The Linux-based Computer Clusters in the Theoretical and Computational Physics Group, School of Physics, USM

Yoon Tiem Leong
Theoretical and Computational Physics Research Group, USM

23 August 2011

ABSTRACT

The first Linux-based computer cluster had been established in the theoretical and computational physics group around the end of year 2010. These Linux-based computer clusters are customised computing facilities used to perform computationally intensive parallel computing calculations. In this workshop, which is intended for novice users, we will discuss how to access these clusters remotely, along with some basic operational knowledge in Linux commands and GUI applications. Some motivations of setting up a Linux system, instead of Windows, will also be explained.

What is a computer cluster?

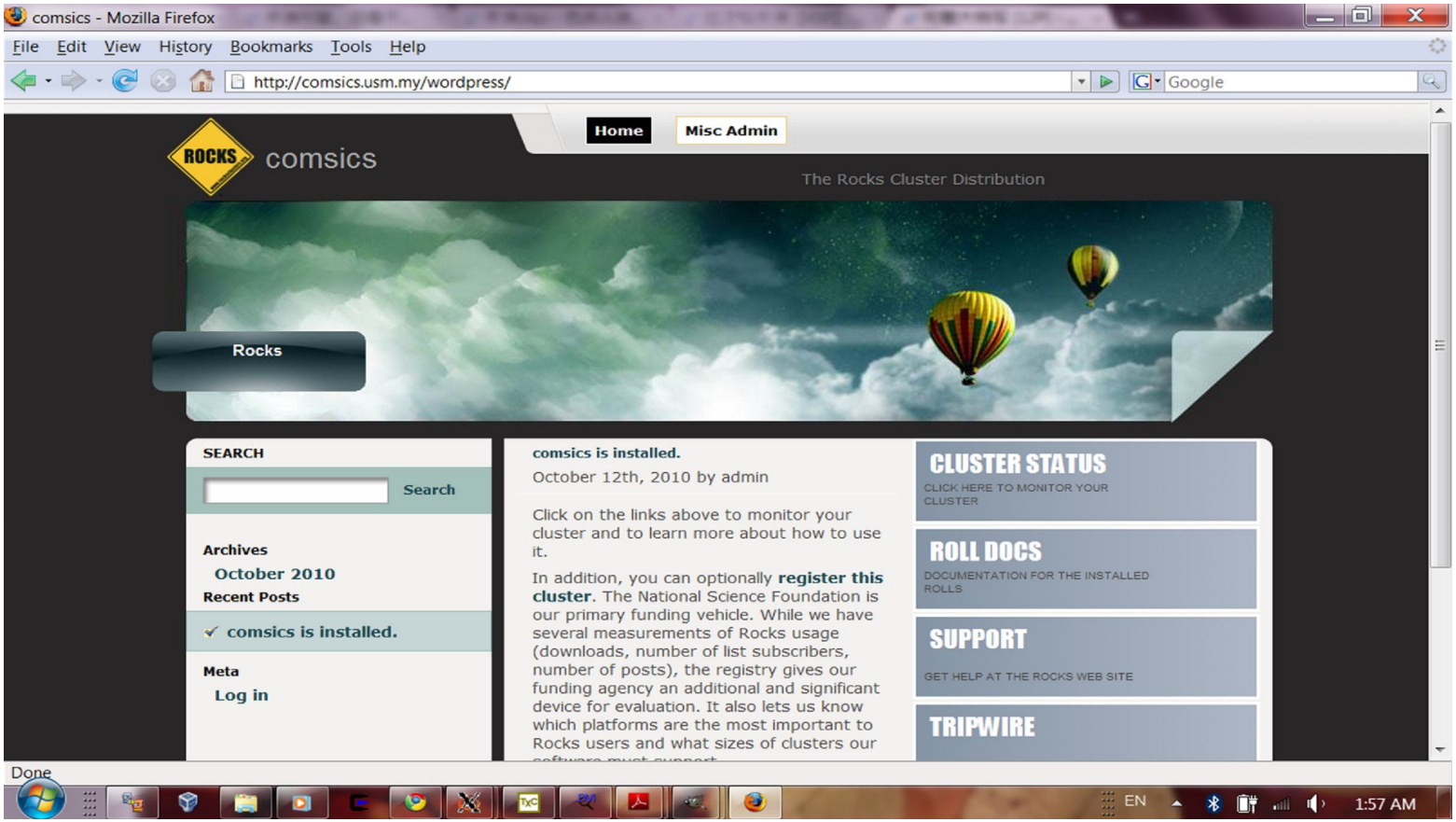
- Computer cluster - computers linked up by LAN wires, allowing direct (passwordless) inter-PCs communication. Mainly for running MPI parallel computing purpose.
- Centralised system management - convenient for a large group of PC and users.
- Cheapest way to get high computational resource per dollar.
- Ours are built by linking up PCs using LAN wires.
- Others rack-mount cluster - slots of blade node in customised metal cabinet.

Rack-mount cluster

The computer clusters available or being planned

- They are all running Rocks Clusters, a type of Linux specifically designed for clustering purpose.
- comsics - 20 PCs (Acer, first generation i5 processors). A total of $20 \times 4 = 80$ CPUs. Already up and running.
- anicca - 30 PCs (Dell, Core2 Quad processors). A total of $30 \times 4 = 120$ CPUs. Currently only 8 nodes running. In the process of being configured.
- anatta - 5 PCs (Inhomogeneous machines). NVIDIA GPU enabled. Planned and is to be realised in the near future.
- The states of these clusters can be monitored online at (when they are up and ready)
- comsics.usm.my (comsics, accessible worldwide)
- anicca.usm.my (anicca, accessible worldwide)

The comsics cluster page



The comsics cluster (photo)

Figure: comsics cluster, located in the Computer Lab, Physics School, USM

The anicca cluster (photo)

Figure: anicca cluster, located in the Theory Lab, Physics School, USM

The anicca cluster (photo)

Figure: anicca cluster, located in the Theory Lab, Physics School, USM

Why you want to use Windows?

- Bill Gates' successful marketing
- Convenient for "dummy" ("click on icons and run")
- Easily available pirated softwares (morality not an issue)
- Comfort zone
- No Linux community for technical support and awarness.

The main hinderence against using clustered Linux

- Users prefer Windows over Linux
- Addicted to pirated Windows software
- Gamming is best with Windows
- Not familiar with Linux
- Prejedice against Linux
- Refuse to learn Linux

Why I don't use Windows?

- It costs $$$ (morality an issue) (networking, Windows Server)
- I lack the technical knowledge to build a Windows-based cluster.
- Many free computational physics software do not have Windows version - e.g., LAMMPS, ABITNIT

Why Linux?

- Because it is not Windows.
- Free softwares, libraries, utilities and applications (gnuplot, fortran, c++, OpenOffice, Python, Java, ATLAS, Lapack, BLAS, etc).
- Programmable (shell scrpit) - allows batching of programmes / codes.
- Makefile for c++ and fortran programming.
- many important free computational physics software only have Linux version.
- Customise the action you wish (instead of relying only on the software)
- E.g., download a load of vedeo files automatically from ictp.tv and put them into a desired folder - done using shell script).
- etc ...

Why Rocks Linux?

- Best for building cluster.
- Best for system maintainence.
- Sharing of software resources.
- Parallel computing - all CPUs are fully utilised.
- Most easily scalable (adding new nodes demands almost no effort)

So, why should I use the cluster as I already have my own PC?

- To run parallelised fortran codes or Mathematica or Mathlab.
- To run the preinstalled software packages such as VASP, ABINIT, AMBER, LAMMPS, WIEN2k, CRYSTAL, DFTB+, HOTBIT, GAUSSIAN, GEANT4 etc.
- Remote access from all over the world.
- Access the GPU-enabled PC in anatta remotely (in plan).
- Best option for system admin
- Optimum share of computing resources for best computational performance (compared against same number of networked, stand-alone PCs).
- Using the software and coding functionalities offerred by the the clusters in a more advanced manner is the topic for another workshop.

