Introduction to CARC

- To provide high performance computing to academic researchers.

Machines

- Metropolis
- Nano
- Galles
- Poblano
- Pequena
- Gibbs
- Ulam

Machines (cont.)

- Machine details, just for example:
- Nano (supercomputer)
  - 35 nodes
  - 4 cores / node
  - Myrinet internal network
- Galles (beowulf cluster)
  - 200 nodes
  - 2 cores / node
  - Gigabit ethernet internal network

Machines (cont.)

What is a core?

-A core is the processor that translates computer code into calculations. When you request “ppn” in your batch file, you are requesting cores.

-”Core” was once synonymous with “CPU”, and we still sometimes use them interchangeably.

-For our purposes they are the same thing.

-More cores = a good thing.

Machines (cont.)

- You can find out how many nodes a machine has by typing “qgrok”.
  - “qgrok” also reports how many nodes are available for your jobs to run.
  - “qgrok” reports the total cores, so you can determine how many cores there are per node.

Machines (cont.)

nano:~ # qgrok

queue: free busy offline Running jobs
Nodes CPUs

DEFAULT	14	20	1	11	35	140
totals:	14	20	1	11	35	140

Getting Access

- To get access to a particular machine you need to have a project registered with CARC Under the project you may request accounts.
- Have your PI request a project at carc.unm.edu. Then…
  - Go to carc.unm.edu and click Request CARC Account.
  - Accounts are per machine. You may request any machine but a few are special access.

Available Resources

- General access machines
  - Metropolis and Galles
    - These have a lot of nodes and cores and are therefore are well suited for “embarassingly parallel” codes.

Available Resources (Cont.)

- General Access (fast network)
  - Nano, Pequena, Ulam, Metropolis
    - These have a moderate amount of nodes and fast internal networks and are therefore best suited for tightly-coupled parallel codes, *although* they may also be useful for “embarassingly parallel codes.”

Available Resource (Cont.)

- Restricted Access (more or less)
  - Gibbs and Poblano
    - These machines were purchases by groups of PI's and may be used with the approval of the Director of CARC.
    - Gibbs is a very fast multi-node machine with a large number of cores per node.
    - Poblano is a fast single node machine with a lot of shared RAM, 256GB.

Available Resources (Cont.)

- Disk space
  - Each machines uses the same home directory.
  - In addition there are one or more scratch areas.
  - Speed: ~75 MB/s is the upper bound for writing data to home dirs or scratch.
- Backups
  - Home directories are backed up.
  - Scratch may disappear at any time.

Linux Basics

- Linux is a free operating system (OS). Examples of Linux flavors include:
  - SUSE, Ubuntu, Mint, Slackware, and RedHat
- CARC uses mainly *SUSE and Ubuntu*, but they all work pretty much the same way.

Linux Basics (Cont.)

- Logging In
  - Once you have an account, you log into CARC machines with Secure Shell (SSH). You make this connection by typing:

  “ssh username@metropolis.alliance.unm.edu”

  in a Linux terminal. Or you can use your favorite Windows SSH client program. Mac has one built in, available from the Terminal program. On Windows, Putty is a good and free ssh program.

Linux Basics (Cont.)

- Transferring Files
  - Three principal ways exist
    - sftp, scp and rsync
      - These are all command line programs, but Windows equivalents exist.
      - A graphical program that does sftp and scp is called filezilla.
    - sftp
      - Type “sftp username@metropolis.alliance.unm.edu”
      - Use regular FTP syntax to transfer files, for example...

Linux Basics (Cont.)

- ebryer@farnsworth:~$ sftp ebryer@metropolis.alliance.unm.edu
- Connected to metropolis.alliance.unm.edu.
- sftp> cd ERIK
- sftp> get hw.f90
- Fetching /users/ebryer/ERIK/hw.f90 to hw.f90
- /users/ebryer/ERIK/hw.f90 100% 1119 1.1KB/s 00:00

Linux Basics (Cont.)

- ebryer@farnsworth:~$ scp -r ebryer@nano.alliance.unm.edu:ERIK/gp .
- myscript.Rout.3999
  100% 1722 1.7KB/s 00:00
- myscript.Rout.4001
  100% 1719 1.7KB/s 00:00
- R.pbs.o347399
  100% 303 0.3KB/s 00:00

Linux Basics (Cont.)

- sftp is good for transferring one file at a time and browsing directories.
- scp excels at transferring whole directories and is often the better choice.
- rsync does it all, however the syntax of the command is a little tricky. For large data transfers, rsync is *the best by far*, because it can continue interrupted transfers.

X Windows

- Local viewing of graphical programs running on (remote) servers.
- Several ways to accomplish this:
  - X Windows
  - VNC
  - Virtual machine combined with either of the above.

X Windows (Cont.)

- Ways to run X Windows (all free)
  - Linux: native
  - Windows: Cygwin (and others)
    - Tricky to install but provides a full Linux-like environment.
  - OS X: Xquartz will provide X Windows for free

X Windows (Cont.)

- To start X Windows on your desktop/laptop
  - on Windows run the X-server
    - Start Menu → Cygwin-X → X-server
  - Starts automatically on OS X
  - Already running on Linux desktops

X Windows (Cont.)

- To remote-display from a server to a desktop:
  - Start the local X server if necessary.
  - ssh into the server
    - *ssh -X ebryer@nano.alliance.unm.edu*
    - Type *xclock*
      - A clock should appear on your desktop.
      - Type *Ctrl-C* to kill the clock.
      - Type *xterm &* to start another xterm and leave you in control of the first one.

Summary

- Seven machines exist for your use. These can run multiple simultaneous jobs to permit you to get more work done. (More on this shortly.)
- Login is via *ssh*.
- File transfer is often best done with *scp*.

Summary (cont.)

Remote-display of graphical programs such as *rstudio* or the *matlab* GUI can be accomplished with X windows, however Matlab is often more efficiently run in command-line mode, i.e. *matlab -nojvm -nodisplay*