HPCC Basics

Wharton Research Computing Training

Resources

Documentation: http://research-it.wharton.upenn.edu/documentation/
Status Overview: http://hpcc.wharton.upenn.edu/ (from Upenn campus network only)
E-mail: hpc-admin@wharton.upenn.edu (bug reports, environment questions and issues, account support)
research-computing@wharton.upenn.edu (more detailed programming questions or requests)
Peers: Each other (look around you, these people can be your best resources)
The web: Google is amazing, particularly for CODE help

Hardware:

- 32 servers with a total of 512 cores
- 16 cores per server
- 5 TB of total cluster RAM, 128GB per server
- Several TB of HD space. Ability to grow easily

Lots of software: Matlab, Mathematica, Python, R, SAS, Stata, GNU and Intel C/C++/Fortran, OpenMPI
Other software available upon request, you may install personal software in your home directory

UNIX (Red Hat Enterprise Linux)

To use the Wharton HPCC environment, you will need to know at least some basic UNIX commands. There are numerous guides and cheat sheets to using UNIX from the command line. I prefer ones that are in a PDF card format, like:

http://research-it.wharton.upenn.edu/wp-content/uploads/2013/09/UNIX-Reference.pdf

UNIX command 'manual' pages: man *command*
You can also do an internet search for 'man *command*'. It's often easier to read in a browser. *NOTE: be aware of possible version differences if you read a man page from the web.*

What Limits Do Users Have?

CPU/cores: each user may (by default) use up to **100 cores** simultaneously, in **any combination** (100 single jobs, 2 x 50-core jobs, etc). Each user may also have another 400 jobs (for a total of 500) queued up and waiting to be launched as running jobs complete. If you want to queue more than 500, use a task array (-t option to qsub). More below.
RAM: 750GB of total RAM (all running jobs combined) per user**
RAM: 250GB of RAM per compute host (for a limited number of hosts, the others have 125GB)
Disk: each users has (by default) a 100GB quota
** What about **RAs** or **Co-Authors**? All limits are combined under the PI's limits. So if you are Prof X's RA, you and the Prof *both* share 100 cores and 100GB disk.

That said, these limits are a generous *baseline*. If you have specific requirements above and beyond these limits, just get in touch and we will discuss your several options, depending on what you may need.

Accessing Wharton HPC Resources

MD5 Fingerprints

When you first logon to Wharton HPC systems (from any different individual computer), you will be notified by your SSH or SFTP client that the keys being received are 'untrusted'.
Below are the MD5 public key fingerprints that you may be offered, and which can be trusted (and saved):

MD5 Fingerprint	BubbleBabble Fingerprint	BITS	Type
60:f8:0c:0c:66:6c:35:33:57:36:51:68:0f:01:d9:78	xeroc-kovyp-redys-puroz-porob-rynep-sysen-cutok-vibuf-vyvos-zyxux	1024	DSA
a8:77:08:5e:9a:ec:15:45:90:ee:45:43:d7:81:66:c9	xunap-fohal-cihob-kohyn-gydep-holol-vunec-poryb-saneg-foniz-boxux	1024	RSA1
2f:50:23:86:ec:e8:22:84:f5:e2:5e:92:00:38:b0:61	xozeb-ladiv-fozab-kotec-bisur-monah-cinad-ruhum-novig-gokol-paxax	1024	RSA
93:d0:4d:31:0e:a2:f2:d3:e3:df:6b:ee:eb:1d:ab:3e	xugaf-metob-rakyk-vumyr-guhan-sigig-tisyd-nulih-baryn-tubom-coxex	256	ECDSA

Command Line: SSH

NOTE: you *must* use the Wharton VPN for all non-Penn Campus network access!!

Mac OSX: built-in ssh (from Terminal)

- OPEN Terminal: Go > Applications > Terminal
- CONNECT (log on): in the Terminal window, type: `ssh` *`username`*`@hpcc.wharton.upenn.edu`
  - Don't forget to LOOK at the MD5 fingerprint and make sure it matches one from the *MD5 Fingerprints* table above
  - `exit` (or `logout`) to exit
  - Optional: Use Keys (instead of passwords):
    - In MobaXterm command line window, type: `ssh-keygen`
      - Defaults are fine
    - Append new public key to your authorized_keys file on the HPCC: `ssh` *`username`*`@hpcc.wharton.upenn.edu echo "$(cat ~/.ssh/id_rsa.pub) >> ~/.ssh/authorized_keys"`
    - You are now all set to ssh using keys (no password needed!)