The main purpose to build a computer cluster: Parallel computing

- Imagine you have 100 non-interacting codes to run independently, each has a different set of parameters.
- You can execute this massive job automatically instead of manually, by writting a "batch script".
- In this script, you will ask the computer to locate which cpu within the cluster is free, and then send a job to execute there. This provides a simple way to tap the full computing power available in a pool of computing resources in a most convenient way.
- This is merely to conveniently way to manage a large amount of individual calculation to be distributed across the cluster.
- Yet this is not "parallel computing" as it is supposed to mean.
- The computing time required by a single program to complete does not scales with the number of cpu available.

The main purpose to build a computer cluster: Parallel computing

- Imagine you have an "expensive" code that contains a section that is parallelible. It may takes 10 hours to finish in a one-node PC.
- E.g., a molecular dynamics simulation with 10^6 molecules or diagonalisation of a very large matrix.
- Parallel computing allows the computing time to scale with number of cpu available. This allows a single program to finish in a much shorter time, say, 1 hours, if the code is to run in parallel in a cluster with 50 cpu's.
- This is the more "authentic" way parallel computing.
- Scalability is a hallmark of paralel computing.

userid and password

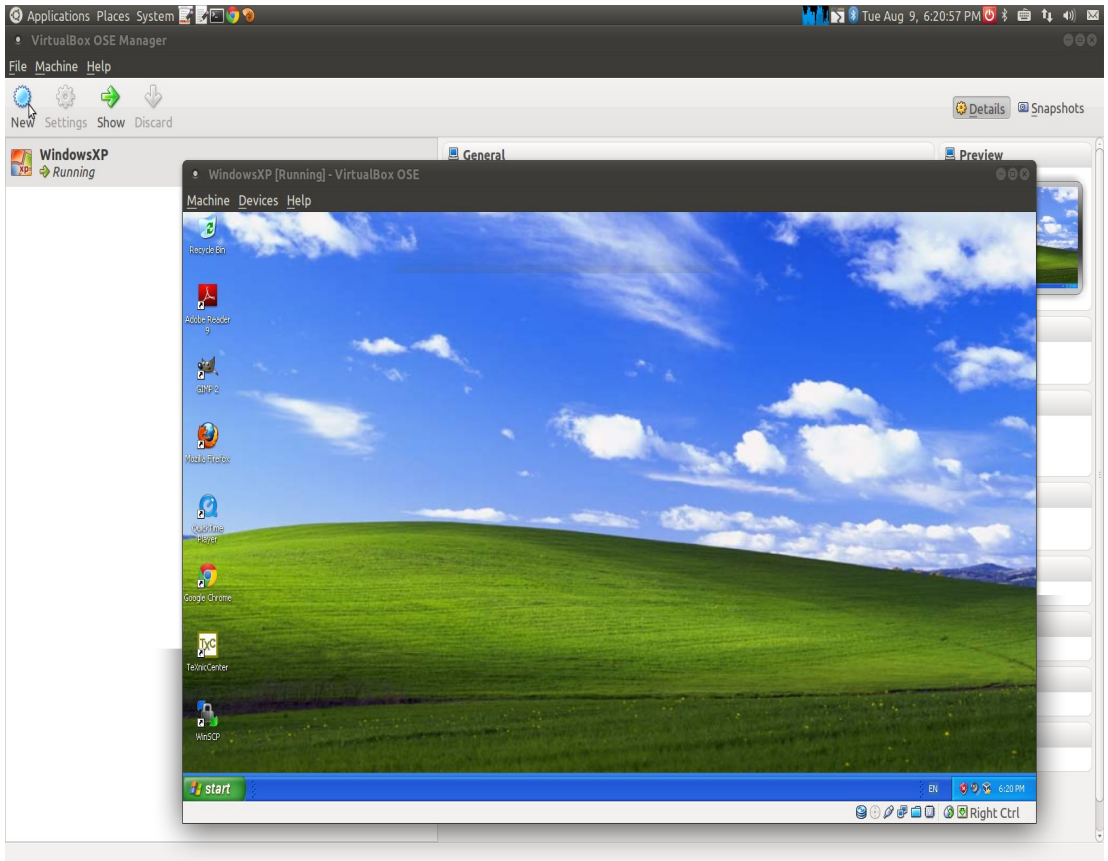
- In order to gain access to these clusters, you must has an account in each of them.
- Apply for an account in these cluster to the administrator via email to `tlyoon@usm.my`.
- If you have not obtained an account, a generic userid and password you can try is `acerX` with password 123 for comsics (`X` stands for interger ranging from 01, 02 to 19); To access anatta, try `acer20` with password 123.

Running WindowsXP in Linux

- It is possible to run Windows XP in Linux. This is done via "virtualisation".
- A virtual machine is a software that impliment the "virtualisation".
- In comsic the "virtual machine" installed is called VirtualBox.
- The virtual WindowsXP client is installed in the VirtualBox host.
- The virtual WindowsXP is actually a large file to be found in `/state/partition1/repo/VBanicca/.VirtualBox/` or `/state/partition1/repo/VBanicca/.VirtualBox/HardDisks/` in all compute nodes.

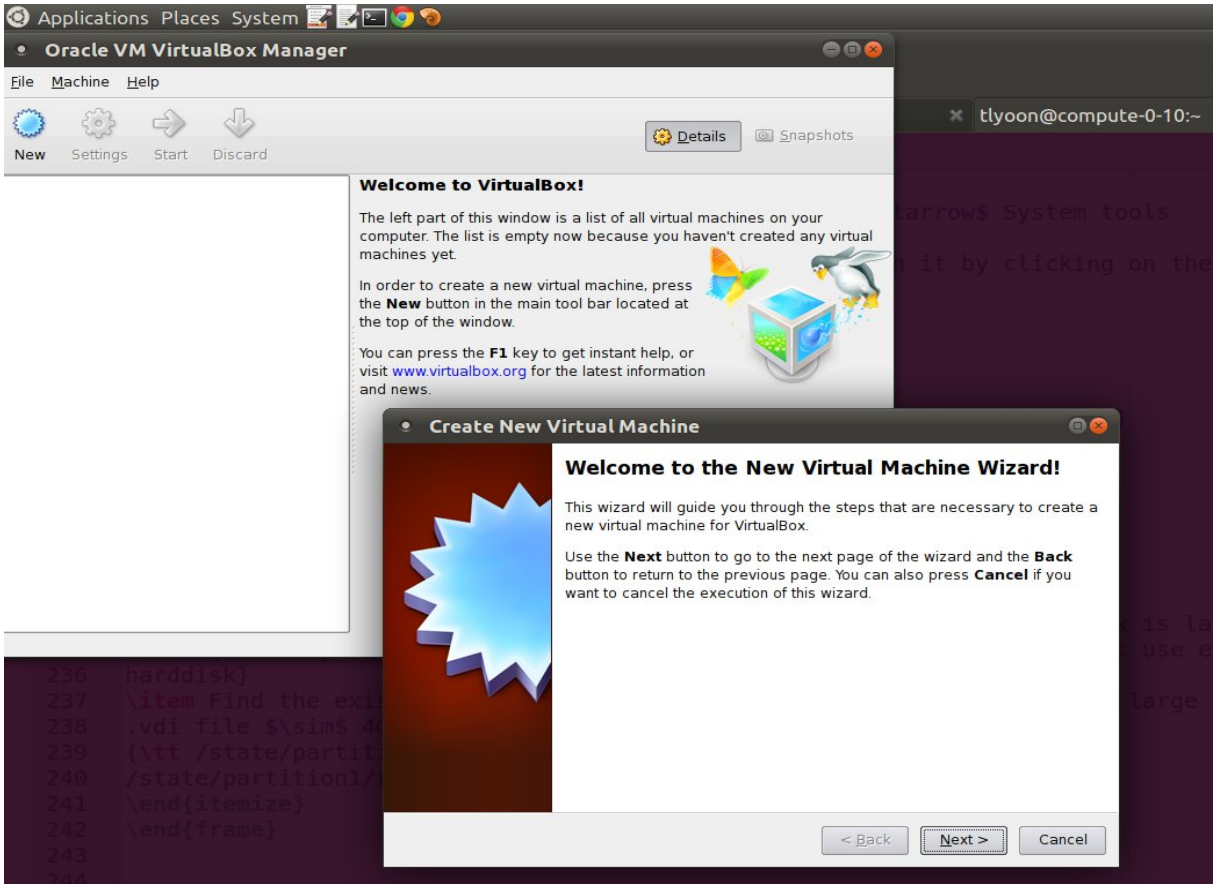
Launching a Windows XP session in a compute node

- Assume you are using compute-0-x.
- Launch Virtual Box by clicking Application → System tools → VirtualBox.
- If a Virtual WindowsXP is already installed, lauch it by clicking on the WindowsXP icon at the left upper corner.

