# Introduction to High Performance Computing (HPC)

sciCORE
Center for scientific computing

Swiss TPH
Swiss Tropical and Public Health Institute
Schweizerisches Tropen- und Public Health-Institu
Institut Tropical et de Santé Publique Suisse

SIB
Swiss Institute of Bioinformatics

Universität Basel

# Outline

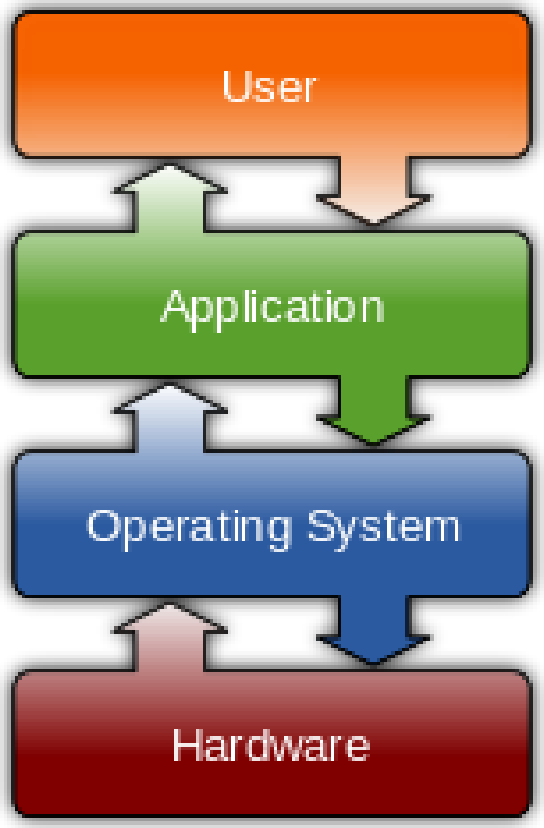

09:30 – Course starts
09:30 – 11:00: Introduction to Linux. Part 1
11:00 – 11:15: Coffee break
11:15 – 12:30: Introduction to Linux. Part 2

12:30 – 14:00: Lunch

14:30 – 16:00: Cluster usage. Part 1
16:00 – 16:15: Coffee break
16:15 – 17:30: Cluster usage. Part 2
17:30 – 18:00: Feedback + Q&A

I will assume no previous knowledge on Linux or on clusters usage.

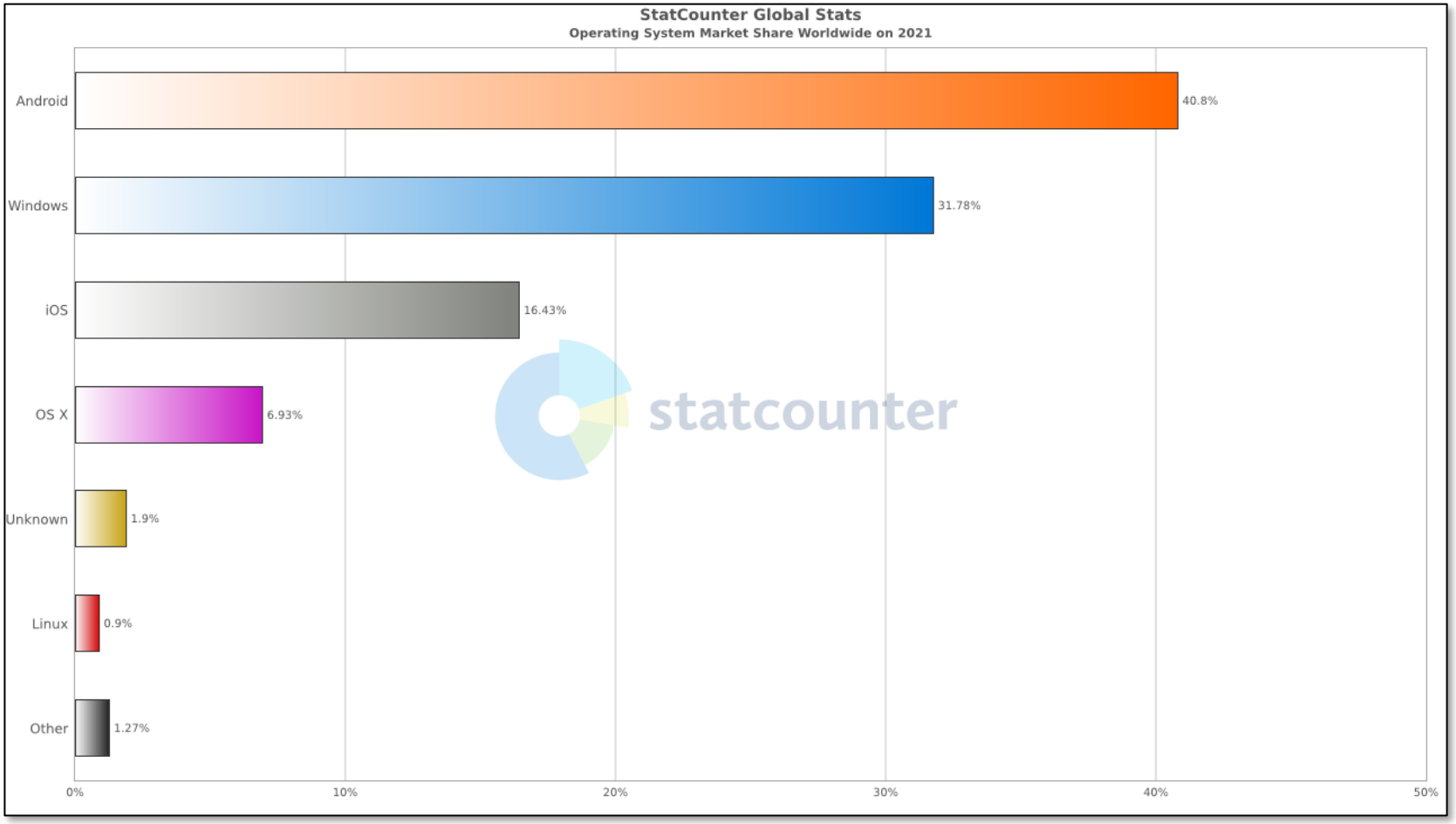
# What's an OS?



