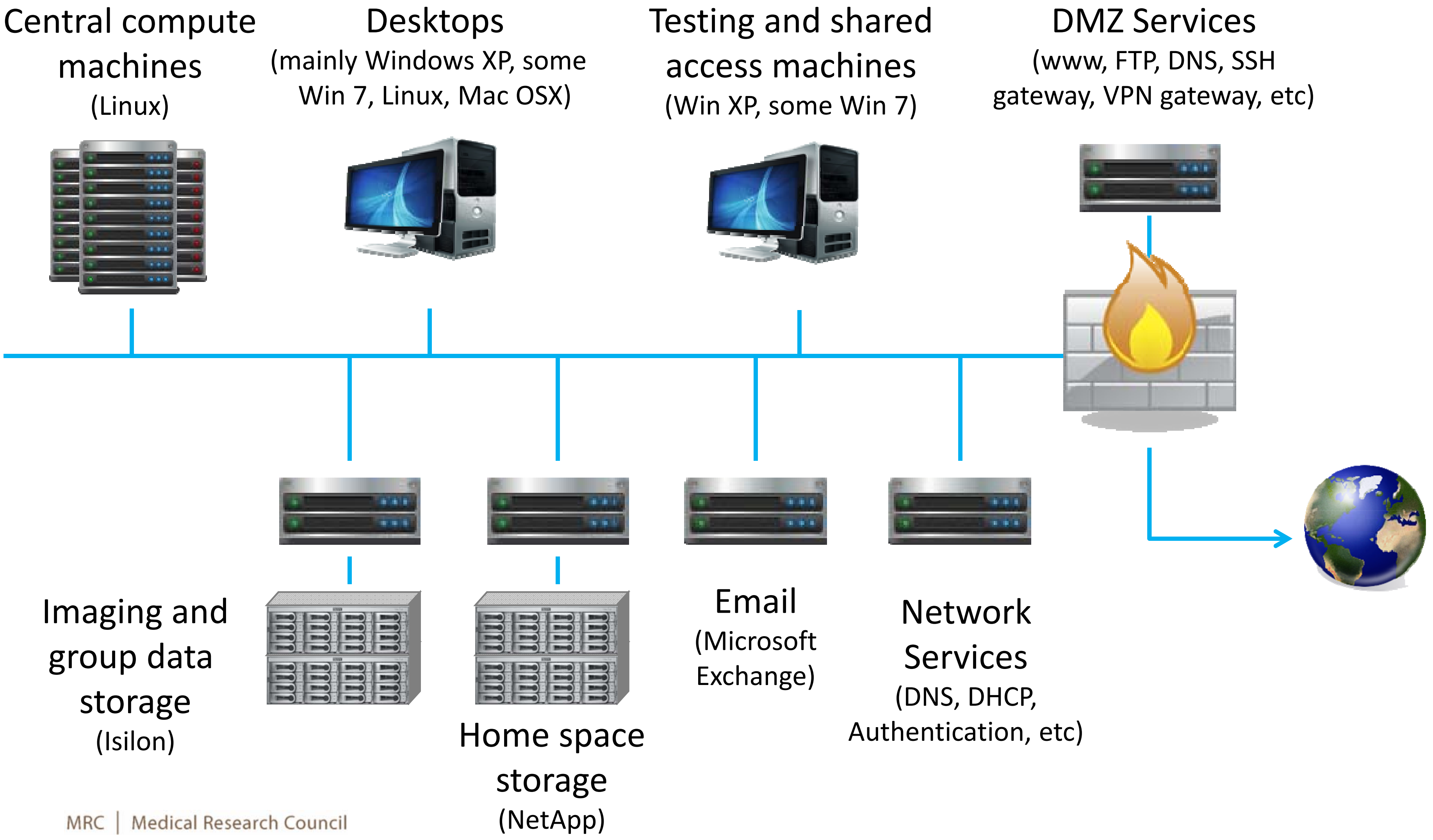
Introduction to the CBU computing system

Russell Thompson

Overview

- Computing resources
- Accessing resources
- Using compute cluster
- Scientific software
- Best practices

Computing Resources



Central compute machines (Linux)

Desktops (mainly Windows XP, some Win 7, Linux, Mac OSX)

Testing and shared access machines (Win XP, some Win 7)

DMZ Services (www, FTP, DNS, SSH gateway, VPN gateway, etc)

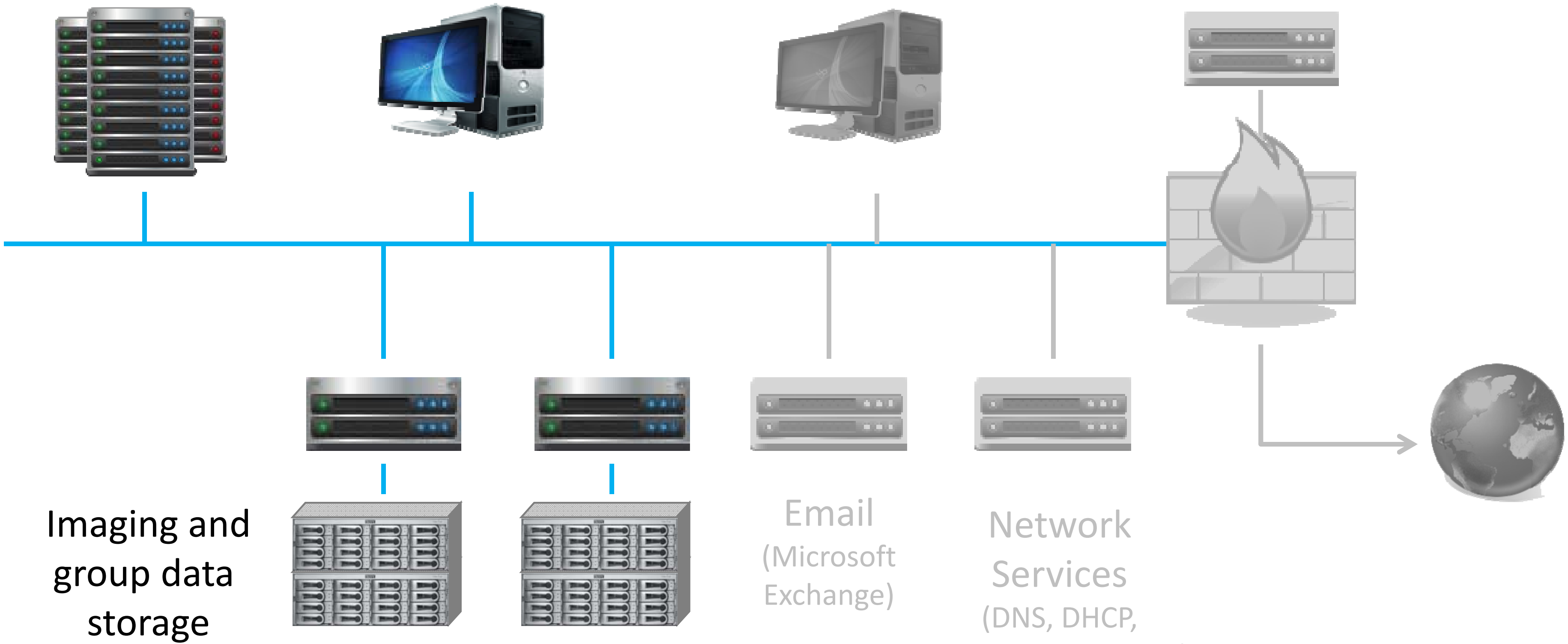
Imaging and group data storage (Isilon)