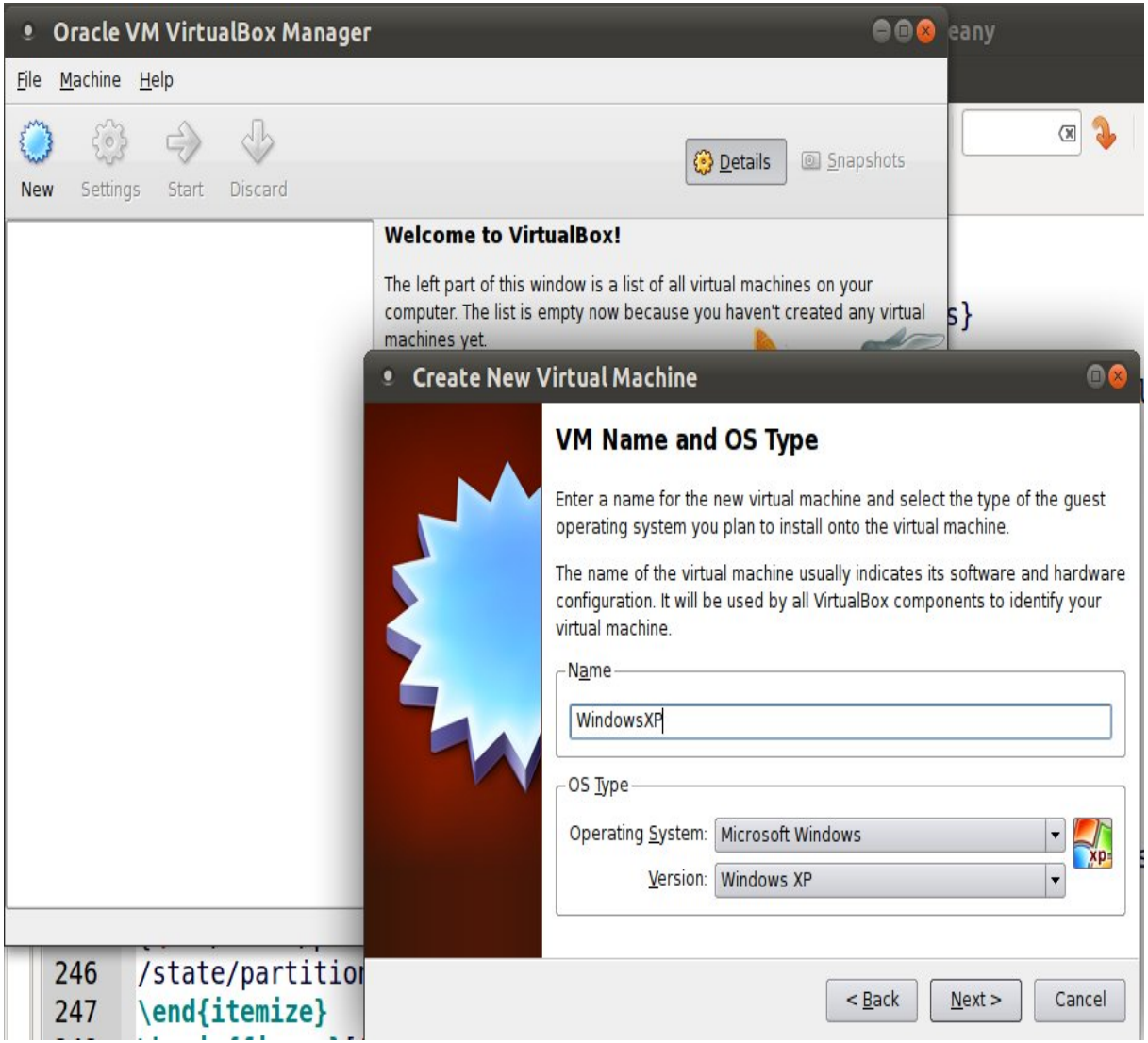


Setting up your Virtual Windows 1

- If no Virtual WindowsXP already set up, do it yourself:
- First, click on the `new` button.



Setting up your Virtual Windows 2

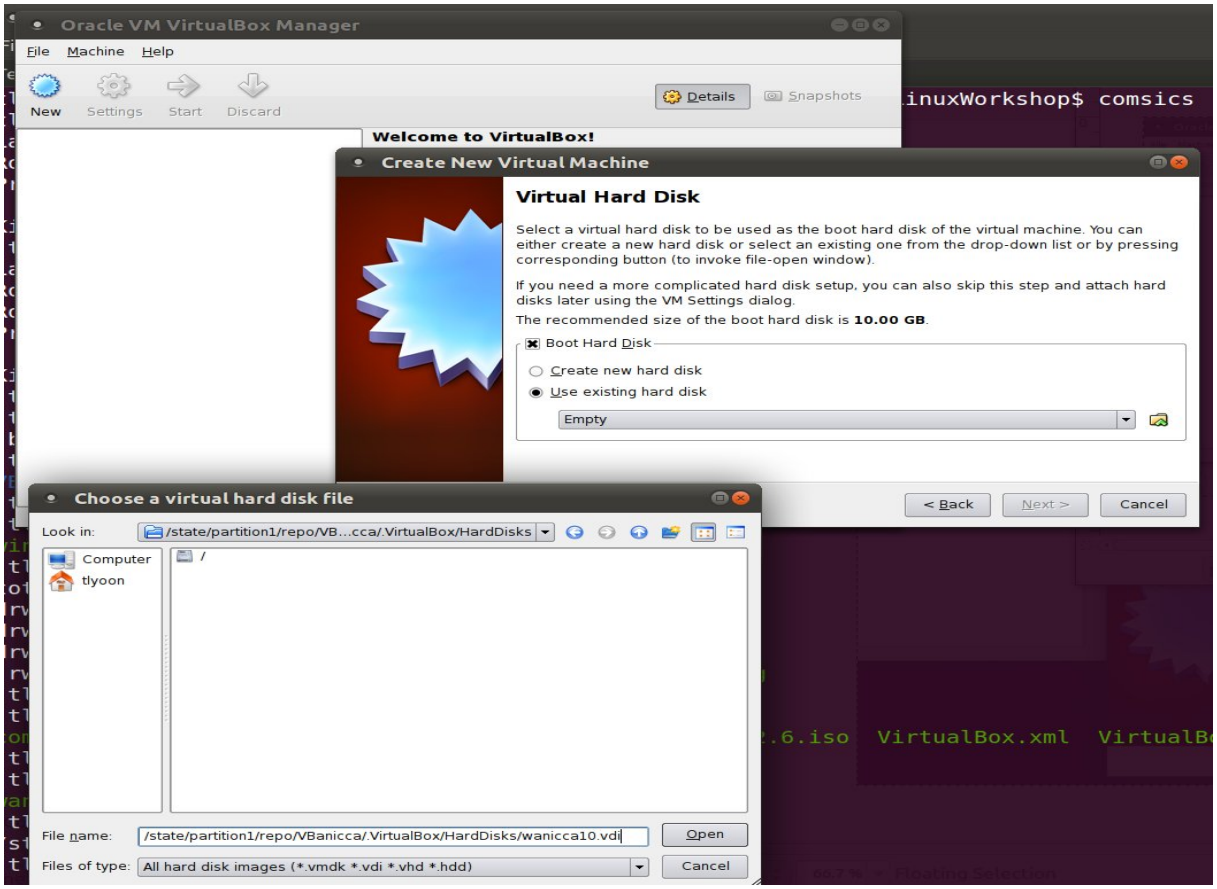


Setting up your Virtual Windows 3

Find the existing virtual WindowsXP "harddisk" file (which is a large .vdi file $\sim$ 4GB) in your local folder `/state/partition1/repo/VBanicca/.VirtualBox/` or `/state/partition1/repo/VBanicca/.VirtualBox/HardDisks/`