# What's an OS?

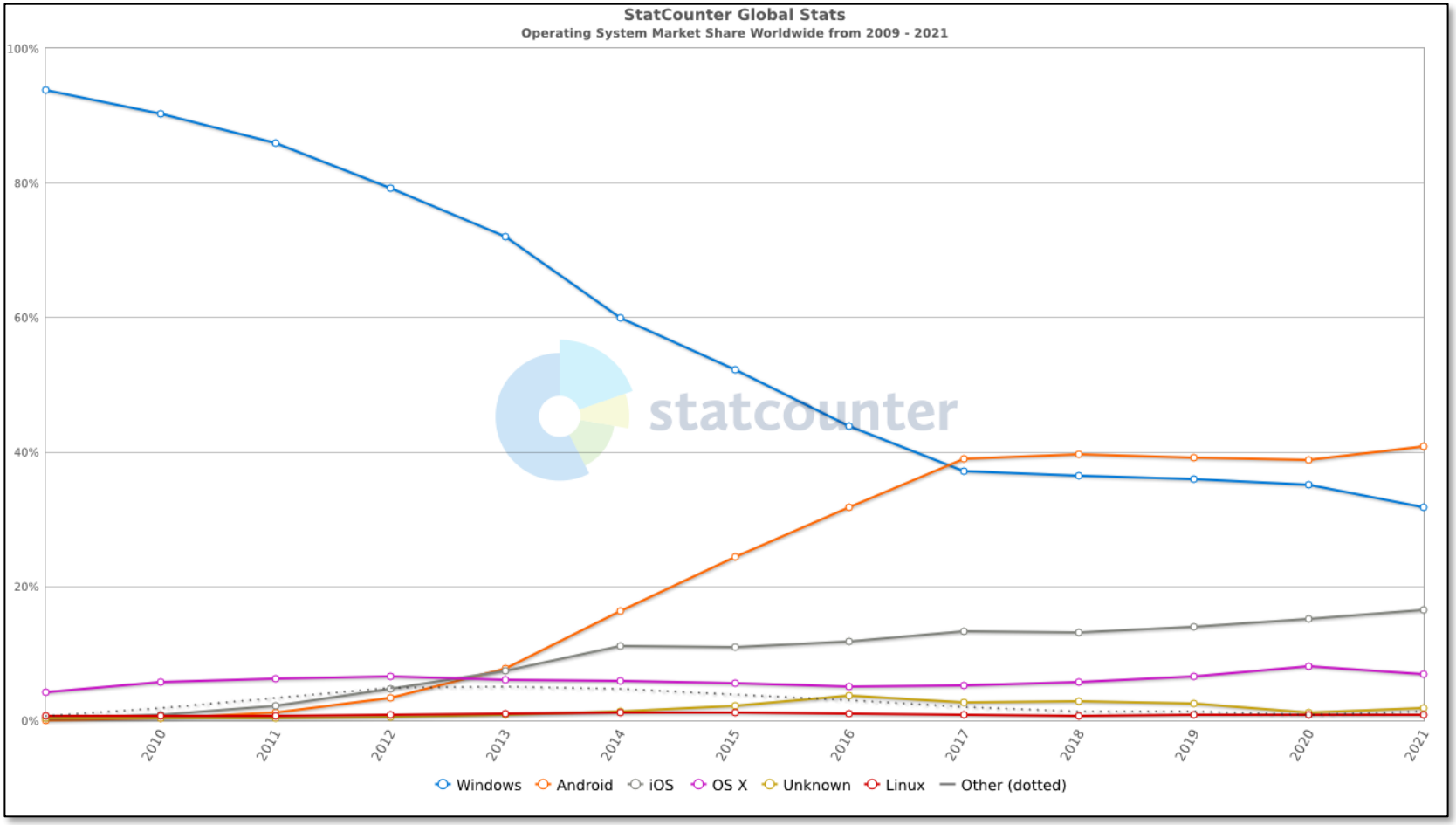

Sun
HP
IBM
VMWare
Apple
Oracle
OS X
Windows
Linux
Xen
Red Hat
Fedora
CentOS
Debian
Ubuntu
Mint
SUSE
Mageia
Arch Linux
Slackware
Mandriva
Gentoo
FreeBSD
OpenBSD
NetBSD
DragonFly BSD
Darwin