Home space storage (NetApp)

Email (Microsoft Exchange)

Network Services (DNS, DHCP, Authentication, etc)

Computing Resources



Central compute machines (Linux)

Desktops (mainly Windows XP, some Win 7, Linux, Mac OSX)

Testing and shared access machines (Win XP, some Win 7)

DMZ Services (www, FTP, DNS, SSH gateway, VPN gateway, etc)

Imaging and group data storage (Isilon)

Home space storage (NetApp)

Email (Microsoft Exchange)

Network Services (DNS, DHCP, Authentication, etc)

Network Storage

Home space:

- Permanent staff get 50GB quota
- All disk based
- Snapshot backups – hourly / nightly / weekly
- Replicated hourly to offsite system
- Intended to store scripts, figures, documents etc – things that can't be recreated via script.
- Personal to you, by default not accessible by anybody else

Network Storage

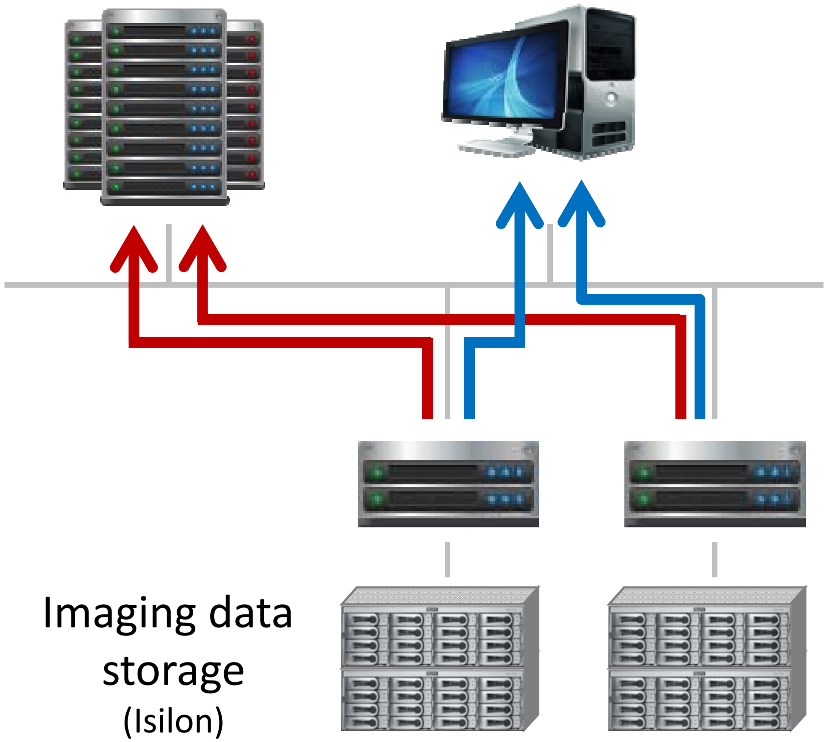
Imaging space:

- No quotas
- Not created by default – available on request for people doing imaging analysis
- Default permissions allow all members of the imagers group to read each others' directories
- 600TB Disk based storage (replicated off site)

Shared research group areas:

- No quotas
- Created to allow members of specific labs / research groups to share data
- Access limited to members of the relevant research group

Accessing Resources - Network storage



Central compute machines (Linux)

Desktops (mainly Windows XP, some Win 7, Linux, Mac OSX)

Imaging data storage (Isilon)

Home space storage (NetApp)