Setting up your Virtual Windows 4

Find the existing Virtual Hard Disks (which is a large .vdi file $\sim$ 4GB) in your local folder `/state/partition1/repo/VBanicca/.VirtualBox/` or `/state/partition1/repo/VBanicca/.VirtualBox/HardDisks/`

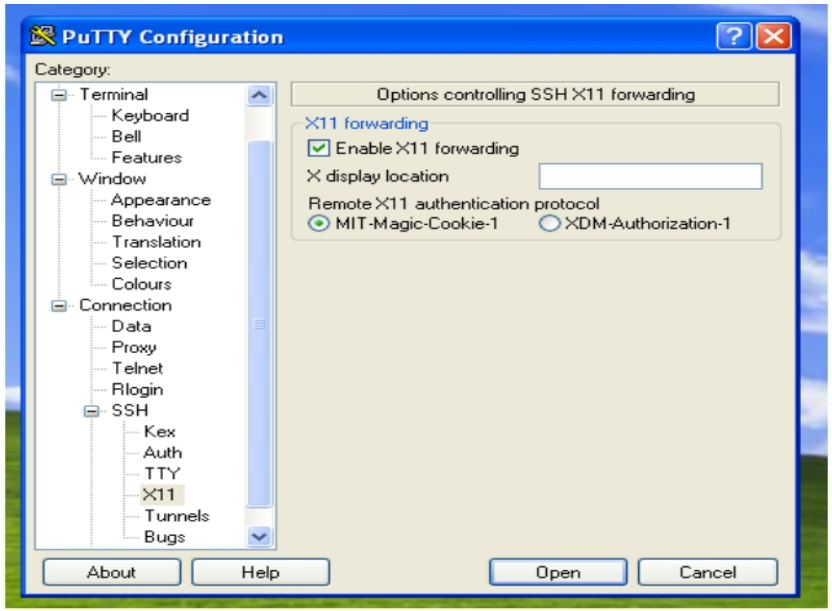


Tools to access the Linux Clusters remotely from a Windows machine

- In Microsoft Windows, you need to install Xming, WinSCP (or Filezilla), Putty. These are small freeware downloadable from
  http://www2.fizik.usm.my/tlyoon/Downloads/.
- Winscp or Filezilla are used to transfer files via your local hardisk and the cluster.
- Launch Winscp or Filezilla from your local Windows machine to see whether you can login to comsics.usm.my and tranfer files to and fro between the cluster and your local hardisk.

Tools to access the Linux Clusters remotely from a Windows machine: `ssh`

- Launch Xming. Then launch Putty.
- Configure Putty so that it uses Xming when ssh into the remote computers:



To access anicca or anatta remotely,

```
ssh -X -Y comsics.usm.my
```

in the Putty terminal.

Accessing the Linux Clusters remotely from a Linux machine with ssh

- If you are accessing the clusters from a Linux machine:
- In the terminal, issue the command

  ```
  ssh -X -Y comsics.usm.my
  ```

  to remotely login to comsics.

The Linux Terminal

```
[tlyoon@anicca ~]$ hostname
anicca.usm.my
[tlyoon@anicca ~]$ hostname --ip-address
10.205.19.225
[tlyoon@anicca ~]$ uname -a
Linux anicca.usm.my 2.6.18-164.6.1.el5 #1 SMP Tue Nov 3 16:12:36 EST 2009 x86_64 x86_64 x86_64 GNU/Linux
[tlyoon@anicca ~]$ date
Mon Aug  8 12:43:13 MYT 2011
[tlyoon@anicca ~]$ whoami
tlyoon
[tlyoon@anicca ~]$ id
uid=500(tlyoon) gid=100(users) groups=100(users)
```

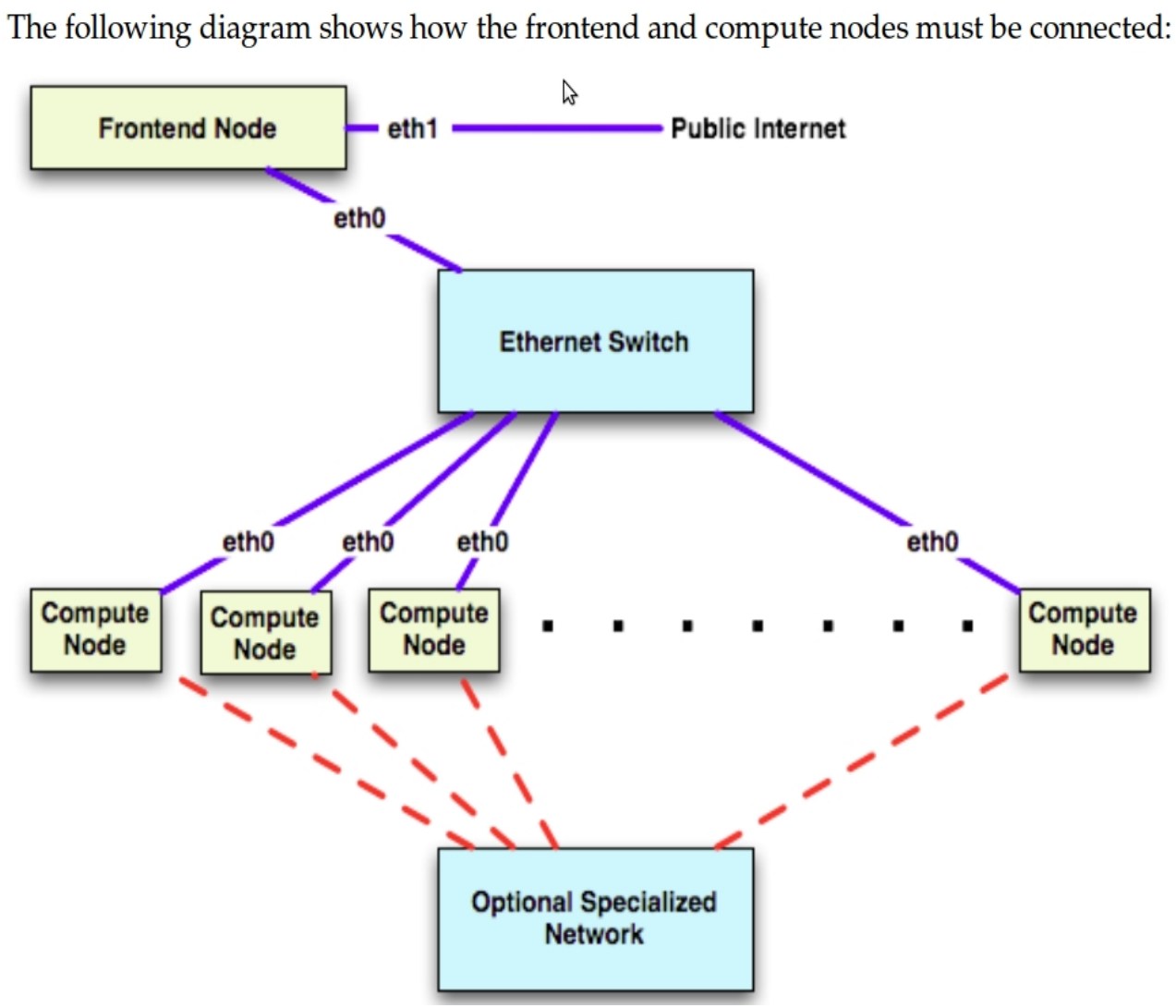
- Each user belongs to one or more `group`. In the above example, `tlyoon` belongs the the group `users`.
- By default, every user in comsics belong to the group `users` too.

The hierachy of a Rocks cluster

- Right after ssh into a cluster, you actually land in the frontend of that cluster.
- A Rocks cluster comprised of a single frontend directly connected to the internet, and compute nodes connected to the frontend via LAN wires.
- These compute nodes (or simply, 'nodes') are not directly accessible from the internet. They can be only accessed through the frontend.
- See figure "The hierachy of a Rocks cluster"
- You can check which cluster you are in by requesting the host's name, by issuing the linux command "`hostname`"

The hierarchy of a Rocks cluster

Figure: The hierachy of a Rocks cluster



Who's here, and what's running

- When you first ssh into comsics, you will first land on the frontend.
- Check your hostname with `hostname`
- See who is in the current host using `who`
- See what process is running in the current host using `top`
- Press `q` or `ctrl Z` to terminate `top`.