# What's an OS?



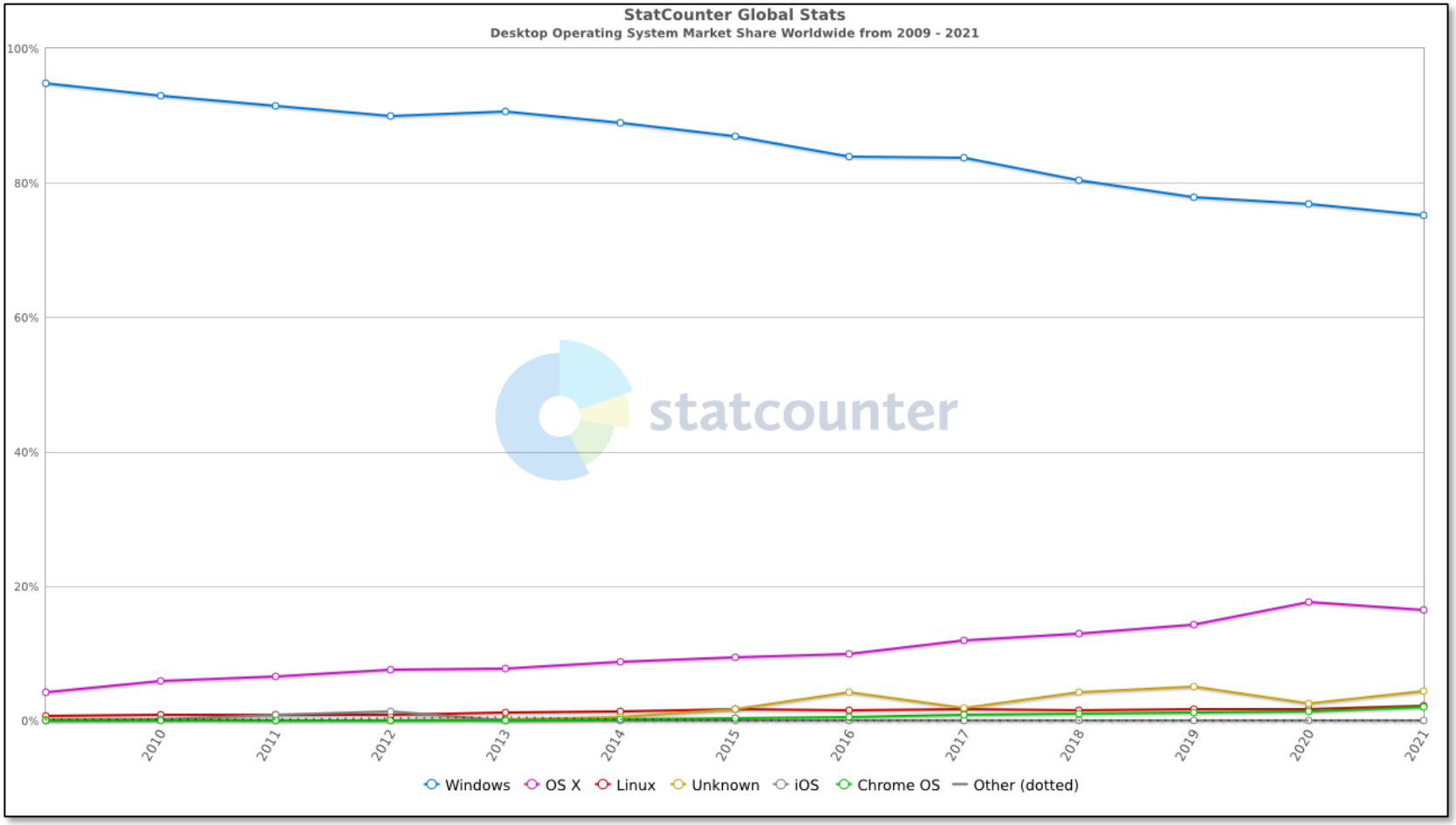

Source: statcounter.com

# What's an OS?



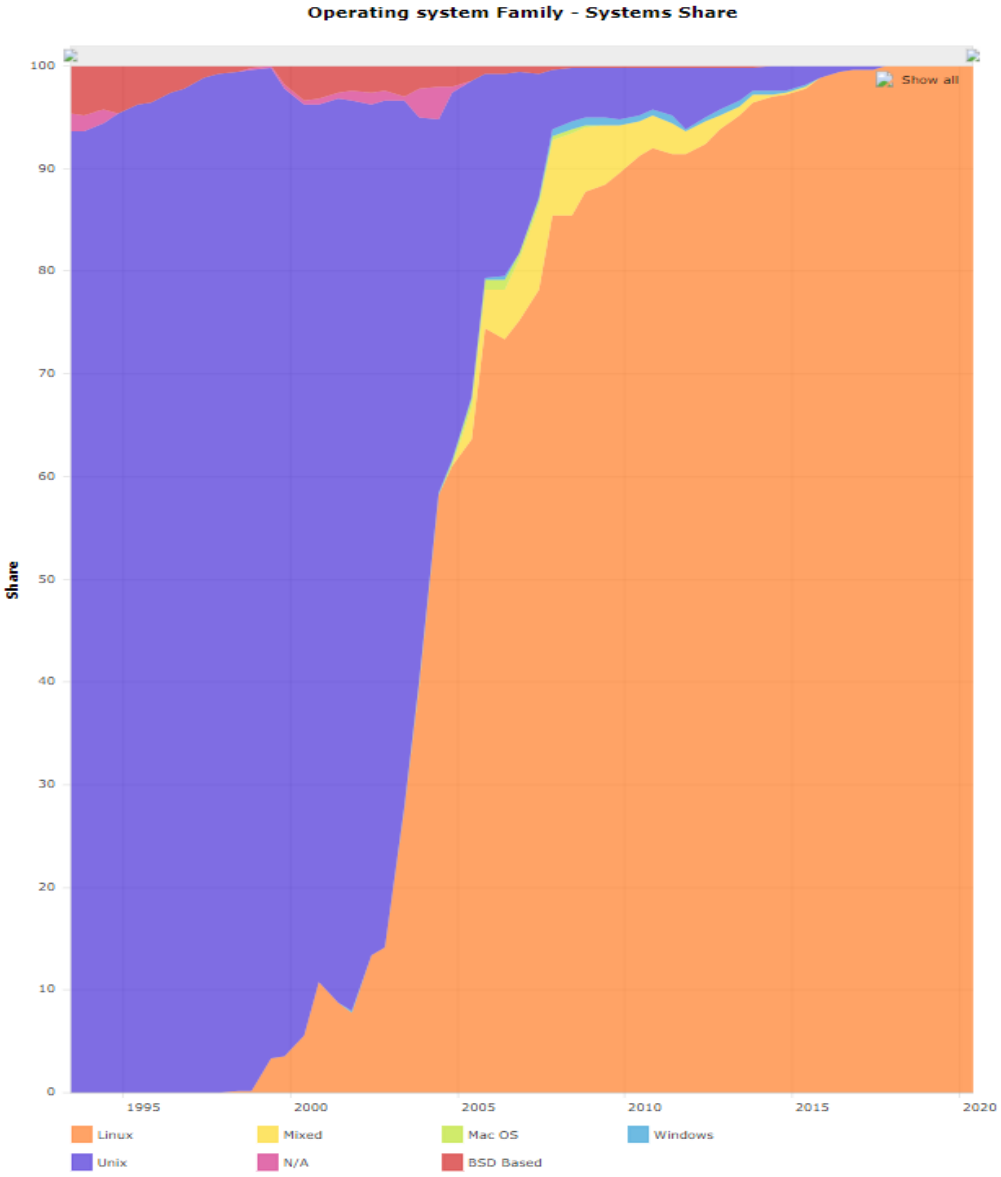

Source: statcounter.com

# What's an OS?




Source: statcounter.com

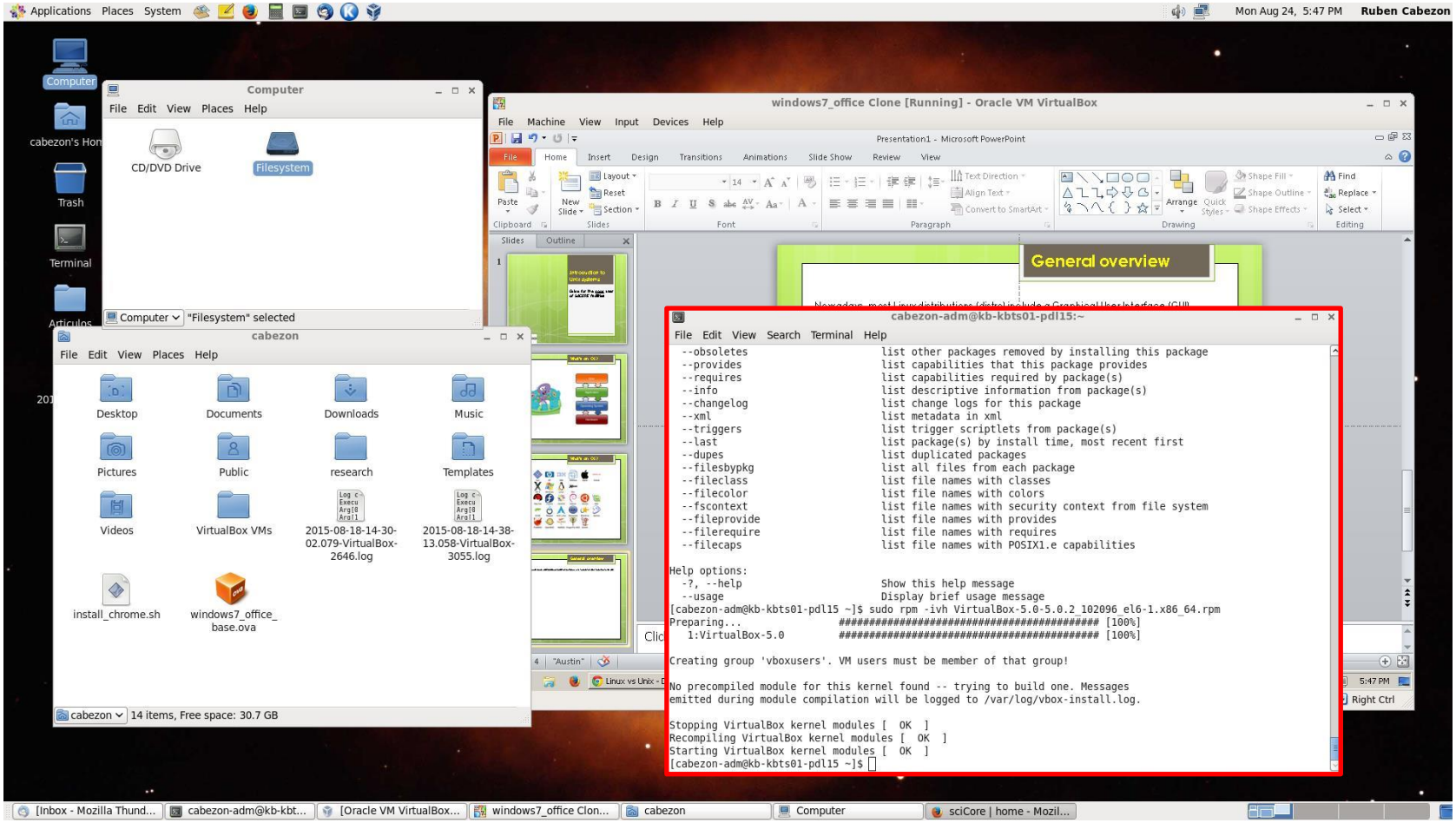
# What's an OS?




Source: top500.org

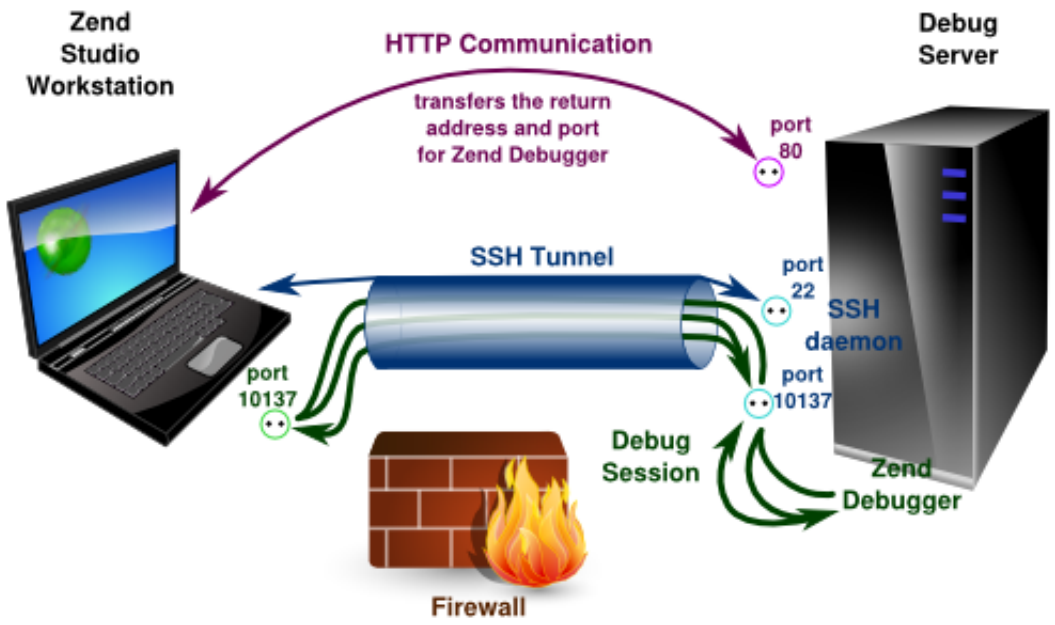
# General overview

Nowadays, most Linux distributions (distro) include a Graphical User Interface (GUI).
Usual ones: KDE, GNOME