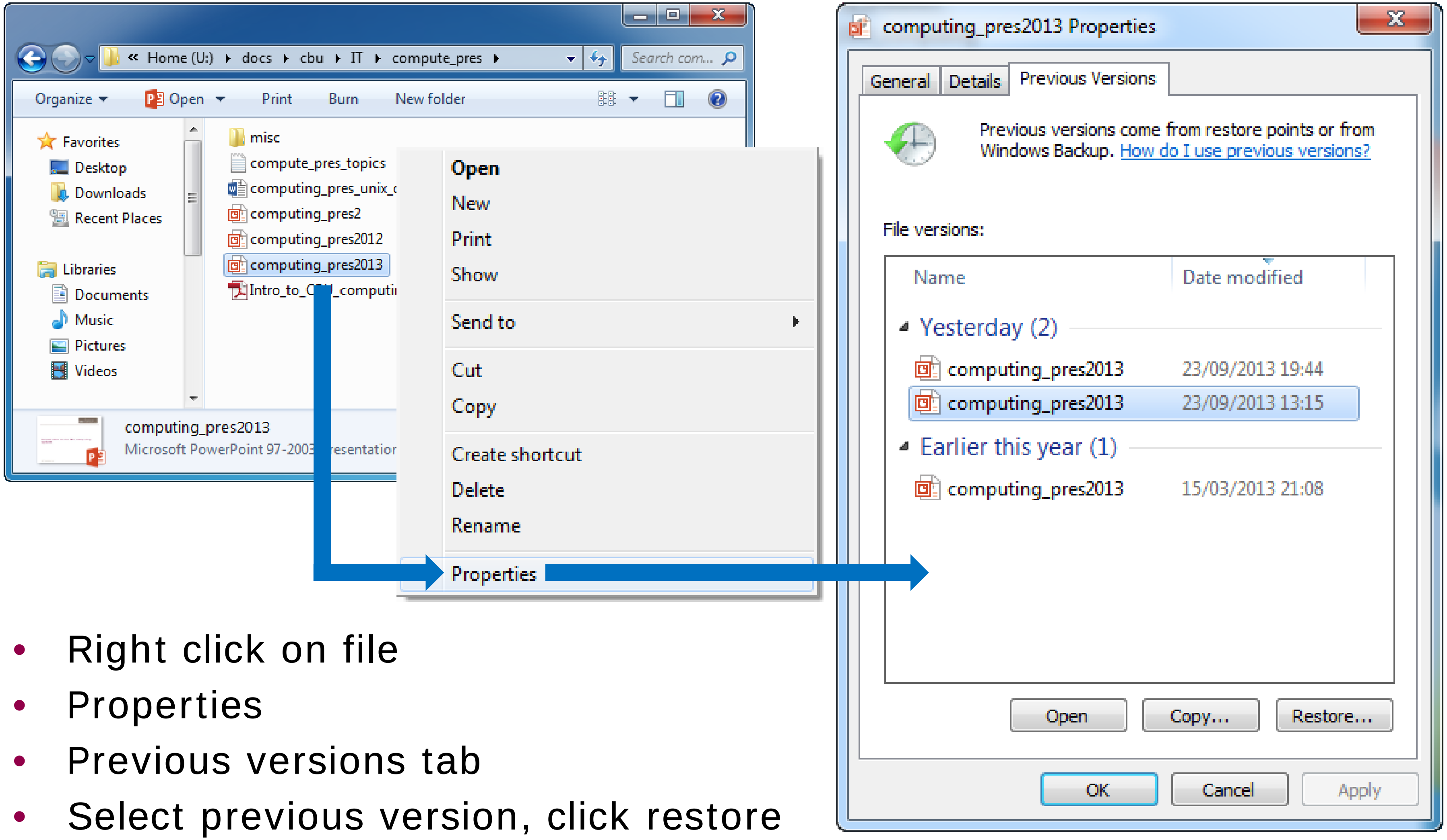
To access:

	Home space	Imaging space
Win	\\home\username U:\	\\ice1\imaging \\cbsu\data\imaging
Unix	/home/*username*	/imaging/*username*

🟦 CIFS / Samba shares

🟥 NFS shares

Restoring from a snapshot - Windows



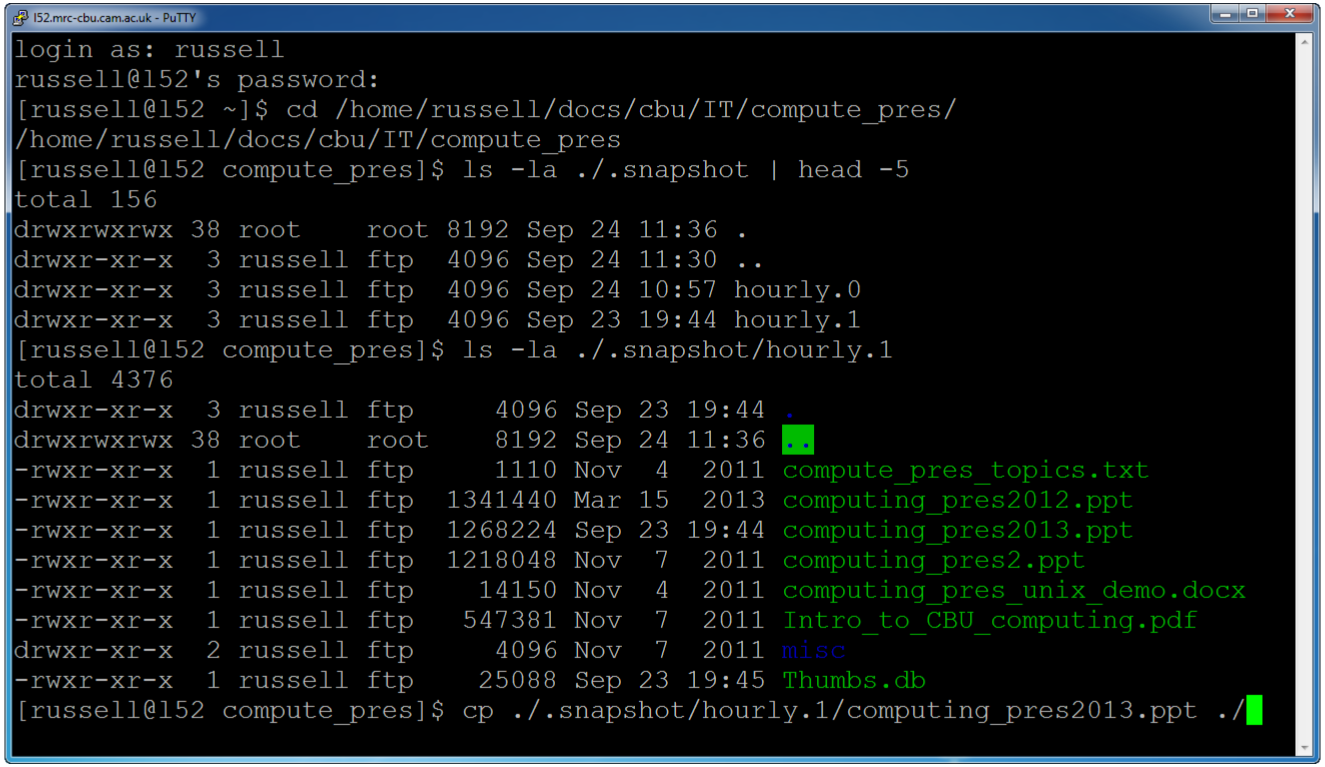
Previous versions come from restore points or from Windows Backup. How do I use previous versions?