Killing a process in top

- You can kill a process by typing `k` while `top` is running
- You will be prompted `PID to kill:`
- Key in a particular process's PID, and you will be prompted `Kill PID xxx with signal [15]`
- Choose 9 as the highest level of killing.
- Press `q` or `ctrl Z` to terminate `top`.
- Warning: Don't do it now.

Who's here, and what's running

```
[tlyoon@anicca ~]$ who
tlyoon   pts/1        2011-08-08 12:40 (10.205.18.57)
[tlyoon@anicca ~]$ top

top - 14:12:53 up 8 days, 14:10,  1 user,  load average: 0.00, 0.01, 0.00
Tasks: 159 total,   1 running, 158 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.2%us,  0.4%sy,  0.0%ni, 99.4%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:   3035264k total,  2926984k used,   108280k free,   366344k buffers
Swap:  3068372k total,        0k used,  3068372k free,  2110452k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 3920 nobody    15   0  213m 2204 1020 S  0.7  0.1  15:45.60 gmetad
 4514 nobody    18   0  239m  60m 3284 S  0.7  2.0  14:28.22 gmond
 3757 root      15   0  232m  11m 2392 S  0.3  0.4   0:49.53 greceptor
    1 root      15   0 10352  672  568 S  0.0  0.0   0:05.96 init
    2 root      RT  -5     0    0    0 S  0.0  0.0   0:00.17 migration/0
    3 root      34  19     0    0    0 S  0.0  0.0   0:00.01 ksoftirqd/0
    4 root      RT  -5     0    0    0 S  0.0  0.0   0:00.00 watchdog/0
    5 root      RT  -5     0    0    0 S  0.0  0.0   0:00.97 migration/1
    6 root      34  19     0    0    0 S  0.0  0.0   0:00.14 ksoftirqd/1
    7 root      RT  -5     0    0    0 S  0.0  0.0   0:00.00 watchdog/1
    8 root      RT  -5     0    0    0 S  0.0  0.0   0:00.17 migration/2
```

Check what is your current directory `pwd`

- Right after ssh into a cluster, you will land in your home directory in that cluster.
- If you are user `acer10`, your home directory will be in `/home/acer10`.
- Check this our yourself by issuing the linux command `pwd` (present work directory).
- By default, the data stored in your home directory `/home/acer10` are physically kept in the frontend's hardisk irrespective of your current host.
- E.g., you may download and save `myfile.dat` a file in compute-0-1 in `/home/tlyoon`. This file is actually being saved under the directory `/export/home/tlyoon` physically found in the frontend's hardisk.

Structure of the compute node's hardisk

- The `/home/acer10` directory as seen in a local node (say compute-0-10) is actually not sitting in the local hardisk but in the frontend's. Everything stored in this directory will be saved in frontend's hardisk.
- The local hardisk is actually sitting in `/state/partition1/`. If you save your files here, you may not be able to see it unless you nevigate to `/state/partition1` of compute-0-10.
- In other words, you won't find your file if you go to the wrong place, e.g., `/state/partition1` of compute-0-11 or in the frontend.

Intra-cluser navigation

- Once you login to a frontend, you can easily nevigate to the nodes in that cluster using `ssh -X -Y compute-0-x`, where `x` stands for some positive interger. This can be done without being requested for a password.
- You can check which nodes are up or down by browsing the webpage comsics.usm.my/ganglia using any web browser.
- Try `ssh -X -Y compute-0-3`. Issue `hostname` and `pwd` to check your current location.

```
[tlyoon@comsics ~]$ ssh -X -Y compute-0-11
Last login: Tue Aug  9 20:50:05 2011 from comsics.local
Rocks Compute Node
Rocks 5.3 (Rolled Tacos)
Profile built 15:49 27-Jul-2011

Kickstarted 15:57 27-Jul-2011
[tlyoon@compute-0-11 ~]$ hostname
compute-0-11.local
[tlyoon@compute-0-11 ~]$ pwd
/home/tlyoon
```