# General overview

When using clusters or supercomputers we log in remotely, usually via a secure connection: ssh



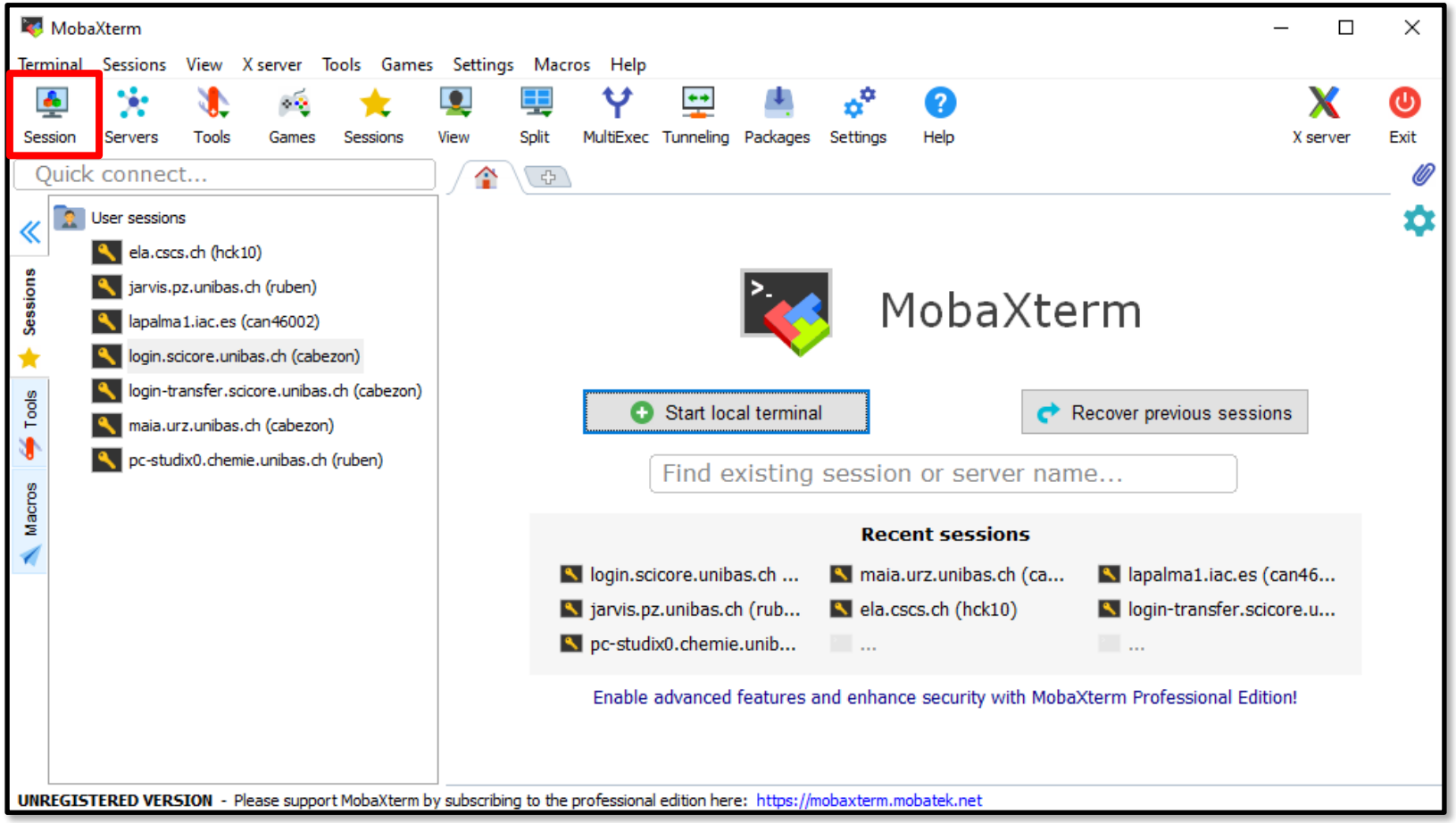

This will open a terminal, in which we will log in and work.

**NOTE:** Windows users without ssh, download MobaXterm
http://mobaxterm.mobatek.net/download-home-edition.html

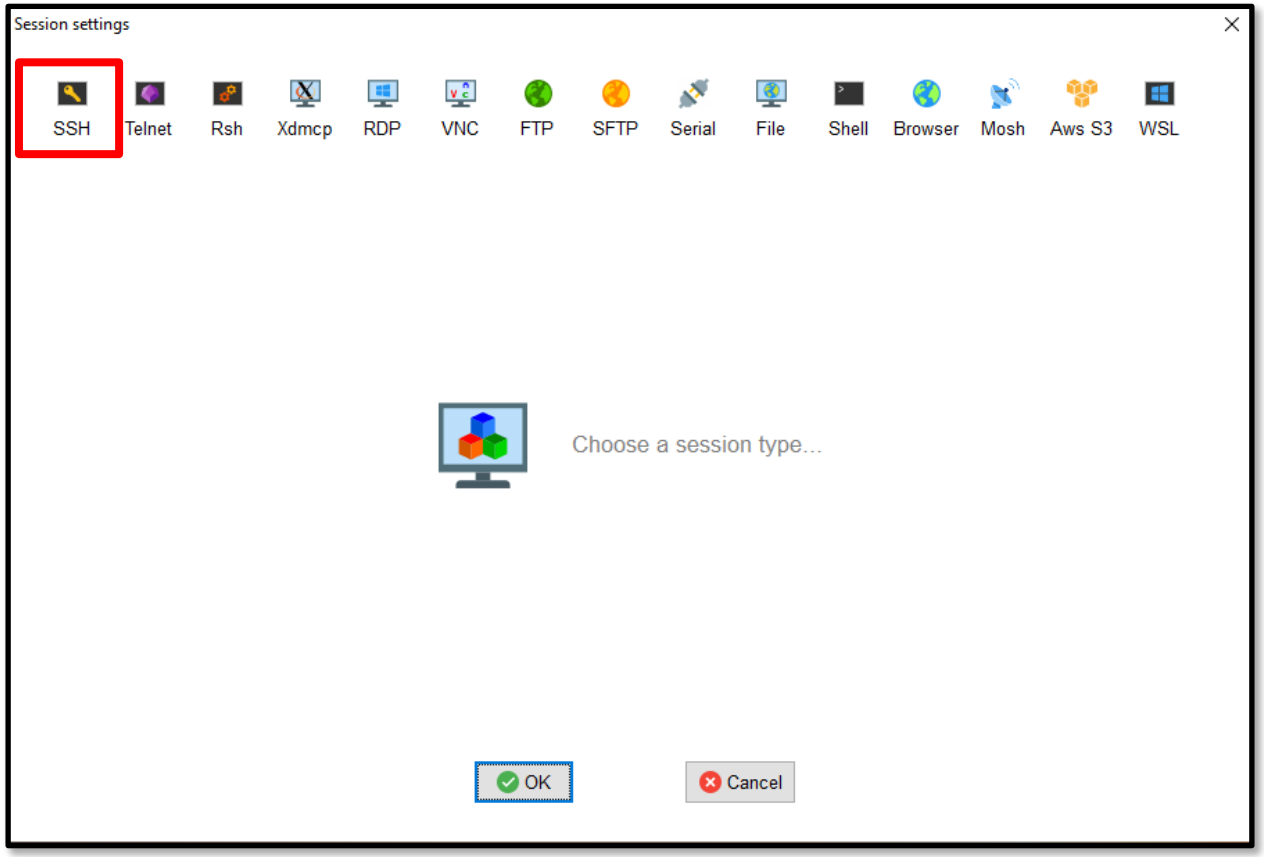
Another option is PuTTY client + Xming
http://the.earth.li/~sgtatham/putty/latest/x86/putty.exe
http://sourceforge.net/projects/xming/