File versions:

Name	Date modified
Yesterday (2)	
computing_pres2013	23/09/2013 19:44
computing_pres2013	23/09/2013 13:15
Earlier this year (1)	
computing_pres2013	15/03/2013 21:08

- Right click on file
- Properties
- Previous versions tab
- Select previous version, click restore

Restoring from a snapshot - Linux



```
login as: russell
russell@l52's password:
[russell@l52 ~]$ cd /home/russell/docs/cbu/IT/compute_pres/
/home/russell/docs/cbu/IT/compute_pres
[russell@l52 compute_pres]$ ls -la ./.snapshot | head -5
total 156
drwxrwxrwx 38 root    root 8192 Sep 24 11:36 .
drwxr-xr-x  3 russell ftp  4096 Sep 24 11:30 ..
drwxr-xr-x  3 russell ftp  4096 Sep 24 10:57 hourly.0
drwxr-xr-x  3 russell ftp  4096 Sep 23 19:44 hourly.1
[russell@l52 compute_pres]$ ls -la ./.snapshot/hourly.1
total 4376
drwxr-xr-x  3 russell ftp     4096 Sep 23 19:44 .
drwxrwxrwx 38 root    root    8192 Sep 24 11:36 ..
-rwxr-xr-x  1 russell ftp     1110 Nov  4  2011 compute_pres_topics.txt
-rwxr-xr-x  1 russell ftp  1341440 Mar 15  2013 computing_pres2012.ppt
-rwxr-xr-x  1 russell ftp  1268224 Sep 23 19:44 computing_pres2013.ppt
-rwxr-xr-x  1 russell ftp  1218048 Nov  7  2011 computing_pres2.ppt
-rwxr-xr-x  1 russell ftp    14150 Nov  4  2011 computing_pres_unix_demo.docx
-rwxr-xr-x  1 russell ftp   547381 Nov  7  2011 Intro_to_CBU_computing.pdf
drwxr-xr-x  2 russell ftp     4096 Nov  7  2011 misc
-rwxr-xr-x  1 russell ftp    25088 Sep 23 19:45 Thumbs.db
[russell@l52 compute_pres]$ cp ./.snapshot/hourly.1/computing_pres2013.ppt ./
```

- Every directory contains a (hidden) .snapshot sub-directory
- Cd into the directory containing the file you want to restore
- Choose which snapshot you want to restore
- Copy ./.snapshot/<*snapshot name*>/<*filename*> to current directory

Best Practice

Storage:

- Home space is backed up – hard drives on desktops aren't!
- Try to get into the habit of storing important documents in your network storage space
- Use your home space to store anything you can't easily recreate via a script (documents, figures, scripts) – not derived data / images
- Data is replicated off-site – in the worst case scenario, analyses could be re-created from raw data and a script stored in your home space

Best Practice

Storage:

- Unquota'd doesn't mean infinite…
- Clean up after your analyses – e.g. delete intermediate pre-processing images once you've finished with them
- If you are using AA version 4, make sure garbage collection is turned on
- Don't copy raw data from /mridata or /megata into your /imaging directory
- Don't create multiple copies of the same files
- You can read data from other peoples' imaging space – you don't need to copy data from their space to your own

Central compute machines – old cluster

- Naming scheme = l (for Linux) plus a number – e.g. l22
- 41 machines available: l23 – l63
- Various ages / specifications - generally, higher number = newer = higher spec (also often = busier!)

- Currently operate independently
- No distinction between login and compute nodes – interactive sessions and large compute jobs are all run on the same machines
- No management of which jobs run on which machine
- Can cause problems – e.g. machines can run out of memory