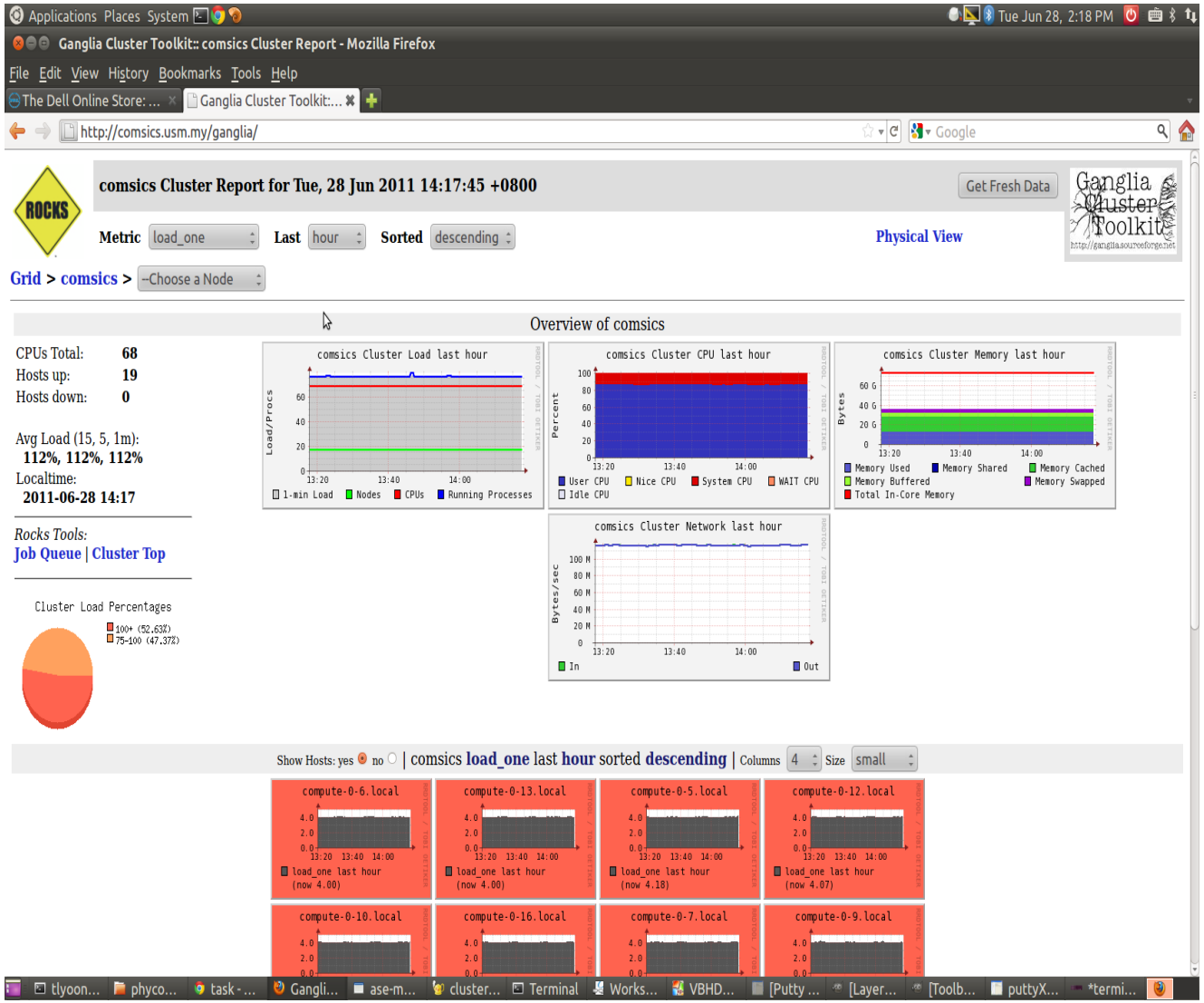
Avoid runing jobs in the frontend

- If you only wish to run a single machine job, e.g., a Mathematica session, try to not run it in frontend.
- Browsing the ganglia comsics.usm.my/ganglia to check which compute node is 'free'.
- ssh to that free node to run your Mathematica session - frontend too full may cause crashing of the whole cluster (it happens!)
- If a compute node is not "free " (like the one below), go run your job in other free node.

```
[tlyoon@compute-0-11 ~]$ who
tlyoon   pts/0        2011-08-09 20:50 (comsics.local)
[tlyoon@compute-0-11 ~]$ top

top - 20:55:48 up 4 days,  5:59,  1 user,  load average: 4.00, 4.00, 4.00
Tasks: 126 total,   6 running, 120 sleeping,   0 stopped,   0 zombie
Cpu(s): 64.4%us, 30.9%sy,  0.0%ni,  0.0%id,  0.0%wa,  0.0%hi,  4.7%si,  0.0%st
Mem:   4002960k total,  1250444k used,  2752516k free,   211136k buffers
Swap:  1020116k total,        0k used,  1020116k free,   590528k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
19696 christop  25   0 2106m  60m 3520 R 100.2  1.5   1197:18 lmp_g++
19840 christop  25   0 2106m  59m 3520 R 100.2  1.5   1182:18 lmp_g++
20158 christop  25   0 2106m  59m 3520 R 100.2  1.5   1143:28 lmp_g++
19576 christop  25   0 2106m  59m 3520 R 99.9  1.5   1211:58 lmp_g++
  518 root      10  -5     0    0    0 S  0.3  0.0   1:57.91 ata/2
30877 tlyoon    15   0 12744 1112  824 R  0.3  0.0   0:00.03 top
```

Ganglia

Figure: The screen shot illustrating the states of the comsics cluster



Common Linux command: `ls`

- `ls` to list all the files in the current directory with minimal information.
- `ls -l` to list all the visible files in the current directory. It also display the files/direcotories' permission and their owner/group.
- Permission and ownership are in the form of

  `drwxrwxrwx tlyoon users`

  .
- `ls -lah` to list all the files in the current directory, including the hidden files. The file size is displayed in human-readble unit (kB or MB or GB).
- `ls -lah` to list all the files in the current directory, including the hidden files. The file size is displayed in human-readble unit (kB or MB or GB).

Common Linux command: `cat`, `cd`, `mkdir`

- `cat filename` display the file's content on the terminal.
- `cd` change directory.
- `cd ..` change to a directory at a higher level.
- Example:

```
tlyoon@compute-0-3:~$ pwd
/home/tlyoon
tlyoon@compute-0-3:/home/tlyoon cd ..
/home/
tlyoon@compute-0-3:mkdir /home/tlyoon/temp
tlyoon@compute-0-3:cd /home/tlyoon
tlyoon@compute-0-3:/home/tlyoon$
tlyoon@compute-0-3:ls
temp
```

Launching GUI applications using command line: `nautilus`

- Launch a GUI (Graphical Utility Interface) 'file explorer' in Linux by issuing the command `nautilus --browser`
- `Ctr + l` to edit the location bar of nautilus. Now you manipulate your files easily using nautilus.
- nautilus can also be used to transfer files from compute nodes to anywhere in the USM network using the "`Connect to server`" function. Choose `ssh` from the drop-down windows and type in the address of a computer's location you want to access (e.g., `compute-0-12`).

Launching GUI applications using command line: `gedit`, `geany`

- `gedit` and `geany` are two commmonly used GUI-based text editor in Linux.
- `gedit` for common text files editing, `geany` for editing programming source codes (Fortran, C++., etc.).
- By default, the files saved are stored in the home directory, irrespective of your current local host.

Launching GUI applications using command line: `mathematica, firefox`

- Mathematica can be launched by issuing "mathematica".
- Firefox can be launched by issuing "firefox".

Things you should know by now

By now. you should be able to:

- Remotely login to the comsics cluster from your local Windows
- Navigate to any nodes within comsics
- Creat files and directories in the frontend as well as in local directories in any compute-nodes.
- Transfer files from your home directories in comsics to any nodes's local directories (e.g., `/state/partition1`)
- Transfer files from your local Windows to your home directories in comsics

Exercise

- So, try transfer a file (e.g., `testmove.txt`) from your local Windows machine to your home directory in comsics.
- Then copy `testmove.txt` from your home directory in comsics to the local hardisk at a chosen compute node. Rename the file to `testmove2.txt`. Then logout from comsics.
- Login to comsics and try to tranfer the `testmove2.txt` file from the local hardisk of the compute node to your local Windows machine.

user and home directory

- E.g., the user `tlyoon` is the owner of his home directory `/export/home/tlyoon`. He can read, write and execute the content in his home directory.
- E.g., the user `tlyoon` is not the owner of the home directory of the other user `human1`, whose home directory is `/export/home/human1`.
- A user can check out his home by issuing `echo $ HOME`.
- `$HOME` is an ENVIRONMENTAL VARIABLE in linux that specify the name of an user's home directory. `echo` is a Linux command that echo to the screen what you type after echo.

user and home directory (cont.)

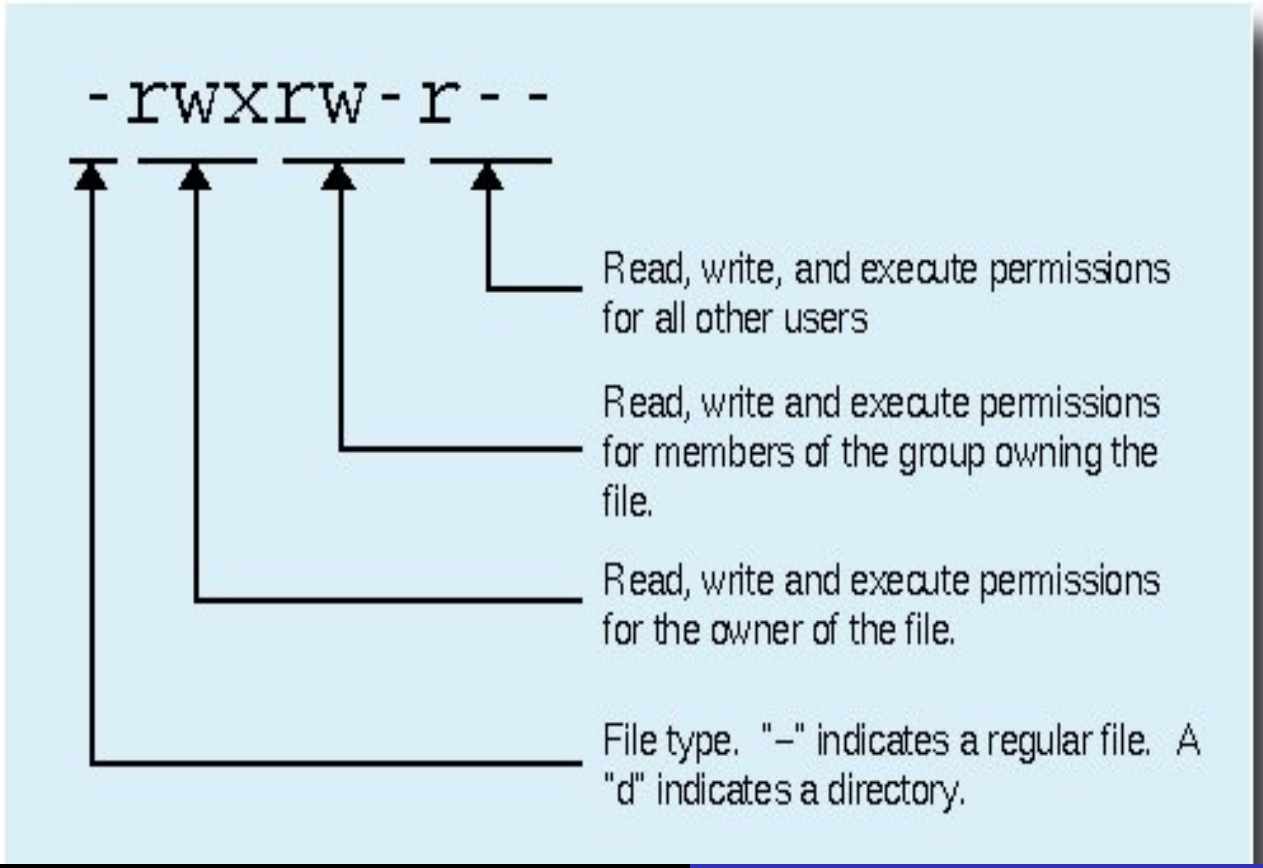
- `tlyoon` may read, and execute the content in the home directory of other users `/export/home/human1` if allowed by the permission set by the administrator.
- In the following example, `tlyoon` is not allowed to read the content of `/home/christopher`.

```
[tlyoon@compute-0-11 home]$ whoami
tlyoon
[tlyoon@compute-0-11 home]$ pwd
/home
[tlyoon@compute-0-11 home]$ ls
christopher  tlyoon
[tlyoon@compute-0-11 home]$ cd christopher/
-bash: cd: christopher/: Permission denied
[tlyoon@compute-0-11 home]$ ls -l
total 24
drwx------ 44 christopher christopher 20480 Aug  9 19:08 christopher
drwx------ 47 tlyoon      tlyoon       4096 Aug  9 21:00 tlyoon
```