**NOTE:** Mac users without X11, download XQuartz
http://xquartz.macosforge.org

# Using MobaXterm



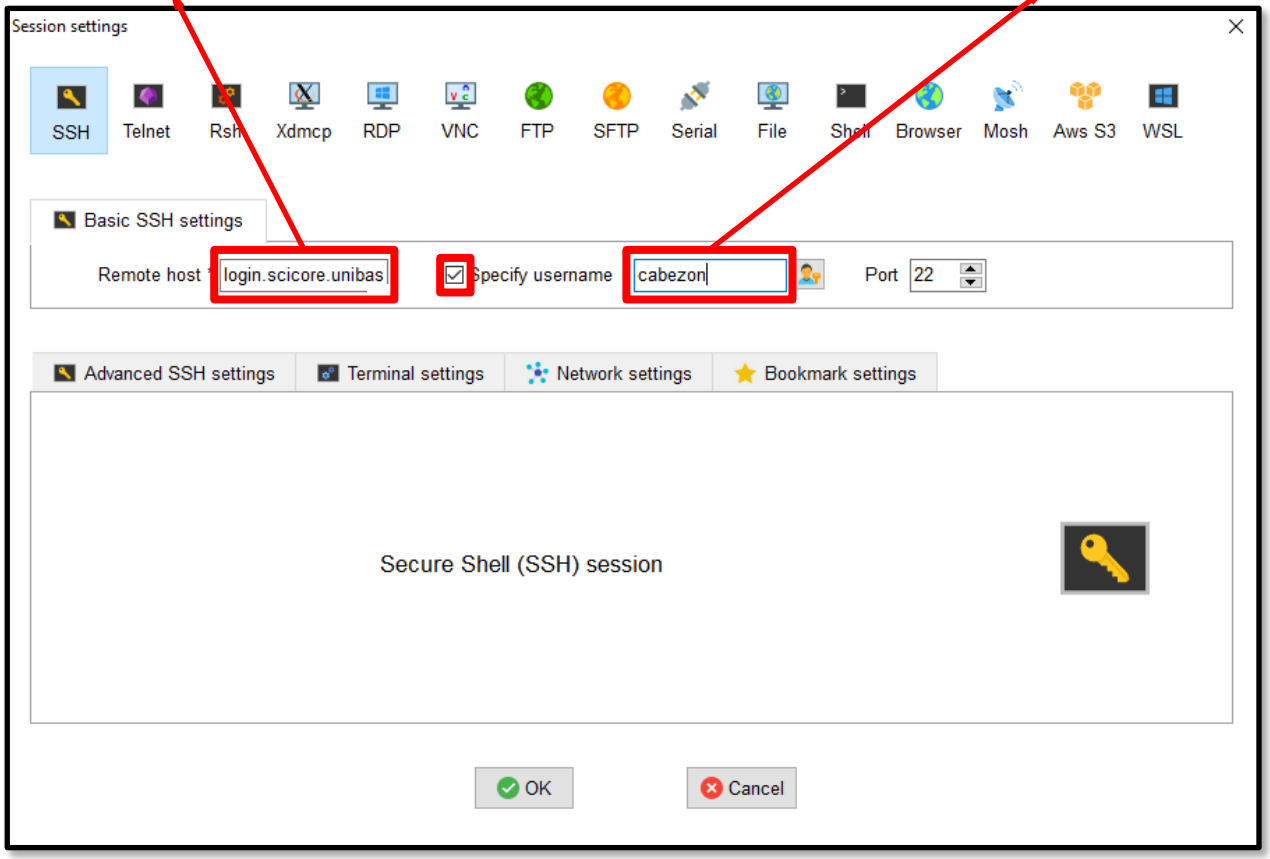
# Using MobaXterm



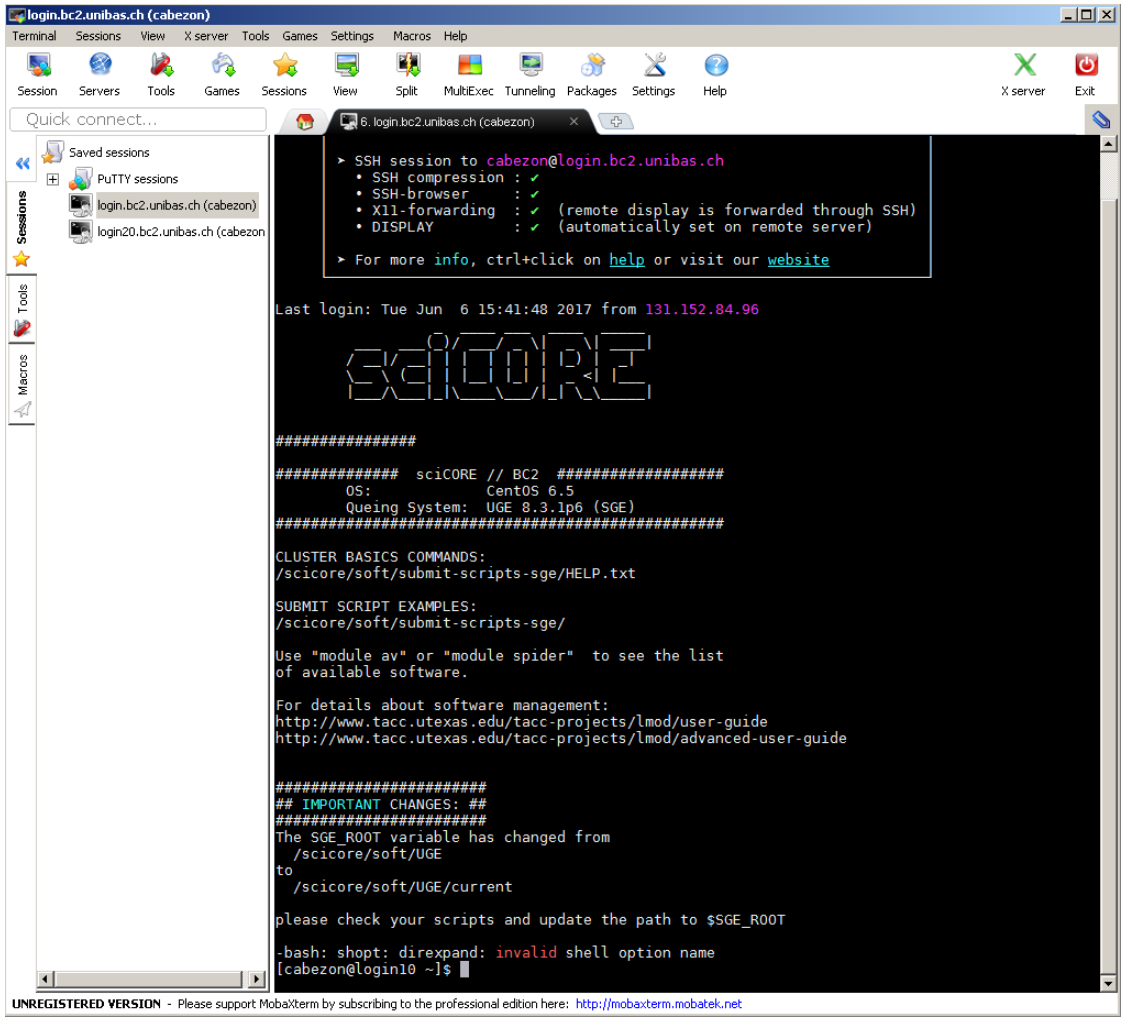
# Using MobaXterm

86.119.35.191

<your username>

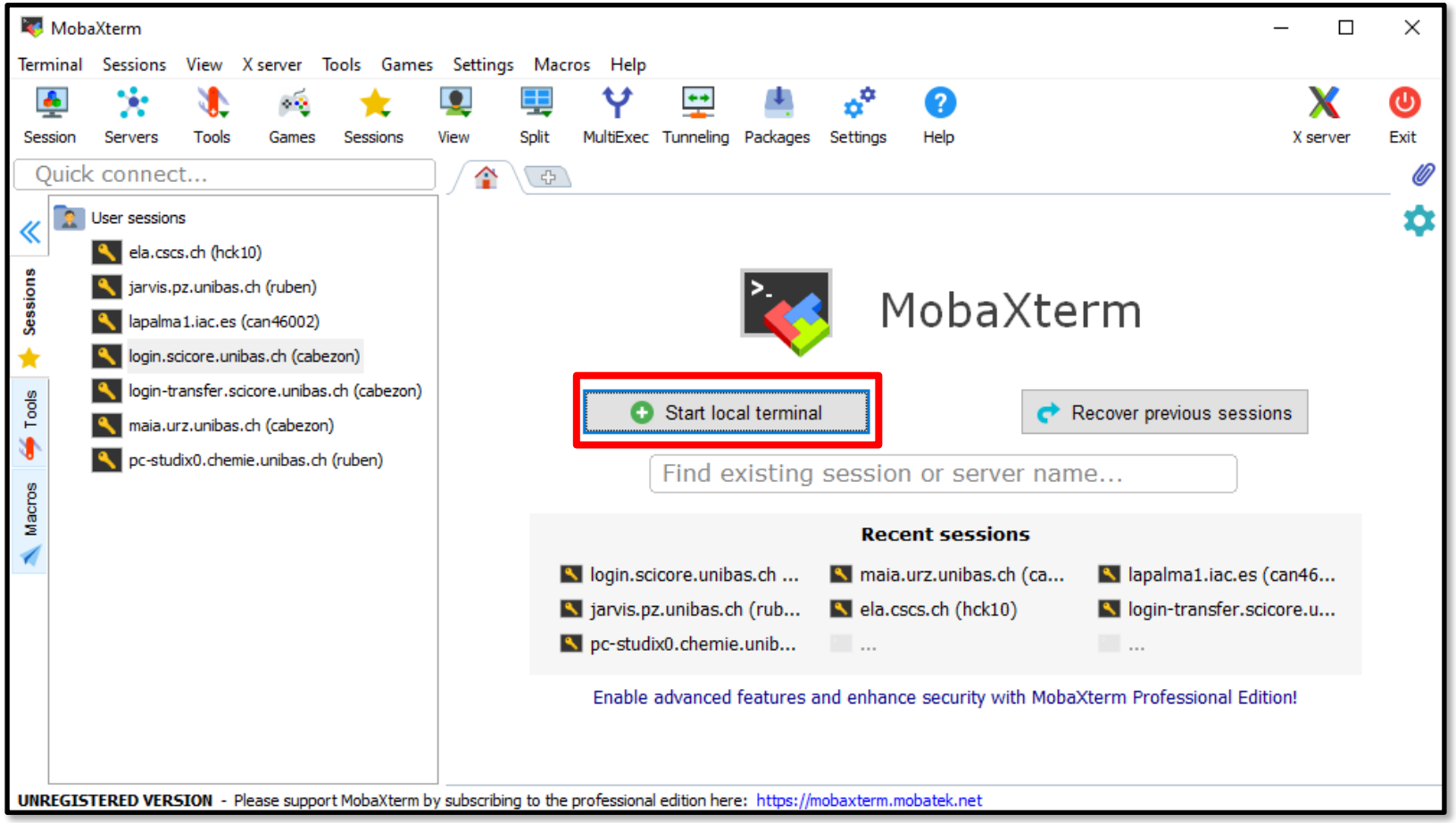


# Using MobaXterm

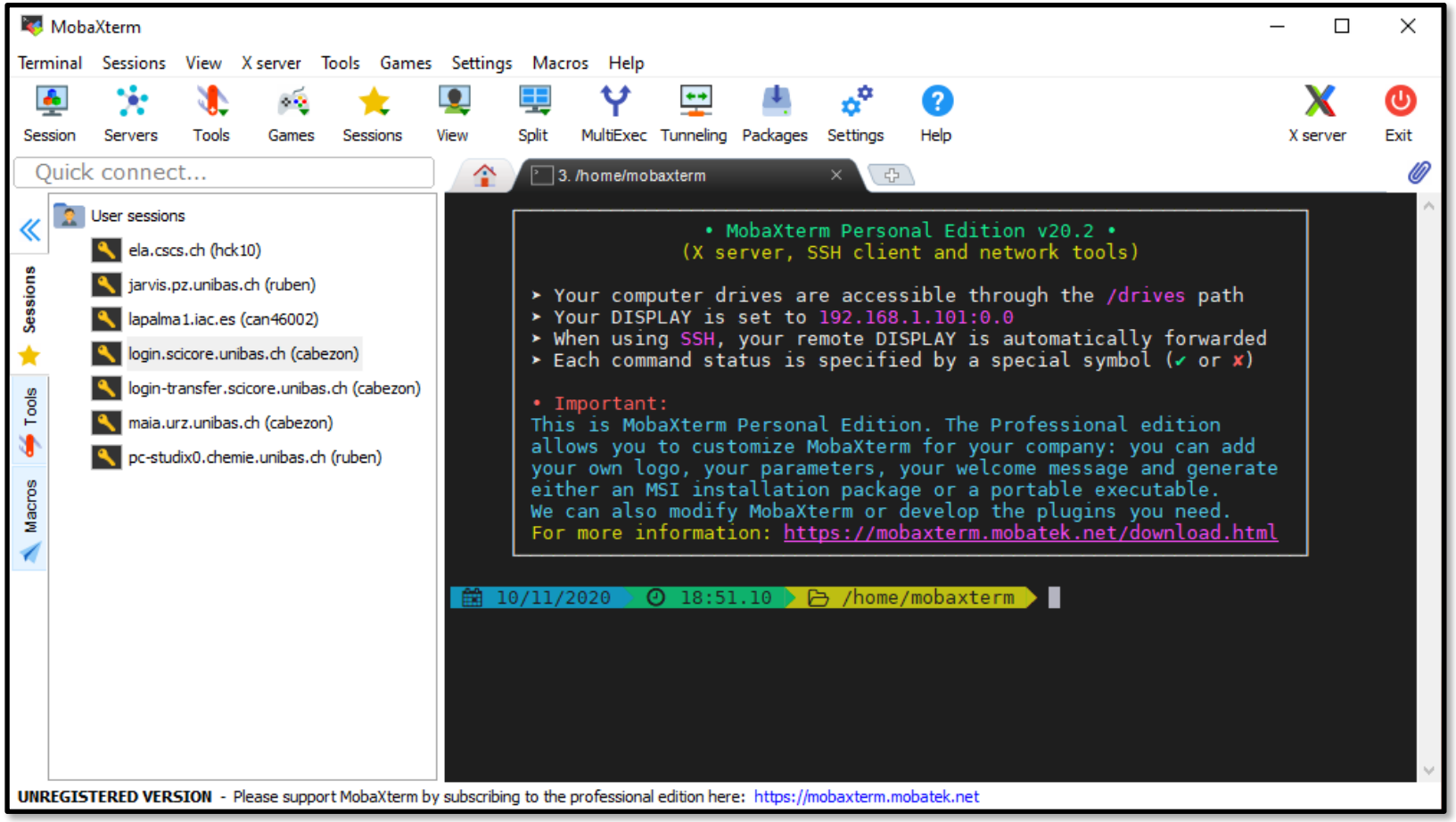



ssh [options] <username>@86.119.35.191

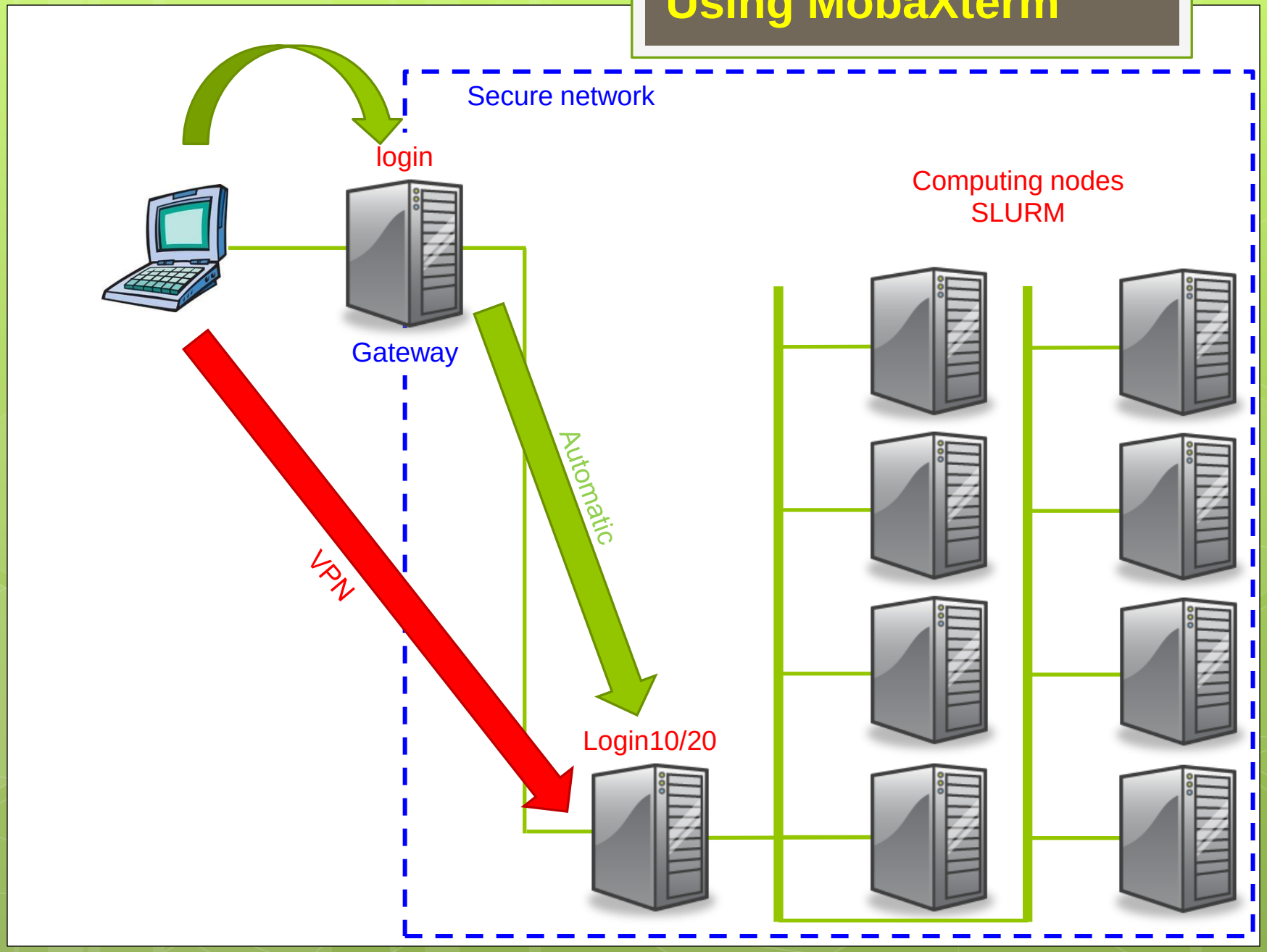
# Using MobaXterm



# Using MobaXterm



# Using MobaXterm



# Nomenclature

Something in red means that it has to be written literally in the command line.
Something in blue is not a literal command but information for you.

Something between < > has to be completed correspondingly.

Ex: squeue –u <username> ➡ squeue –u ruben

Ex: Find information about the command squeue

**ATTENTION**: Capital letters, blank spaces, dots (**.**), hyphens (–), and slash(**/**)

**Linux is case-sensitive!**

# 0. Exercise

1. Log in to the cluster (ssh –Y <username>@86.119.35.191)
   (If you have windows, simply use MobaXterm to connect)
2. Check that the computer knows who you are with who am i
3. Clear the screen with clear
4. Write yes. Stop the command with CTRL+C
5. Write sleep 5 and hit Enter. What happens?
6. Write time sleep 5 and hit Enter. What information does time provide?
7. Close the connection with CTRL+D.

Do you already know all this?

1. Do the exercises of the handout.
2. Write a script that prints in the screen the total size (in Mb) of the contents of a folder passed as an argument, without using du