Windows: MobaXterm

Available from: http://mobaxterm.mobatek.net/download-home-edition.html

- INSTALL & LAUNCH MobaXterm
- CONNECT (log on): in the MobaXterm window, type: `ssh` *`username`*`@hpcc.wharton.upenn.edu`
  - Don't forget to LOOK at the MD5 fingerprint and make sure it matches one from the *MD5 Fingerprints* table above
  - `exit` (or `logout`) to exit
- Optional: Use Keys (instead of passwords):
  - In MobaXterm command line window, type: `ssh-keygen`
    - Defaults are fine
  - Append new public key to your authorized_keys file on the HPCC: `ssh` *`username`*`@hpcc.wharton.upenn.edu echo "$(cat ~/.ssh/id_rsa.pub) >> ~/.ssh/authorized_keys"`
  - You are now all set to ssh using keys (no password needed!)
- MULTIPLE WINDOWS: click the '+' to create a new window tab, or choose Split type from the Split icon at the top (I love 4!)

File Transfer (SFTP)

Mac OSX: Fetch

University licensed, available at: http://www.upenn.edu/computing/product/ (FTP)

Windows: WS_FTP Professional

University licensed, available at: http://www.upenn.edu/computing/product/ (FTP)

- Install and launch
- "Open a Remote Connection" link
  - Create Site …
    - Site Name: Wharton HPC (or whatever you like, this is just a 'name')
    - Connection Type: SFTP/SSH
    - Server Address: hpcc.wharton.upenn.edu

  - Username: your Wharton Username
  - Password: your Wharton Password
- CONNECT
  - LOOK at the "Untrusted Public Key" MD5 Fingerprint and make sure it matches one from the *MD5 Fingerprints* section on page one. If so, you can "Trust this key". If NOT please contact hpc-admin@wharton.upenn.edu and let them know that there is an SSH key mismatch, and do not continue to connect
- You can drag-and-drop in and out of the windows, or use the arrows to do transfers
- **ASCII tip:** Linux and Windows use different end-of-line characters, which can cause problems when trying to move scripts between Windows and Linux. You may wish to add .DO, .M, .R, .SAS and any other "text" (script) files that you commonly use to the ASCII Filenames list in Tools menu > Options… > Transfers > ASCII Filenames list. This will 'translate' the files during transfer. .SH, .LOG, and many others are already configured to 'translate' during transfer.

File Transfer (Windows Share /network drive)

FROM ON CAMPUS ONLY or VIA WHARTON VPN CONNECTION (see your departmental rep): \\hpcc.wharton.upenn.edu\username
You can map this as a network drive.

- Windows: My Computer > Map Network Drive > Folder: \\hpcc.wharton.upenn.edu\username. If the system is not on the Wharton domain (a Wharton-owned system), you should check the "Connect using different credentials" checkbox, and use User name: "wharton\username" and your normal Wharton password
- Mac OSX: Go menu > Connect to Server … Server: smb://hpcc.wharton.upenn.edu/*username*

NOTE: if you upload files via Windows Share/Samba they may need conversion before running. If you experience an error when you run a program that you've uploaded this way, try 'dos2unix <filename>' and resubmitting.

File Transfer (Dropbox)

Dropbox integration allows you to sync your Dropbox (or a subset) to Wharton's HPC environment.

Prerequisite: Dropbox Account (http://dropbox.com/)
On hpcc.wharton.upenn.edu:

```
cd ~
ln -s /usr/local/dropbox .dropbox-dist
.dropbox-dist/dropboxd
```

You will see a reply similar to:

```
Please visit https://www.dropbox.com/cli_link?host_id=4e0393a9546d462ef86127&cl=en_US to link this machine.
```

Copy the link (**NOTE: CTRL-c *does not mean "copy" in UNIX!! DON'T do it!!***), open a web browser (on your laptop or desktop, not on the server) and paste in the link from your ssh window. You should see a reply like:

Client successfully linked, Welcome Hugh!