Permission

Hierarchy of permission: root > owner > all other users.

```
[tlyoon@anicca ~]$ ls -lhr
total 16K
-rw-r--r-- 1 tlyoon users  445 Jul 31 10:55 orig
drwxr-xr-x 2 tlyoon users 4.0K Jul 29 11:55 Desktop
drwxr-xr-x 4 tlyoon users 4.0K Jul 29 11:55 bio
-rwxr-xr-x 1 tlyoon users  197 Jul 31 11:10 addmathlm
```



root directories vs. home directories

- Root directories broadly speaking are directories which are not a user's home directory.
- All the directories shown below are root directories, except `/home/tlyoon` and `/home/christopher`.
- `/home/tlyoon` and `/home/christopher` are the home directories belong to users `tlyoon` and `christopher`.

```
[tlyoon@compute-0-11 /]$ ls
bin  boot  etc     home     lib    lost+found  misc  opt   root  selinux  srv    sys  usr
bio  dev   export  install  lib64  media       mnt   proc  sbin  share    state  tmp  var
[tlyoon@compute-0-11 /]$ cd home
[tlyoon@compute-0-11 home]$ ls
christopher  tlyoon
```

root vs. ordinary users

- Only Adminstrator (a.k.a, the user `root` a.k.a `su` (superuser)) can access root directories.
- By default ordinary users are allowed limited access to some root directories (read and execute, but never write).
- Essentially, ordinary users can never mess with the root directories of Linux. They can only mess with their home directories without destabalising the Linux system.
- Usually installation of a system-wide software (which means a software that can be executed and read by all users) has to be done by `su`, as this involves writting to the root directories.

More linux command - chmod

- Change the mode of a file using chmod, e.g.,

```
[root@anicca ~]# touch testing
[root@anicca ~]# ls -l testing
-rw-r--r-- 1 root root 0 Aug  8 15:17 testing
[root@anicca ~]# chmod +x testing
[root@anicca ~]# ls -l testing
-rwxr-xr-x 1 root root 0 Aug  8 15:17 testing
```

The most commonly used permission is `chmod 755 filename` - can be read and executed but not write by group members.

More linux command - `file`

- `man file` to see the manual for this command

More linux command - `touch`

- `touch newfile` creat a new file called `newfile`
- `touch oldfile` change the last modied time of `oldfile` to now.

vi

- `vi` is the default Linux editor.
- `vi newfile` will open a new file named `newfile`.
- Vi has two modes: command mode vs. insert mode.
- By default, when you first open a vi session, you are in the commond mode. You can't enter any text in a commond mode.
- To write text into the file, press `i` to go into insert mode.
- Use the `HOME`, `END`, `DEL`, `BACKSPACE`, $\uparrow$, $\downarrow$, $\rightarrow$, $\leftarrow$ keys to move the cursor around.
- After finish editing the text, press `ESC` key to return to the commond mode.
- You may delete a line of text in the commond mode by pressing the key `d` twice.
- To save the edited file, press `ESC` followed by :w.
- To save and quit the edited file forcefully, press `ESC` followed by :wq!.
- To quit without saving the edited file forcefully, press `ESC`

An executable shell script

- Use vi editor to creat you first executable shell script.

```
#  /bin/bash
echo "hello world"
```

- List the details of the file hello.sh to check that the file is not executatable.

```
ls -l hello.sh
```

- Change the permission of the file to executable

```
chmod +x hello.sh
```

- Execute the .sh file by issuing `./`

```
./hello.sh
```

```
[tlyoon@compute-0-11 temp]$ ls -l
total 4
-rw-r--r-- 1 tlyoon users 34 Aug  9 22:35 hello.sh
[tlyoon@compute-0-11 temp]$ chmod +x hello.sh
[tlyoon@compute-0-11 temp]$ ls -l
total 4
-rwxr-xr-x 1 tlyoon users 34 Aug  9 22:35 hello.sh
[tlyoon@compute-0-11 temp]$ ./hello.sh
hello world
```

wget

- A particularly useful tool in Linux is `wget` where you use to download a file or the associated content of a url into your local directory.
- Try

```
wget http://www2.fizik.usm.my/tlyoon/presentation/cluster.pdf
```

- The following script in download.conf allows me to download automatically all the ICTP vedeo lectures into my hard disk.

download.conf

```
#! /bin/bash

wget http://www.ictp.tv/diploma/rooms/roomB/2009/04/16/2009.04.16_10.00-10.59.zip;
wget http://www.ictp.tv/diploma/rooms/roomB/2009/04/15/2009.04.15_09.00-09.59.zip;
wget http://www.ictp.tv/diploma/rooms/roomB/2009/04/15/2009.04.15_10.00-10.59.zip;
wget http://www.ictp.tv/diploma/rooms/roomB/2009/04/09/2009.04.09_10.00-10.59.zip;
wget http://www.ictp.tv/diploma/rooms/roomB/2009/04/09/2009.04.09_10.00-10.59.zip;
wget http://www.ictp.tv/diploma/rooms/roomB/2009/04/07/2009.04.07_10.00-10.59.zip;
wget http://www.ictp.tv/diploma/rooms/roomB/2009/04/07/2009.04.07_10.00-10.59.zip;
wget http://www.ictp.tv/diploma/rooms/roomB/2009/04/06/2009.04.06_10.00-10.59.zip;
wget http://www.ictp.tv/diploma/rooms/roomB/2009/04/06/2009.04.06_10.00-10.59.zip;
wget http://www.ictp.tv/diploma/rooms/roomB/2009/04/03/2009.04.03_10.00-10.59.zip;
wget http://www.ictp.tv/diploma/rooms/roomB/2009/04/03/2009.04.03_10.00-10.59.zip;
wget http://www.ictp.tv/diploma/rooms/roomB/2009/04/02/2009.04.02_10.00-10.59.zip;
wget http://www.ictp.tv/diploma/rooms/roomB/2009/04/02/2009.04.02_10.00-10.59.zip;
wget http://www.ictp.tv/diploma/rooms/roomB/2009/03/30/2009.03.30_10.00-10.59.zip;
wget http://www.ictp.tv/diploma/rooms/roomB/2009/03/30/2009.03.30_10.00-10.59.zip;
wget http://www.ictp.tv/diploma/rooms/roomB/2009/03/27/2009.03.27_10.00-10.59.zip;
wget http://www.ictp.tv/diploma/rooms/roomB/2009/03/27/2009.03.27_10.00-10.59.zip;
wget http://www.ictp.tv/diploma/rooms/roomB/2009/03/26/2009.03.26_10.00-10.59.zip;
wget http://www.ictp.tv/diploma/rooms/roomB/2009/03/26/2009.03.26_10.00-10.59.zip;
wget http://www.ictp.tv/diploma/rooms/roomB/2009/03/25/2009.03.25_10.00-10.59.zip;
wget http://www.ictp.tv/diploma/rooms/roomB/2009/03/25/2009.03.25_10.00-10.59.zip;
wget http://www.ictp.tv/diploma/rooms/roomB/2009/03/23/2009.03.23_10.00-10.59.zip;
wget http://www.ictp.tv/diploma/rooms/roomB/2009/03/23/2009.03.23_10.00-10.59.zip
```

wget

- `chmod +x download.conf; ./download.conf nohup`
- `nuhup` means the script will continue to run in the backgroud, even if you have logged out. You will find your downloaded files in current directory when you log in again later.