   Hint: You might need to use ls, awk, echo and bc. Google a bit to learn about these if you don't know them already.

## Summary

| Command | Used to… | Common options |
|---|---|---|
| ssh | establish a secure connection<br>**ssh [options] <username>@hostname** | -X (enables trusted X11 forwarding)<br>-Y (enables untrusted X11 forwarding) |
| who am I | show user connection info | |
| clear | clear the screen | |
| logout | log out of the session | |
| yes | continuously output ‘y’ | |
| sleep | puts the CPU to sleep<br>**sleep <number od seconds>** | |
| time | provides the cpu, user and wallclock time of any command | /usr/bin/time –v <command> |
| CTRL + C | interrupts a running command | |
| CTRL + D | shortcut to logout | |

# 1. Exercise

1. Log in to the cluster
2. List the contents of the temporary directory: ls /tmp
3. Learn about ls by looking at its manual: man ls (press q to exit). Try also info ls (press q to exit).
4. Enter again in the manual of ls and search for the option -1 by writing: /-1 You can search the next one forwards pressing n and backwards with SHIFT+n
5. Try more options of ls (-l, -lt, -lS, -m, -1, -lh, etc…)
6. Create a directory named test: mkdir test
7. Check that the directory exists: ls
8. Enter in test/: cd test
9. Exit test/: cd .. (attention to the blank space after 'cd' !)
10. Create a file containing the list of files of the tmp directory: ls /tmp > data.d
11. Check that the file data.d exists
12. Check the contents of data.d: less data.d (press q to exit)
