Quick Summary

Center for Research Computing
University of Pittsburgh

Access

- Request an Account

`http://core.sam.pitt.edu/apply`

- Login

HTC ..... `htc.sam.pitt.edu`
SMP/OPA/NTA ..... `h2p.crc.pitt.edu`

- Storage

ihome (75G per user) ..... `/ihome`
mobydisk (2T per *group*) ..... `/mnt/mobydisk`
ZFS (5T per *group*) ..... `/zfs1 & /zfs2`

- your home directory on each of these storage systems is under your primary group name directory. It is backed up.
- Use the `id` command to find your primary group.
- Both mobydisk and ZFS are **not backed up**.
- To get ZFS storage space open a help ticket.

Software

- Applications

- Chemistry: NAMD^{s}, casino, lammps, Amber^{g}, Molpro^{s}, Turbomole^{s}, CP2K^{s}, Gaussian^{s}
- Bio: CLCbio^{h}, galaxy^{h}, bowtie^{h}, samtools^{h}, picard^{h}, trinity^{h}
- Material Sci: Material Studio, Westpa, abaqus, VASP^{s,o}

- Libraries and Programming

- Compilers: GCC^{o,s,h}, Java^{h,s}, Intel^{o,s}
- Scripting: Python^{h,s}, Perl^{h,s,o}, R^{h,s}, Matlab
- Libraries: Boost^{h,s}, FFT^{s}, tensorflow^{h,g}, mkl^{s}, hdf5^{s}, bioconductor^{h}

- Others

- Editors: vim^{o,s,h}, emacs^{o,s,h}, nano^{o,s,h}, gedit^{o,s,h}
- Debuggers: gdb^{o,s,h}, gprof^{o,s,h}
- Shells: bash^{o,s,h}, zsh^{o,s,h}

Common Commands

- Connect

```
ssh h2p.crc.pitt.edu #connect to the SMP, OPA and GPU clusters
ssh htc.sam.pitt.edu #connect to the HTC cluster
```

- Data

```
scp afile.tgz htc.sam.pitt.edu:~/ #copy from local computer to smp cluster
scp htc.sam.pitt.edu:~/afile.tgz .  #copy from htc to local computer
rsync -aP $HOME/src /zfs1/1/sam/ketan #copy src to zfs using rsync
df -h /zfs1/2/kjordan/ #find quota of a group's ZFS storage
lfs quota -g jpipas /mnt/mobydisk #find quota of a group's mobydisk storage
```

- LMod: load and unload software

```
module spider fftw #search for module named fftw
module avail #list available modules
module list #list currently loaded modules
module load compiler/python/2.7.10-Anaconda-2.3.0 #load anaconda module
module purge #unload all modules
```

Hardware

- SMP$_{Standard}$ 10 GIGE NETWORK

- 100 nodes of 24-core Xeon Gold 6126 2.60 GHz (Skylake), 192 G RAM, 256 G & 500 G SSD
- 24 nodes of 12-core Xeon E5-2643v4 3.40 GHz (Broadwell), 256 G RAM, 256 G SSD & 1 T SSD

- SMP$_{Specialty}$

- 2 nodes of 12-core Xeon E5-2643v4 3.40 GHz (Broadwell), 256 G RAM, 256 G SSD & 3 T SSD, 10 GigE
- 2 nodes of 12-core Xeon E5-2643v4 3.40 GHz (Broadwell), 512 G RAM, 256 G SSD & 3 T SSD, FDR-IB
- 1 node of 12-core Xeon E5-2643v4 3.40 GHz (Broadwell), 256 G RAM, 256 G SSD & 6 T NVMe, GigE

- MPI$_{OP}$ OMNI-PATH NETWORK

- 96 nodes of 28-core Intel Xeon E5-2690 2.60 GHz (Broadwell), 64 G RAM, 256 G SSD, 100 Gb

- MPI$_{IB}$ INFINIBAND NETWORK

- (IB) 32 nodes of 20-core Haswell (E5-2660 v3), 2.6 GHz (Haswell), 128 G RAM, 256 G SSD

- HTC INFINIBAND NETWORK

- 4 nodes of 24-core Xeon Gold 6126 2.60 GHz (Skylake) 384 G RAM, 256 G & 500 G SSD
- 20 nodes of 16-core Intel Xeon E5-2630v3, 2.4GHz (Haswell-EP), 256 G RAM, 256 G SSD

- NTA

- 8 nodes of 256-core (hyper-threaded) Intel KNL Xeon Phi 7210 1.30 GHz, 94 G RAM
- 7 nodes with 4 NVIDIA Titan X GPGPUs/node
- 8 nodes with 4 NVIDIA GeForce GTX 1080 GPGPUs/node
- 1 node with 2 NVIDIA K40 GPGPUs

Help

Does the FAQ answer your question?
`http://core.sam.pitt.edu/faqs`
Search the website:
`http://core.sam.pitt.edu/search`
Read the documentation
`http://core.sam.pitt.edu/node/6`

- When submitting support ticket

1. Provide a descriptive, specific title
2. Specify the cluster the ticket applies to
3. Provide directory location if applicable

`core.sam.pitt.edu/node/add/support-ticket`

- CRC Consultants and their Expertise

Kim Wong: Bio Simulation | Agent-based Modeling | Physics-based Modeling
Fangping Mu: Bioinformatics | Computational Biology | Computational Genomics
Barry Moore: Quantum Chemistry | HPC
Ketan Maheshwari: GPU Computing | Scientific Workflows
Shervin Sammak: Turbulent Combustion | Fluid Dynamics

- Contact / Feedback

Please send your feedback and suggestions for improvement to this document at `http://core.sam.pitt.edu/contact`

Job Scheduler Policy

Details coming soon.

