   State=ALLOCATED ThreadsPerCore=1 TmpDisk=0 Weight=1 Owner=N/A MCS_label=N/A
   BootTime=2017-09-19T08:09:42 SlurmdStartTime=2017-09-19T08:12:24
   CapWatts=n/a
   CurrentWatts=0 LowestJoules=0 ConsumedJoules=0
   ExtSensorsJoules=n/s ExtSensorsWatts=0 ExtSensorsTemp=n/s


[kimwong@opa-n25.sam.pitt.edu ~]$
```

SLURM Queueing System (4/5)

Use `sbatch <slurm_job_script>` to submit job to queue.

kimwong@opa-n25:/ihome/crc/how_to_run/amber/mocvnhlysm_4N.28C (ssh)

```
#!/bin/bash
#SBATCH --nodes=4
#SBATCH --ntasks-per-node=28
#SBATCH --cluster=mpi
#SBATCH --partition=opa
#SBATCH --time=1:00:00
#SBATCH --job-name=mocv

# Load Modules
module load intel/2017.1.132
module load amber/16

# Run over Omni-Path fabric
export I_MPI_FABRICS_LIST=tmi
export I_MPI_FALLBACK=0

# Amber input files and output name
INP=md.in
TOP=mocvnhlysm.top
CRD=mocvnhlysm.crd
OUT=mocvnhlysm

# Executable
SANDER=pmemd.MPI

# Launch MPI
mpirun -n $SLURM_NTASKS \
        $SANDER  -O     -i   $INP   -p   $TOP   -c   $CRD   -r   $OUT.rst \
                        -o   $OUT.out   -e   $OUT.ene   -v   $OUT.vel   -inf $OUT.nfo   -x   $OUT.mdcrd
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"amber.slurm" 29L, 657C
```

SLURM directives

load modules

Set communication interface

domain-specific execution line

Example job submission scripts: `/ihome/crc/how_to_run/`

SLURM Queueing System (5/5)

Use `scancel -M <cluster> <job ID>` to delete a job.

```
kimwong@login1:~ (ssh)
[kimwong@login1.crc.pitt.edu ~]$squeue -M mpi -u $USER
CLUSTER: mpi
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
            255555       opa     mocv  kimwong  R       0:30      4 opa-n[17,93-95]
[kimwong@login1.crc.pitt.edu ~]$
[kimwong@login1.crc.pitt.edu ~]$scancel -M mpi 255555
[kimwong@login1.crc.pitt.edu ~]$
[kimwong@login1.crc.pitt.edu ~]$scancel -M mpi 255555
[kimwong@login1.crc.pitt.edu ~]$squeue -M mpi -u $USER
CLUSTER: mpi
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
[kimwong@login1.crc.pitt.edu ~]$clear
[kimwong@login1.crc.pitt.edu ~]$squeue -M mpi -u $USER
CLUSTER: mpi
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
            255560       opa     mocv  kimwong  R       0:07      4 opa-n[17,93-95]
[kimwong@login1.crc.pitt.edu ~]$
[kimwong@login1.crc.pitt.edu ~]$scancel -M mpi 255560
[kimwong@login1.crc.pitt.edu ~]$
[kimwong@login1.crc.pitt.edu ~]$squeue -M mpi -u $USER
CLUSTER: mpi
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
            255560       opa     mocv  kimwong CG       0:22      1 opa-n17
[kimwong@login1.crc.pitt.edu ~]$
[kimwong@login1.crc.pitt.edu ~]$squeue -M mpi -u $USER
CLUSTER: mpi
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
[kimwong@login1.crc.pitt.edu ~]$
[kimwong@login1.crc.pitt.edu ~]$
[kimwong@login1.crc.pitt.edu ~]$crc-usage.pl ece2166_2019s
================================================================================
---------------------- H2P Service Unit Usage ----------------------
================================================================================
--------------------------------------------------------------------------------
Account:                                                    ece2166_2019s
Total SUs:                                                         100000
Proposal End:                                                    01/14/20
--------------------------------------------------------------------------------
Cluster:                                                             smp
--------------------------------------------------------------------------------
                  User        SUs (CPU Hours)       Percent of Total
------------------------ ---------------------- ----------------------
         Cluster Total                       0                 0.0000
                 abv18                       0                 0.0000
                 amj92                       0                 0.0000
                amp318                       0                 0.0000
                brs151                       0                 0.0000
                cjs188                       0                 0.0000
                cmc183                       0                 0.0000
                dal181                       0                 0.0000
                 dec71                       0                 0.0000
                emb153                       0                 0.0000
                 ewg13                       0                 0.0000
                  gbw6                       0                 0.0000
                jal234                       0                 0.0000
                jam425                       0                 0.0000
                 jsg59                       0                 0.0000
                 mwm29                       0                 0.0000
                 nds40                       0                 0.0000
                 nep36                       0                 0.0000
                nis102                       0                 0.0000
                ram203                       0                 0.0000
                 rub26                       0                 0.0000
                 sdp38                       0                 0.0000
```

Recap of Key Points (1/2)

- To access h2p.crc.pitt.edu
  1. You need to establish VPN (slide 11)
  2. Use ssh to connect to remote machine (slides 12-13)
  3. Use WinSCP or Cyberduck to transfer files (slides 14-15)

- To use GNU Compilers
  1. Load the modules by executing the following on the commandline

     ```
     module purge
     module load gcc/5.4.0
     module load openmpi/3.0.0
     ```

  2. Use the MPI compiler wrappers to build code: `mpicc, mpicxx`
     Example: `mpicc hello_world.c -o hello_world.x`

- To use Intel Compilers
  1. Load the modules by executing the following on the commandline

     ```
     module purge
     module load intel/2017.3.196
     ```

  2. Use the MPI compiler wrappers to build code: `mpiicc, mpiicxx`
     Example: `mpiicc hello_world.c -o hello_world.x`

Recap of Key Points (2/2)

- Examples for running MPI and OpenMP jobs
  `/ihome/ece2166_2019s/h2p_examples`

  1. `mpi/amber`: The two directories here contain an example job submission script (`amber.slurm`) for running the molecular dynamics code. The `_IMPI` directory runs the package that was built using Intel MPI and the `_OMPI` directory runs the package that was built using OpenMPI. To submit the job, execute `sbatch amber.slurm` (see slide 25).
  2. `mpi/hello_world`: This directory contains build scripts (`compile-gnu.sh`, `compile-intel.sh`) for compiling a simple MPI hello world program. You can simply execute these build scripts on the commandline. The job submission scripts are `hello-gnu.slurm` and `hello-intel.slurm`. Be sure to take a look at these files to see examples for compiling and examples for crafting a job submission script.
  3. `openmp/hello_world`: This directory contains build scripts (`gnu-parallel.sh`, `intel-parallel.sh`) for compiling a simple OpenMP hello world program. The job submission scripts are `hello-gnu.slurm` and `hello-intel.slurm`. Be sure to inspect these files to see how to build OpenMP code and for setting the environment variable for controlling the number of threads to run in parallel.