13. Move data.d into test/: mv data.d test (do you need / ?)
14. Enter in test/ and check that data.d is in there.
15. Copy data.d into a file named data2.d: cp data.d data2.d
16. Delete data.d: rm data.d (careful!!)
17. With the commands that you know figure out a way of renaming data2.d to finaldata.d
18. Create a directory named Exercise1 and move finaldata.d inside
19. Go up one level and rename test/ to linux_course/

The final outcome should be a directory structure linux_course/Exercise1 that includes a file named finaldata.d with the list of files from tmp/

## Summary

| Command | Used to… | Common options |
|---|---|---|
| ls | list files in a directory | -l (long listing),<br>-h (human readable),<br>-S (sort by size),<br>-t (sort by modification time)<br>-1 (show 1 file / line) |
| cd | change directory<br>**cd <path>** | |
| . | point to the present directory | |
| .. | point to the parent directory | |
| cp | copy files<br>**cp [options] <source> <destination>** | -r (recursive), -v (verbose) |
| mv | move files (works with directories too)<br>**mv <source> <destination>** | |
| rm | remove files | -rf (recursive+force) **(!)**,<br>-i (ask before) |
| less | show contents of a file | |
| man / info | show manual about a command | |
| > | transfer the output from screen to a file | |

# Wildcard characters

Wikipedia:

“ A wildcard character is a single character used to represent a number of characters.”

# Wildcard characters

Wikipedia:

" A wildcard character is a single character used to represent a number of characters."