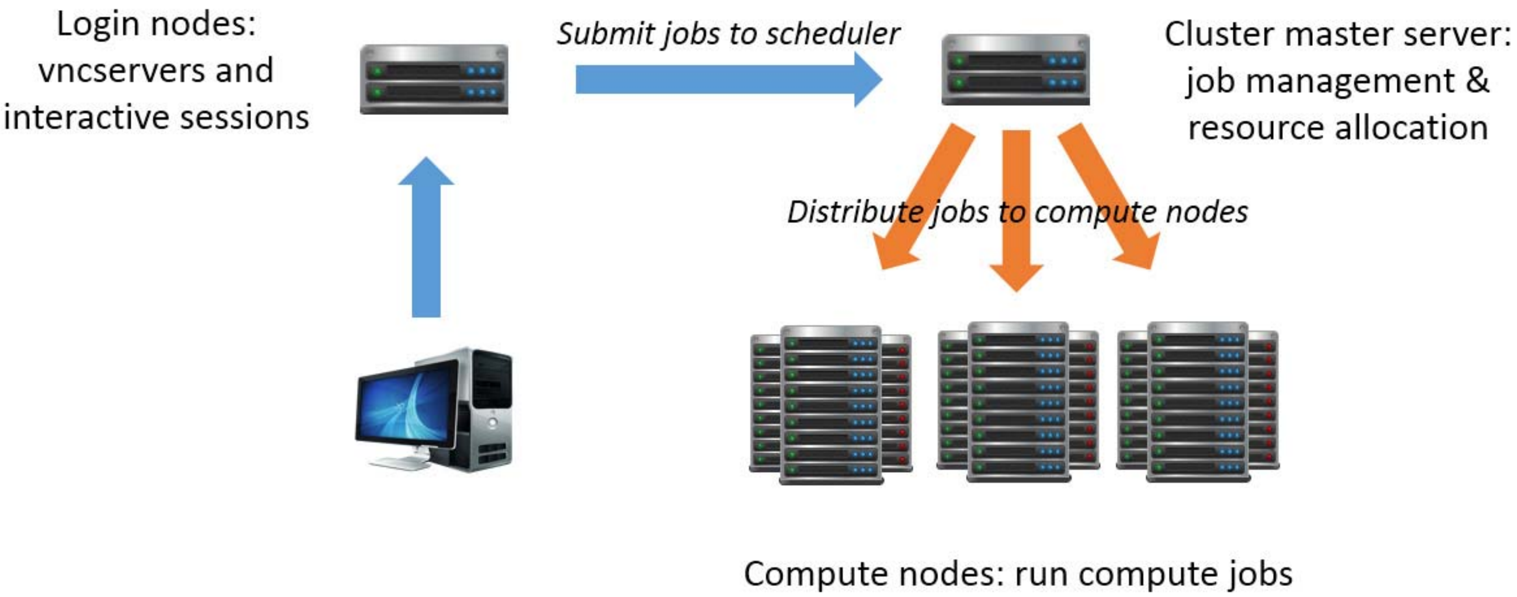
Central compute machines – old cluster

Name	CPU (MHz)	N Cores	RAM (GB)	Open GL graphics	Operating System	
l23 – l34	3.2	4	4	No	Centos 6.4	(64 bit)
l35 - l36	3.2	4	4	No	RHEL 4	(32 bit)
l37 – l41	3	8	16	Yes	Centos 6.4	(64 bit)
l42	3	8	16	Yes	RHEL 4	(32 bit)
l43 – l58*	3	8	16	No	CentOS 5.7	(64 bit)
l59	2.67	12	144	No	CentOS 6.4	(64 bit)
l60 - l63	2.67	12	48	No	CentOS 6.3	(64 bit)

* L48 and l49 have 32GB RAM

284 cores @ ~2.67 GB RAM / core

Central compute machines – new cluster



- Distinction between login and compute nodes
- Login and run interactive sessions on a login node
- Run large compute jobs on compute nodes
- Submit compute jobs to a scheduling system (Torque) that manages allocation of compute resources

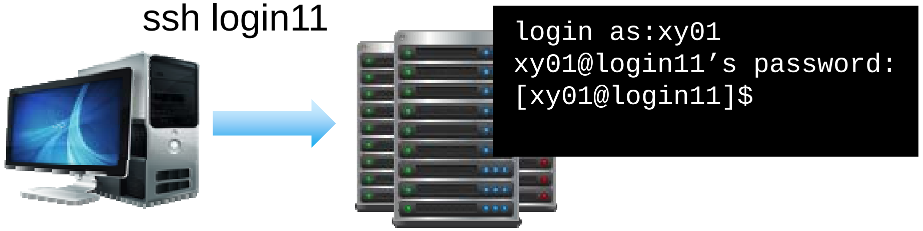
Central compute machines – new cluster

Name	CPU (MHz)	N Cores	RAM (GB)	Open GL graphics	Operating System
Login11,12,14	2.67	12	48	No	Scientific Linux 6.3 (64 bit)
Node-d02-4, 9-18	2.67	12	48	No	Scientific Linux 6.3 (64 bit)
Login13	2.67	16	96	No	Scientific Linux 6.3 (64 bit)
Node-e01-16	2.67	16	96	No	Scientific Linux 6.3 (64 bit)
Node-cc01-7	2.67	16	96	No	Scientific Linux 6.3 (64 bit)
Login-gpu01	2.67	12	48	Yes	Scientific Linux 6.3 (64 bit)
Login-gpu02 – login-gpu03	2	12	64	Yes	Scientific Linux 6.3 (64 bit)
Node-gpu01 – node-gpu02	2.67	16	64	Yes	Scientific Linux 6.3 (64 bit)

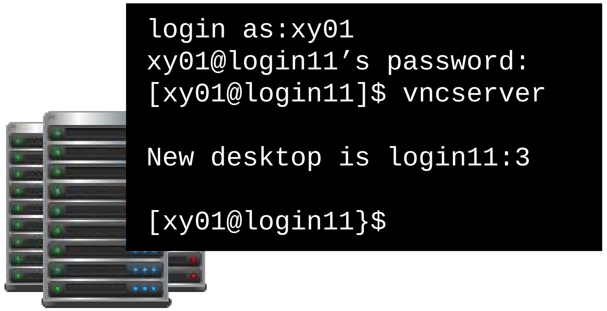
- 644 cores @ ~5.2 GB/core
- Including l59-l63 = 704 cores @ ~5.2 GB/core

Accessing compute machines

ssh login11



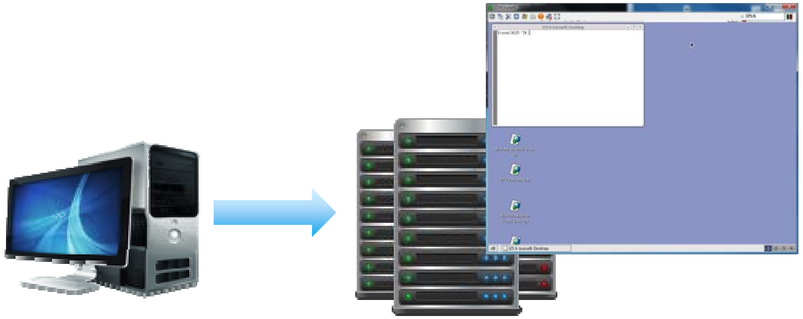
```
login as:xy01
xy01@login11’s password:
[xy01@login11]$
```



```
login as:xy01
xy01@login11’s password:
[xy01@login11]$ vncserver

New desktop is login11:3

[xy01@login11}$
```