Running Fortran

- Create a file `hello1.for` that contains the following lines:

```
Program hello1
Print *,'Hello World 1'
End Program
```

- Compile the .for file by issuing the command

```
gfortran hello1.f90
```

```
[tlyoon@anicca temp]$ ls -l
total 4
-rw-r--r-- 1 tlyoon users 52 Aug  8 16:46 hello1.f90
[tlyoon@anicca temp]$ gfortran hello1.f90
[tlyoon@anicca temp]$ ls -l
total 12
-rwxr-xr-x 1 tlyoon users 7860 Aug  8 16:51 a.out
-rw-r--r-- 1 tlyoon users   52 Aug  8 16:46 hello1.f90
[tlyoon@anicca temp]$ ./a.out
 Hello World 1
```

Compilation fortran with a specified output name

- You specify the name of the output file rather than accepting the default one `a.out` using the switch -o

```
[tlyoon@anicca temp]$ ls -l
total 4
-rw-r--r-- 1 tlyoon users 50 Aug  8 17:06 hello1.f90
[tlyoon@anicca temp]$ gfortran hello1.f90 -o hello1
[tlyoon@anicca temp]$ ls -l
total 12
-rwxr-xr-x 1 tlyoon users 7860 Aug  8 17:07 hello1
```

- Here the source file hello1.for is compiled by gfotran into the executable file `hello1`.
- The `hello1` executable is then executed using `./hello1`

Compilation and executing a fortran code with shell script

- The procedure to run a fortran code:
  1. Prepare the source code in *.f90.
  2. Compile the source code from *.f90 into an executable.
  3. Execute the resultant executable.
- We can write a simple shell scipt to compile and run it in one short.

```
#!  /bin/bash
gfortran hello1.f90 -o hello1
./hello1
```

- The shell script `run.sh` must be in the same directory as hello1.f90
- `chmod +x run.sh`

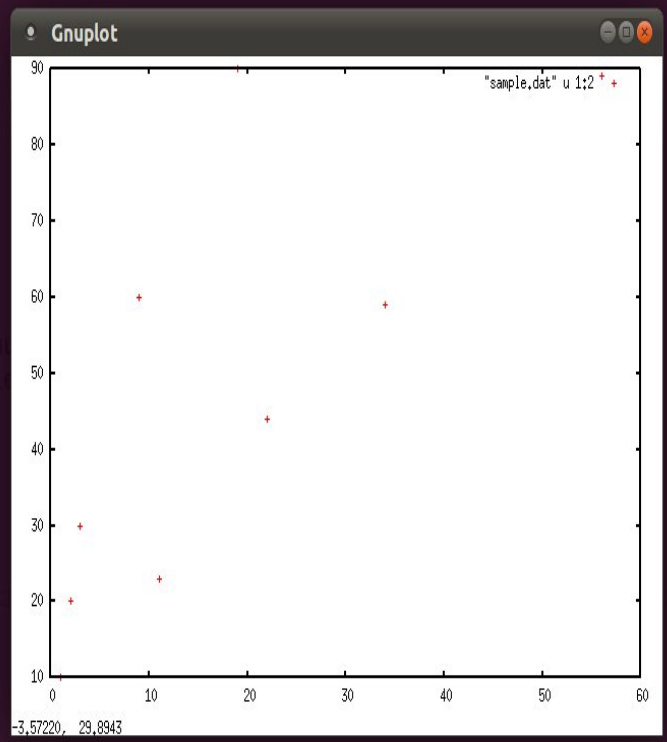
```
[tlyoon@anicca temp]$ ls -l
total 8
-rw-r--r-- 1 tlyoon users 50 Aug  8 17:06 hello1.f90
-rwxr-xr-x 1 tlyoon users 56 Aug  8 17:14 run.sh
[tlyoon@anicca temp]$ ./run.sh
 hello world 1
```

Gnuplot

- A command line plotting tool, handy for plotting data.
- Download a data file in your current directory, then use gnuplot to plot it.
- `wget http://www2.fizik.usm.my/tlyoon/info/sample.dat`
- `gnuplot`
- `plot "sample.dat" u 1:2 w points`

Gnuplot

```
[tlyoon@comsics tempwget]$ wget http://www2.fizik.usm.my/tlyoon/info/sample.dat
--2011-08-09 19:39:34--  http://www2.fizik.usm.my/tlyoon/info/sample.dat
Resolving www2.fizik.usm.my... 10.205.19.205
Connecting to www2.fizik.usm.my|10.205.19.205|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 53 [text/plain]
Saving to: `sample.dat'

100%[======================================================================================>] 53          --.-K/s   in 0s

2011-08-09 19:39:34 (10.1 MB/s) - `sample.dat' saved [53/53]

[tlyoon@comsics tempwget]$ ls
sample.dat
[tlyoon@comsics tempwget]$ gnuplot

        G N U P L O T
        Version 4.0 patchlevel 0
        last modified Thu Apr 15 14:44:22 CEST 2004
        System: Linux 2.6.18-164.6.1.el5

        Copyright (C) 1986 - 1993, 1998, 2004
        Thomas Williams, Colin Kelley and many others

        This is gnuplot version 4.0.  Please refer to the documentation
        for command syntax changes.  The old syntax will be accepted
        throughout the 4.0 series, but all save files use the new syntax.

        Type `help` to access the on-line reference manual.
        The gnuplot FAQ is available from
                http://www.gnuplot.info/faq/

        Send comments and requests for help to
                <gnuplot-info@lists.sourceforge.net>
        Send bugs, suggestions and mods to
                <gnuplot-bugs@lists.sourceforge.net>


Terminal type set to 'x11'
gnuplot> plot "sample.dat" u 1:2 w points
gnuplot>
```



Shell scripts for batching

- One can use shell script to batch a multiple steps programming procedure.
- Example: Compile a source code, execute it, output the resultant data to a pre-specified folder with a specified name, renaming the output files sequentially, then merging them into a single file, then plot the result file using gnuplot.
- More complicated procedure can be batched by manipulating shell scripts coupled with a programming language (fortran, Mathematica, C++, etc.).

My best reason to use Linux over Windows

- To batch a programming package code with a complicated structure.
- In Windows this can be achieved with very specific software package such as Microsoft visual studio. But it could also be machine-dependent. Windows lacks much flexibility Linux offers.
- Appears to be complicated to impliment, esp. to run in parallel model in a cluster.
- Programming in Linux is most convenient and natural. A code in one Linux machine can run easily in another Linux machine.

My best reason to use Linux over Windows

- Linux is shipped with all the necessary programming-related stuff for free (`gcc`, `cpp`, `python`, `lapack`, `blas`, `mpi`, `openmpi`, `gnuplot`, `numpy` and others numerical libraries).
- E.g., the Make function (used to link up a complicated fortran code) and shell script, are the most command tools used by programmers in scientific researach fiels. Linux is the primary choice by scientists.
- The equivalent of Make and shell scripting functions in Windows are hardly popular.

Cygwin

- Allows you to run Linux in Windows. See http://www.cygwin.com/.
- Slightly slower (cygwin is an "emulator"). Not a full Linux but enough for certain purposes e.g., shell scripting, Fortran, C++ programming, Latex, ssh, gnuplot etc.
- The default installation installs only very limited functionalies and is insufficient for most purpose.
- Instead, clicking on the "Default" field next to the "All" category will install every Cygwin package. This will download and install hundreds of megabytes to your computer.

Cygwin advantages

- Cygwin is recommended for people who:
- Don't want to have a dual boot in thier computer
- Don't want a fully-fledged Linux.
- Need to run Linux environment for not-too-complicated applications only occasionally.
- It is most convenient while running Windows but need to do some Linux-only functions such as ssh or shell sripting.