- Charging

Users are charged in terms of Service Units (SU) which depend on both memory and CPU usage.

- Queues and limits

– Minimum and maximum walltimes
– Minimum and maximum nodes that may be requested

Quick Summary

Center for Research Computing
University of Pittsburgh

Computation

CRC clusters use the SLURM scheduler.

- Job Management

```
sinfo                    #view info about nodes
sbatch job.sbatch        #submit a job
squeue -u $(whoami)      #check job status
scancel 12345            #cancel job with id 12345
```

Below is the command to see details of the nodes on a particular cluster. The key column is the CPUS(A/I/O/T) where,
A = allocated, I = idle, O = other, T = total
Single-core jobs should fit within nodes with non-zero I.

```
sinfo --format="%10n
%9P %.14C %.10T"
```

Note: The above commands will show info about the *SMP* cluster by default. To see info about other clusters, use the **-M** option with name of cluster, eg.`squeue -M gpu -u $(whoami)`

```
crc-sinfo.py             #current hardware status
crc-spider.py            #regular expression search for modules
crc-squeue.py --start    #show job start time
crc-squeue.py --watch    #watch job progress
crc-squeue.py --all      #show all jobs
crc-usage.py             #show your usage on cluster
crc-interactive.py --smp --time=1 --num-cpus=2   #run interactive job on cluster
```

- Job Status Codes

PD	Pending
R	Running
CA	Cancelled
F	Failed
CF	Configuring
TO	Timed Out
PR	Preempted
NF	Node Failed
S	Suspended
CG	Completing
CD	Completed

Example SLURM Scripts

- for HTC

```
#!/bin/bash
#SBATCH -N 1                          # request one node, ensure that all cores are on one machine
#SBATCH --job-name=testR              #name of job: will show up in status output of squeue
#SBATCH --output=testR.out            #standard output goes to this file, SLURM will create one if not provided
#SBATCH -t 00:10:00                   # 10 minutes walltime in HH:MM:SS format
#SBATCH --cpus-per-task=16            # Request that ncpus be allocated per process.
#SBATCH --mem=230g                    # Memory requested for all cores (see also --mem-per-cpu)
module load HISAT2/2.0.3-beta
hisat2-build -p 16 --ss hg38.ss --exon hg38.exon hg38.fa hg38_tran
```

- for SMP/MPI/OPA

```
#!/bin/bash
#SBATCH --job-name=p7n64
#SBATCH --nodes=2                     #number of nodes requested
#SBATCH --ntasks-per-node=1
#SBATCH --partition=opa               # partition name is required
#SBATCH --cluster=mpi                 # cluster name is required
#SBATCH --mail-user=shs159@pitt.edu   #send email to this address if ...
#SBATCH --mail-type=END,FAIL          # ... job ends or fails
#SBATCH --time=4-00:00                # 4 days walltime in dd-hh:mm format
#SBATCH --qos=long                    # required if walltime is greater than 3 days
#SBATCH --reservation=sam_4
module purge                          #make sure the modules environment is sane
module load intel/2017.1.132 intel-mpi/2017.1.132 fhiaims/160328_3
export I_MPI_FABRICS_LIST=tmi:I_MPI_FALLBACK=0
srun dg3d_explicit_mpi.mpi $HOME/p6n32/run.inpt
```

- for GPU

```
#!/bin/bash
#SBATCH --job-name=gputf
#SBATCH --output=gputf.std.out
#SBATCH --error=gputf.std.err         #standard error goes to this file
#SBATCH --time=00:10:00
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --cluster=gpu
#SBATCH --partition=gpu
#SBATCH --gres=gpu:4                  #use all four GPU devices on this node
module purge
#if your executable was built with CUDA, load the CUDA module:
module load cuda/8.0.44
module load python/anaconda2.7-4.2.0
module load gcc/5.4.0
python tensorflowtest.py
```

Troubleshoot

- ssh connection

- If you are on a wireless network, make sure the **VPN** connection is established.
- Use `ping` to check network connectivity to host, eg. `ping htc.sam.pitt.edu`
- Use ssh in verbose mode with `-v` to identify possible causes, eg. `ssh -v h2p.crc.pitt.edu`

- jobs submission

- Sanity test the environment by submitting a simple job, eg.

```
#!/bin/bash
#SBATCH --output=test.out
#SBATCH --error=test.err
#SBATCH -t 00:10:00
srun echo "Hello $(hostname)"
```

- Check output of `squeue -t PD` and `smap`

- software access

- Check if the software is available using `module spider` and `module avail`
- Check if the module is loaded with `module list`

Notes

- In software, superscripts indicate availability over a cluster: h=HTC, s=SMP, g=GPU, o=OPA, items in red indicate licensed software
- To print this document on letter size paper use the printer's **fit to size** option.
- LaTeX source on git: `github.com/ketancmaheshwari/crc_cheatsheet`.

- Acronyms

HTC	High Throughput Computing
OPA	Omni-Path
IB FDR	InfiniBand Fourteen Data Rate
GPU	Graphics Processing Unit
SMP	Shared Memory Processing
SSD	Solid State Drive
NTA	Non-Traditional Architecture
NVMe	Non-volatile Memory express
ZFS	Zettabyte File System