vnc viewer

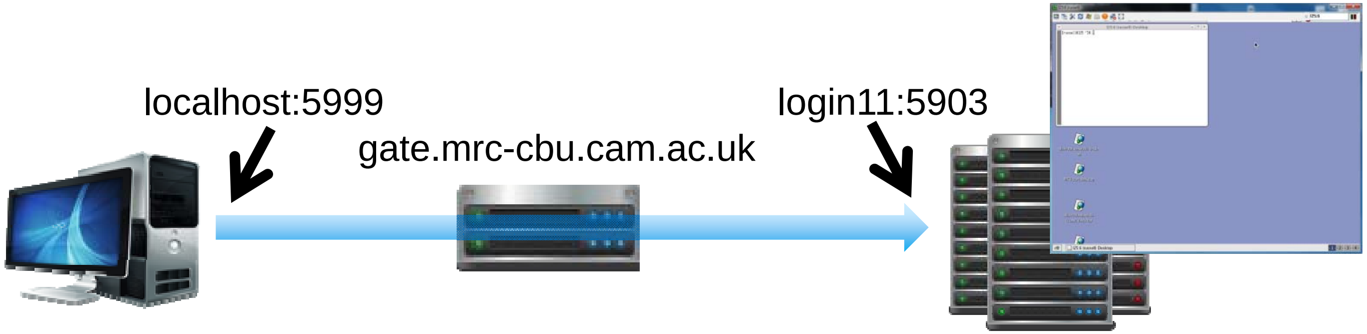
- Compute machines are directly accessible by machines on the CBU network (including those connected via VPN)
- Pick a machine to use
- log in using ssh (**S**ecure **SH**ell)
- On Windows, ssh client = PuTTY
- Text only terminal

- Graphical sessions via VNC (virtual network computing)
- Launch a vnc server, make a note of the desktop number.

- Connect to your vnc server using a vnc viewer running on your local machine.

Accessing compute machines from home

- Compute machines are not directly visible outside the CBU network
- Log into the ssh gateway and ask it to create a connection between a port on your local machine and the port associated with your vncserver on the compute machine (port forwarding; ssh tunnelling)
- Vncserver port number = desktop number + 5900 (e.g. login11, desktop 3 will run on login11:5903
- Connect your vnc viewer to the local port you are forwarding



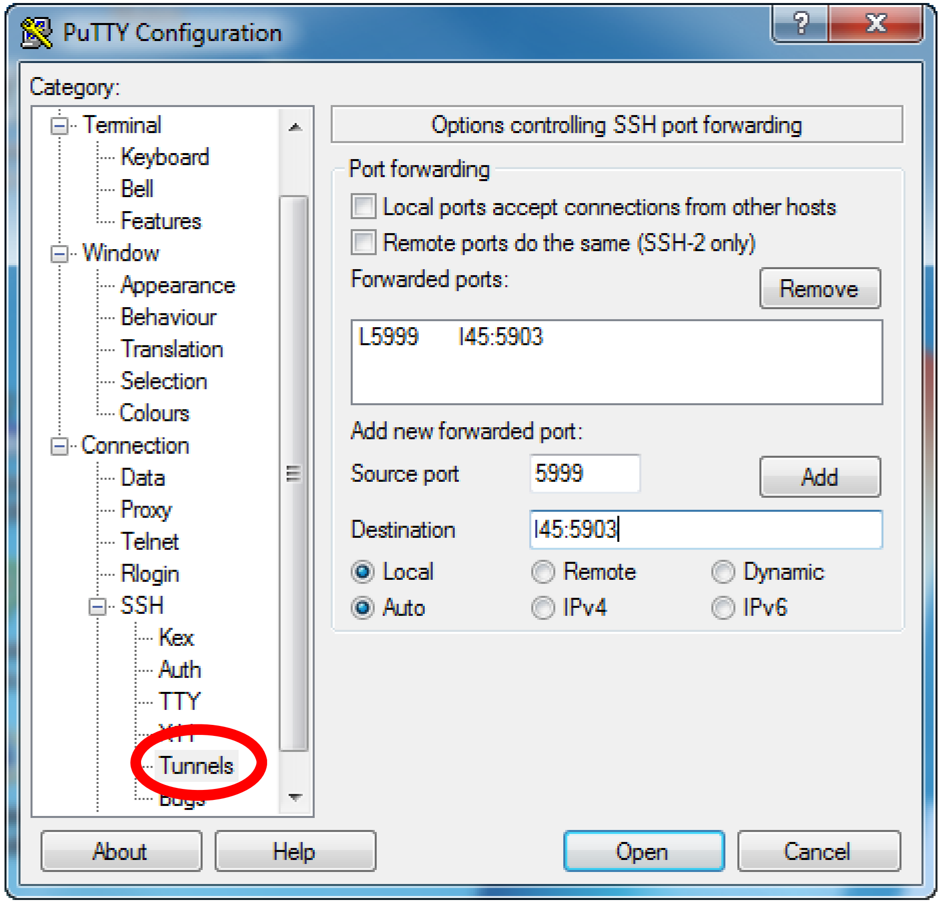
localhost:5999

gate.mrc-cbu.cam.ac.uk

login11:5903

```
ssh –L 5999:login11:5903 xy01@gate.mrc-cbu.cam.ac.uk
```

Accessing compute machines from home



Options controlling SSH port forwarding

Port forwarding

□ Local ports accept connections from other hosts

□ Remote ports do the same (SSH-2 only)

Forwarded ports:

```
L5999      l45:5903
```

Add new forwarded port:

Source port: 5999

Destination: l45:5903

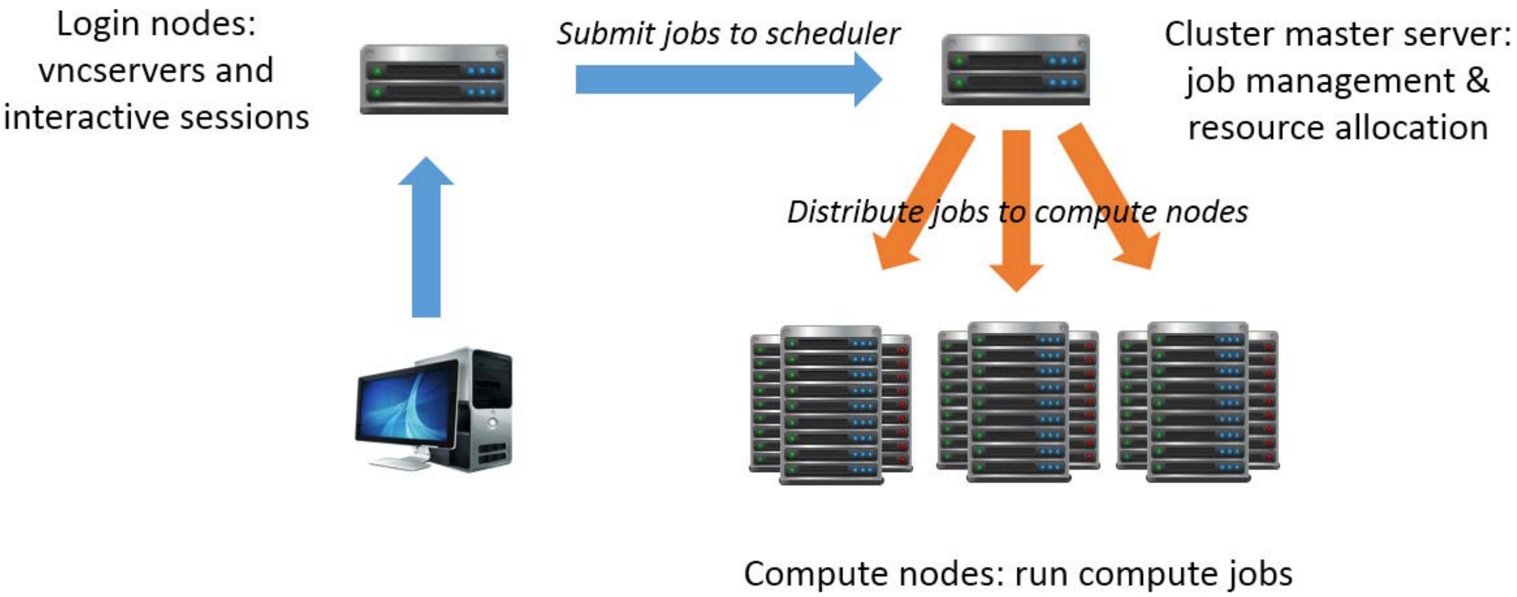
● Local ○ Remote ○ Dynamic

● Auto ○ IPv4 ○ IPv6

Why use a scheduling system?

- Efficient management of resources
- Scheduler determines when and where compute jobs will run, rather than allowing jobs from many users to compete for the same resources
- The scheduler knows what resources are available on each compute machine and will try to make sure they are fully utilized, but not overloaded
- Each node will only run jobs from one user at a time, so any problems will only affect the person that caused them!
- The system supports true parallel processing, e.g. using protocols such as MPI

Using the scheduling system



- Log in to a login node and start a vnc server
- Create a batch script to run your analyses
- Test the batch script and determine what resources it needs (esp. memory and CPU time)
- Submit the script to the scheduling system

Using the scheduling system

- Submitting jobs to the scheduler:

```
qsub <arguments> <command to run>
```

- The qsub command takes numerous arguments, including:
  - -n job name
  - -q which queue to use
  - -o path to file where standard output should be redirected
  - -e path to file where standard error output should be redirected
  - -F list of arguments to pass to job script
  - -l request specific resources, including:

mem=<value kb/mb/gb>	maximum memory required
walltime=<value hh:mm:ss>	maximum job execution time

- E.g.:

```
qsub –q compute –l mem=16gb –F “sub1” my_analysis.pbs
```

submits the script my_analysis.pbs to the compute queue, requesting 16GB memory and passing the argument “sub1” to the script.

Using the scheduling system

- qsub arguments can also be submitted using #PBS directives within the script itself:

```
#!/bin/bash

#PBS –q compute
#PBS –l walltime=12:00:00,mem=16gb

<command 1>
<command 2>
...
```

- This would allow the script to be submitted using simply:

```
qsub my_analysis.pbs
```

Using the scheduling system

- Monitoring jobs:

```
qstat
```

- On it’s own, this gives a list of jobs currently held on the queue:

```
Job ID          Name       User    Time Use  S     Queue
-----------     ---------  -----   --------  -     -----
1520.master01   Job1Task1  ab01    00:58:28  C     compute
1521.master01   Job1Task2  ab01    00:45:03  R     compute
```

Using the scheduling system

- For more information, including the amount of memory requested by each job, and the node on which it’s running:

```
   qstat -n
```

```
Job ID          U’name  Queue   Jobname   Sess ID NDS TSK Req Mem Req Time S Elap Time
-----------     ------  -----   -------   ------- --- --- ------- -------- - ---------
1520.master01    ab01   compute Job1Task   154451  1  --    47gb   12:00   C   00:58
node-e01/0

1521.master01    ab01   compute Job1Task2   50846  1  --    47gb   12:00   R    00:45
node-e02/0
```

Using the scheduling system

- To delete a job from the queue:

```
qdel <job id>
```

- To remove all your jobs from the queue:

```
qselect -u `whoami` | xargs qdel
```

- To place and release a hold on a job:

```
qhold <job id>
qrls <job id>
```

- To alter the properties of a job (e.g. to change some of the resources requested in a qsub command):

```
qrls <job id>
```

Submitting Matlab / SPM jobs

- Use qsub directly (could produce some very long commands…)

```
qsub <qsub arguments> matlab –r <matlab command / script name>
<other matlab arguments>
```

- Use the matlab parallel functions together with CBU-specific wrapper functions
- Use parfor loops / spmd with CBU cluster profile
- Use aa version 4

Submitting Matlab / SPM jobs

- Matlab Distributed Computing Server (DCS) and Parallel Computing Toolboxes provide functions for running matlab jobs in parallel over multiple compute nodes / CPU cores
- DCS supports 3rd party schedulers such as Torque
- In general, the procedure for submitting to a scheduler using DCS is:
  - Create a script to run your analysis
  - Create a scheduler object using the DCS functions
  - Configure the scheduler object (define properties such as queue name, resources required,etc).
  - Configure the scheduler with a list of jobs to run
  - Call a submit method to submit the jobs
- When using a Torque / PBS scheduler object, Matlab translates the properties of the scheduler into a series of qsub commands.