CTRL-c to stop the daemon (see, *that's* what CTRL-c does! Stops things)

```
dropbox start
```

Your Dropbox folder should now be synced (or syncing, depending on size).

To EXCLUDE directories from syncing with the Wharton HPC environment (very useful, so only my Wharton/Research directory gets synced, or whatever):

```
cd ~/Dropbox
```

```
dropbox exclude add dir1/file1 dir2/file2 dir3/file3
```

Even better, *start with everything excluded:*

```
cd ~/Dropbox
dropbox exclude add *
```

Then unexclude needed directories or files:

```
dropbox exclude remove Wharton/Research/HPC
```

See '`dropbox help`' for more details.

Graphics: X

Windows: MobaXterm

X forwarding to MobaXterm is 'on' by default!

SECURITY NOTE: When you run MobaXterm X for the first time, you may receive a request from Windows Security to allow access. You do *not* need to allow that access for X to work correctly, as we will be 'tunneling' it over the SSH connection. So you should click Cancel, otherwise you are unnecessarily opening a security "hole" in your firewall.

Mac OSX: X11 (Applications/Utilities/X11.app)

Available or not based on OSX version:

- Lion (10.7) – pre-installed, you're all set
- Tiger (10.4), Leopard (10.5), and Snow Leopard (10.6), on OS disk, but needs to be installed
- **Mountain Lion (10.8) – *not installed or on OS disk***, get XQuartz from The Quartz Project (http://xquartz.macosforge.org/)

Once installed, just add -Y or -X to your normal ssh command (see SSH section above), and graphics will be forwarded to your local system's monitor for viewing.

Portable Devices, other OSes, etc.

Many portable devices have SSH apps. We like iSSH for the iPad and iPhone. We will make an effort to support these means of access, but can't promise much support, as the volume of apps in this space is quite large. But often quite useful!

Sharing Files, Working Collaboratively

You may wish to share files or work with other users. Dropbox is an easy solution, and you can also work in shared folders on the system. For details see http://research-it.wharton.upenn.edu/documentation/sharing/. If there is time at the end we can work with you to get either of these solutions set up for you.

Working Efficiently

Organization

Use an organized directory structure. The format of this structure will depend on your project(s), but it might look something like:

```
common
common/data
common/data/datafile1
common/code
common/code/codefile1
common/logs
project1
project1/data
project1/code
```

```
project1/logs
project2
project2/data
project2/code
project2/logs
```

Relative Paths

Use relative paths. In other words, try to *not* have /home/*dept*/*username/* at the beginning of any file paths in your code. ~ is better, and *nothing* is even better. This makes the code much more portable!

Software Tricks

There are some tricks in specific software packages to translate from Windows to Linux/OSX. For example Matlab has both the fullfile and the filesep functions:

```
f = fullfile('dog','cat','frog')

f = dog/cat/frog

>> f = ['dog' filesep 'cat' filesep 'frog']

f = dog/cat/frog
```

Setting Up and Running Jobs!

*You may not run **intensive processes*** on the login nodes. This is to prevent users from creating problems for other users. To this end, there is very little usable research software on the login nodes, and if we notice you doing intensive things on the login nodes we may restrict your access or kill (cancel) your running processes without warning.

Gaming the system: our hardware and software does a pretty good job of keeping you from doing any harm to other users' research activities. If we suspect this kind of activity, you may lose your account. Work with us! If you have questions about your methods, or need resources, let us know.

Interactively

`qrsh` *`command`* – run a single command on a compute node
`qlogin` – log onto a compute node (generally used for code development and on-the-fly testing)

Batch

This is what Wharton HPC is all about!

Get the example code for an R job and submit a job:

Method 1:

```
cd ~
cp -r /usr/local/demo/R .
cd R
qsub job-script.sh
```

Method 2:

```
qsub -N Rdemo -b y 'R --no-save < demo.R'
```

Modify your command so we can examine running job details:

```
qsub -N Sleeper -b y 'R --no-save < demo.R; sleep 300'
```

```
qstat
qstat -j job-ID
qdel job-ID
qstat -s z
qacct -j job-ID
```

Job Arrays

```
qsub -t 1-4 job_script.sh
qsub -N array1 -t 1-4 'R --no-save < demo.R; sleep 300'
```

You can 'continue' an array job with higher numbers by changing `'-t 1-4'` to `'-t 5-100'`, etc.

