

# Getting Started Using the FAS Clusters

Stephen Weston

High Performance Computing  
Yale University

April 5, 2013



# Why use a cluster?

Reasons that you might want to use a cluster include:

- Don't want to tie up your own machine for many hours or days.
- Need a lot of memory/RAM.
- Need a lot of disk space.
- Want the results sooner.
- Want to run on multiple nodes to get more aggregate memory, disk and CPU.



# Why not use a cluster?

Reasons that you might not want to use a cluster include:

- Your program is very interactive.
- Your program only runs on Windows.
- You hate using the command line.



# Clusters are not like desktops

They are primarily intended for long running, non-interactive jobs. For that reason, they can take some getting used, particularly if you've only used Windows or Mac OS X.



# Getting Access to an FAS cluster

Submit a request to... Generate an ssh key pair...

`https://hpc.research.yale.edu`



## Log into a login node

You must first login to an Omega login node using **ssh**. From a a Mac or Linux machine, you simply use the **ssh** command:

```
$ ssh omega.hpc.yale.edu
```

From a Windows machine, you can choose from programs such as PuTTY or WinSCP, both of which are available from the Yale Software Library:

<http://software.yale.edu/Library/Windows>

For more information on using PuTTY, go to the following URL and search for "create ssh key":

<https://hpc.research.yale.edu>



# The most important shell commands

- ls
- pwd/cd
- cp/mv/rm
- cat/more/less
- nano/vi/emacs
- R/python/matlab/math
- man



# Allocate a compute node

From a login node, you allocate one or more compute nodes with the `qsub` command.

- Specify the queue with `-q` option
- Specify the memory with `-l mem=XXgb`
- Specify the walltime with `-l walltime=DD:HH:MM:SS`
- For an interactive job using `-l` option
- Optionally specify the number of processors with `-l procs=XX`

```
$ qsub -I -q fas_devel -l mem=8gb,walltime=1:00:00
```





# Notes on compute nodes

- Many things will mysteriously fail if you do them on a login node.
- Always specify **mem**, or your session could be killed if you exceed 256 MB.
- **fas\_devel** is useful for interactive sessions like installing packages.
- **fas\_devel** is a *shared* queue.
- **fas\_devel** has a maximum **walltime** limit of one hour.
- **fas\_normal** has a maximum **walltime** limit of one day.



# Module files initialize your environment

Load the module files for software that you wish to use

```
$ module load Apps/R/2.15.3  
$ module load Langs/Python/2.7.3  
$ module load Apps/Matlab/R2012b  
$ module load Apps/Mathematica/9.0.1
```

- Sets all the appropriate environment variables for the package.
- Can be loaded in **.bashrc** or Torque/Moab batch script.
- Supports multiple versions of packages.



# Installing R packages in your home directory

To install a package from CRAN to your home directory, use the **install.packages()** function:

```
$ R  
> install.packages('caret')
```

- This will download the package and its dependencies from CRAN.
- It may prompt you for the CRAN mirror to use.
- The first time, R will ask if it should create a directory in which to install this package.
- If others will need this package, send a request to [hpc@yale.edu](mailto:hpc@yale.edu).



# Installing Python modules in your home directory

To install a package from CRAN to your home directory, use the **easy\_install** or **pip** command:

```
$ easy_install Pydap
```

- This will download the package and install it.
- It may not install its dependencies.
- If others will need this package, send a request to [hpc@yale.edu](mailto:hpc@yale.edu).