Submitting Matlab / SPM jobs

- 2 cbu-specific functions have been created to simplify the process submitting to the CBU cluster via DCS:
  - cbu_scheduler – creates and configures a scheduler object
  - cbu_qsub – submit jobs to the queue

```
subjects={'CBU130001','CBU130002};

clear J;
for s=1:size(subjects,2)
   J(s).task=str2func(my_analysis_script);
   J(s).n_return_values=0;
   J(s).input_args=subjects(s);
   J(s).depends_on=0;
end

clear S;
S=cbu_scheduler();
cbu_qsub(J,S,[]);
```

Loop through all subjects. For each subject, add an entry to a structure array containing details of the analysis to run

Create a scheduler object. Without any other arguments, cbu_scheduler will return a default configuration

Submit the jobs

Parfor and spmd

- Need to open a matlabpool on the cluster, rather than on the local host
- In matlab 2012a onwards:

```
P=parallel.importProfile(‘/hpc-software/matlab/cbu/CBU_Cluster.settings’);
matlabpool(P)
```

- To modify the properties of the CBU_Cluster profile:

```
P=parallel.importProfile(‘/hpc-software/matlab/cbu/CBU_Cluster.settings’);
P=parcluster(P);
P.<property>=<value>;
matlabpool(P)
```

- Once the matlabpool is open, parfor and spmd should work in the same as they would with a local pool

Best Practice

Old cluster and login nodes:

- The machines are a shared resource – think about other users when you're using them
- You should only ever need one or two vnc sessions – re-use old vnc sessions (they will persist until the host machine is rebooted), or kill vnc sessions if you know you won't need to use them for a while
  - ssh *machine-name*
  - vncserver –kill :*desktop-number*
- Close SPM/Matlab when you have finished using them, especially if your session has been using a lot of memory.
- Run jobs that will take a lot of resources (e.g. parallel jobs using multiple cpu cores) at quiet times (overnight, at weekends etc)
- If your job crashes a machine, let computing know – please don't launch it again on another machine and hope for the best!
- Please don't run large compute jobs or matlabpools on the login nodes!

Best Practice

Scheduling system:

- Develop and debug your scripts on the login nodes before submitting to the scheduler
- Make a note of the resources your job requires – especially memory and cpu time
- Requesting the appropriate resources allows the scheduling system to operate most efficiently. The scheduler will try to launch as many jobs on each machine as possible, without overloading that machine
  - Under-requesting (e.g. requesting 4GB RAM when you need 16GB) can cause the machines to run out of memory and become unresponsive
  - Over-requesting (e.g. requesting 64GB RAM when you only need 16GB) means fewer jobs will run simultaneously

Scientific software

- /imaging/local
- Readable by everyone, writeable by members of imagers_devel
- Older software tends to be in /imaging/local/linux, newer versions in /imaging/local/software
- SPM:
    - Pre SPM 8: /imaging/local/spm
    - SPM 8: /imaging/local/software/spm_cbu_svn
- Main exception is Matlab – that's installed on each machine individually
    - /usr/local/matlab/*version*/bin/matlab

  Some applications use load balancing scripts to find the machine with the lowest load

Scientific software

- Packages are managed by specific scientists:

EEG Lab	Jason Taylor, Tristan Beckinschtein
Fieldtrip	Maarten Van Casteren, Lucy MacGregor, Olaf Hauk
Freesurfer	Kristjan Kalm, Marta Correia, Olaf Hauk
FSL	Marta Correia
Camino	Marta Correia
Real time fMRI	Marta Correia, Danny Mitchell
MNE	Olaf Hauk
SPM	Rik Henson
AA	Danny Mitchel
Neuromag	Yury Shtyrov
R	Dennis Norris
Python	Dennis Norris
C++	Maarten Van Casteren, Dennis Norris

http://imaging.mrc-cbu.cam.ac.uk/imaging/AvailableSoftware

Scientific software

- Many packages are launched using wrapper scripts
- These will parse any options (or use default values), set any necessary paths, and then launch the application itself
- The paths to the wrapper scripts are configured in various login scripts – e.g. /imaging/local/ (everyone), or /home/*username*/.cshrc (specific to each user)
- Provided your login scripts are working, the wrapper scripts themselves should be on your path, so you don't have to remember the location
- You can configure your own .cshrc file to use specific versions of the a software package, set custom paths etc.
- Some wrapper scripts also use load balancing functions – they will find the machine with the lowest overall load and launch your job there
- You can also use the "showload" command to display various load statistics from each machine and use that to choose a machine

Scientific software

Specific wrapper scripts:

- spm
  - Takes various (optional) arguments, including spm version, matlab version, machine to use, etc
  - On its own, “spm” will launch spm 8 using matlab r2009a on the lowest load machine.
  - “spm 5 l45 matlab2009a” will launch spm 5 using matlab r2009a on l45
- Matlab
  - Different machines have various different versions installed
  - Default version launched by “matlab” command
  - Links to different versions are in /user/local/bin
  - ls –l /usr/local/bin/matlab* will show all links (and matlab versions) available on a particular machine.

Further Information

Imaging wiki:

http://imaging.mrc-cbu.cam.ac.uk/

Computing group intranet page:

http://intranet.mrc-cbu.cam.ac.uk/computing/

Detailed guide to accessing a vnc session from outside the CBU:

http://intranet.mrc-cbu.cam.ac.uk/computing/putty/Remote%20VNC%20Session%20v4.pdf

Introduction to Unix commands:

http://www.ee.surrey.ac.uk/Teaching/Unix/

Unix Demo

- Outline unix type filesystem – no drive letters, directories can be local and network shares, file tree root = /, names are case sensitive
- CBU specific directories - /home, /imaging, /group, /mridata
- Snapshot directories - /home/*username*/.snapshot
- Useful commands – ls, mkdir, cp, mv, rm, find, which, ps, top
- ssh to log in to other machines
- Permissions
- Paths, PATH environment variable
- Symbolic links
- Login scripts, /home/*username*/.cshrc, /home/*username*/.login