WHY would you want to run the same thing over and over? SGE_TASK_ID is the answer to some tricky job setups. Consider these two use case examples:

Example #1: Create 10 script files named mycode-1.R (or mycode-1.m, etc.) through mycode-10.R. Now run:

```
qsub -N MyRArray1 -t 1-10 -b y 'R --no-save < mycode-$SGE_TASK_ID.R'
```

So there you go, we're launching all ten with one command.

Example #2: *more* useful: let's say you 10 tab-separated text data files to evaluate with the same code ... name them mydata-1.txt through mydata-10.txt, and within a Matlab script file called 'mydataread.m':

```
cd ~
cp -r /usr/local/demo/job_array .
cd job_array
qsub -N array -t 1-10 mydataread.sh
```

Each of the ten jobs will read the data file generated with strcat, getenv, and the SGE_TASK_ID environment variable, which is different in each array sub-job.

Parallel Jobs

Do stuff in parallel.

Create a script using cat (or some other method that you prefer):

```
cat > mpitest.sh
#!/bin/bash
hostname
sleep 120
Ctrl-d
```

Make sure it looks good:

```
cat mpitest.sh
```

Change permissions so it will run:

```
chmod +x mpitest.sh
ls -l mpitest.sh
```

Run the script via mpiexec in the mpi parallel environment ('pe'):

```
qsub -pe openmpi 4 -N mpitest -j y -b y 'mpiexec ./mpitest.sh'
qstat
qstat -f
```

Using More RAM

To use more than the default 7.5GB of RAM for a job, modify the `m_mem_free` option (default 7.5) of the job, either in your job script:

```
#$ -l m_mem_free=12G
python myPythonCode.py
```

Or as an option to your qsub command: `qsub -l m_mem_free=12G`

That option will be passed to all slots used for a job, so if you're doing a parallel job, each worker will have the adjusted value.

Keep in mind the RAM limits:

- **RAM:** 750GB of RAM total (all jobs combined) per user*
- **RAM:** 250GB of RAM per compute host (for a limited number of hosts, the others have 125GB)

For example: if you request 50GB of RAM (`-l m_mem_free=50G`) 100 task Array Job, you will have 15 running jobs (750/50=15), and 85 queued jobs.

Job Status and Monitoring

- What jobs am I currently running? `qstat`
- How can I CANCEL a running job? `qdel job-ID`
- What jobs are complete? `qstat -s z`
- How can I get stats for a running job? `qstat -j job-ID`
- How can I get stats for a completed job? `qacct -j job-ID`
- How can I look at job output? `cat jobname.ojob-ID`
- How can I look at job error? `cat jobname.ejob-ID` (see *Logging* below for how to combine output and error logging)

Logging

- SGE creates an output and an error log. Can I combine them? Yes! Add the following option to qsub: `-j y`
- SGE names the output and error log files based on the script name. Can I change this? Yes! Add this option to qsub: `-o outputfilename`
- SGE names the job after the script name. Sometimes I use the same script for different jobs, or I use the `‘echo “command” | qsub’` submission method, which uses STDIN as the job name. Can I change the job name? Yes! Add this option to qsub: `-N jobname` (cannot start with a number)

E-Mail Notification

- To send an e-mail at job completion, add these options to qsub: `-m e -M e-mail@email.com`

Practicing and Examples

There are various demo scripts in the /usr/local/demo directories for your use. These directories generally include a `README.txt` file with details on running the demo, a demo script written in the language being demoed, and a job launch script to launch the job with qsub. Please feel free to make use of these scripts! This is accessible via the command line, or SFTP.

WRDS Data

WRDS data is available at the WRDS website (along with a lot of great ways to get and query it!): http://wrds.wharton.upenn.edu/, in the /wrds directories. If you’re using WRDS and SAS exclusively, and have SAS on your local system, you can use the SAS/CONNECT method to connect and use our SAS/CONNECT server (sastcpd.wharton.upenn.edu). Take a look at the SAS Page in the Research Wiki (https://wiki.wharton.upenn.edu/researchcomputing/SAS).