* Matches zero or more characters
? Matches one single character
[ ] Matches a range or a selection

# Wildcard characters

Wikipedia:

" A wildcard character is a single character used to represent a number of characters."

- ***** Matches zero or more characters
- **?** Matches one single character
- [ ] Matches a range or a selection

```
[cabezon@maia run]$ ls
1                                   dumpfile005000                hostfilerun~
a123                                dumpfile005500                init_scenario.f90
b111                                dumpfile006000                init_scenario.f90~
c134                                dumpfile006500                j2Msniaflame
conservelawsRel2MpM1_08HeRot.d      dumpfile007000                nohup.out
conservelawsRel2MpM1_08HeRot.d~     dumpfile007500                parameters.f90
createfileplot7.f90                 dumpfile008000                parameters.f90~
d144                                dumpfile008500                parameters_idsa.f90
data                                dumpfile009000                readdata.f90
dumpfile000500                      dumpfile009500                readdata.f90~
dumpfile001000                      dumpfile010000                REPORT
dumpfile001500                      dumpfile010500                save
dumpfile002000                      e165                          temp
dumpfile002500                      escrituramod.f90              tiempos.d
dumpfile003000                      estabilRel2MpM1_08HeRot.d     timectrl.d
dumpfile003500                      estabilRel2MpM1_08HeRot.d~    timing.d
dumpfile004000                      find56Ni.f90
dumpfile004500                      hostfilerun
[cabezon@maia run]$
```

- Copy all dumpfiles into save/:
- Copy all .d files into save/:
- Copy all dumpfiles greater than 9500 into save/:
- Remove all dumpfiles between 5000 and 9500 (both included):
- Remove the five files a123, b111, c134, d144, e165:

# Wildcard characters

Wikipedia:

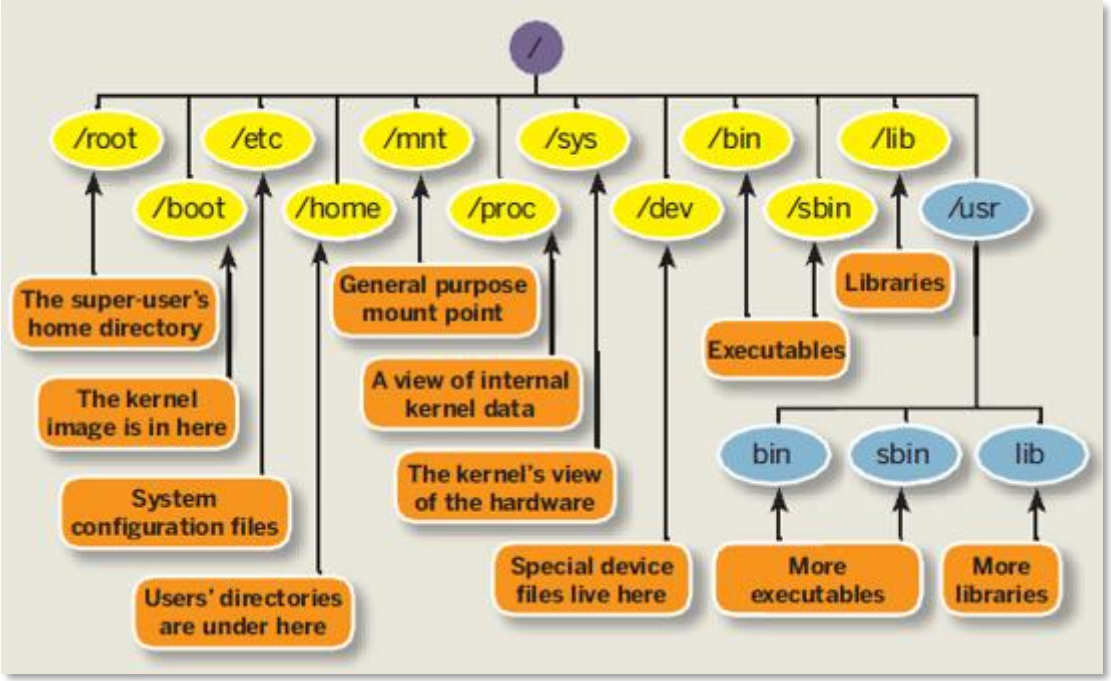
" A wildcard character is a single character used to represent a number of characters."

- ***** Matches zero or more characters
- **?** Matches one single character
- [ ] Matches a range or a selection

```
[cabezon@maia run]$ ls
1                                  dumpfile005000                hostfilerun~
a123                               dumpfile005500                init_scenario.f90
b111                               dumpfile006000                init_scenario.f90~
c134                               dumpfile006500                j2Msniaflame
conservelawsRel2MpM1_08HeRot.d     dumpfile007000                nohup.out
conservelawsRel2MpM1_08HeRot.d~    dumpfile007500                parameters.f90
createfileplot7.f90                dumpfile008000                parameters.f90~
d144                               dumpfile008500                parameters_idsa.f90
data                               dumpfile009000                readdata.f90
dumpfile000500                     dumpfile009500                readdata.f90~
dumpfile001000                     dumpfile010000                REPORT
dumpfile001500                     dumpfile010500                save
dumpfile002000                     e165                          temp
dumpfile002500                     escrituramod.f90              tiempos.d
dumpfile003000                     estabilRel2MpM1_08HeRot.d     timectrl.d
dumpfile003500                     estabilRel2MpM1_08HeRot.d~    timing.d
dumpfile004000                     find56Ni.f90
dumpfile004500                     hostfilerun
[cabezon@maia run]$
```

- Copy all dumpfiles into save/: REDACTED
- Copy all .d files into save/: REDACTED
- Copy all dumpfiles greater than 9500 into save/: REDACTED
- Remove all dumpfiles between 5000 and 9500 (both included): REDACTED
- Remove the five files a123, b111, c134, d144, e165: REDACTED

# Linux filesystem




/home/<username> is the place where your stuff is stored. Usual place to work in the local computer.

**In the test cluster is /shared/home/<username>**

# More about files

```
cabezon@login11:~
-rw-r--r--  1 cabezon physik      121 Jun 22  2012 .kshrc
-rw-------  1 cabezon physik      234 Aug 19 16:26 .lesshst
drwxrwxr-x  3 cabezon physik       19 Jan 16  2015 .lmod.d/
drwx------  3 cabezon physik       18 Dec 11  2013 .local/
drwxr-xr-x  7 cabezon physik       74 Jan 22  2015 martin/
drwxr-xr-x  3 cabezon physik       19 Jan 24  2014 .matlab/
drwxr-xr-x  2 cabezon physik       27 May  2  2013 .matplotlib/
drwx------  2 cabezon physik       58 Jan 19  2015 .mc/
drwxr-xr-x  5 cabezon physik       51 May  2  2013 .mozilla/
-rw-r--r--  1 cabezon physik     2458 Sep 12  2014 new_absor_effect.f90
-rw-r--r--  1 cabezon physik    18240 May  8 11:55 nuprox3D.f90
drwxr-----  3 cabezon physik       18 May  2  2013 .pki/
-rw-r--r--  1 cabezon physik     6525 Jun 23 12:05 progenitor.f
-rw-------  1 cabezon physik      256 May  2  2013 .pulse-cookie
-rw-r--r--  1 cabezon physik     1092 Jan 22  2015 REPORT
drwxr-xr-x 21 cabezon physik     4096 Aug 25 09:33 research/
lrwxrwxrwx  1 root    root         13 Nov 16  2012 save -> /save/cabezon
drwxr-xr-x  3 cabezon physik       19 Jan 12  2015 software/
drwx------  2 cabezon physik       95 Jan 16  2015 .ssh/
drwxr-xr-x  6 cabezon physik     4096 Jan 22  2015 tmp/
-rw-------  1 cabezon physik     8260 Aug 20 17:18 .viminfo
-rw-------  1 cabezon physik    13390 Aug 26 14:53 .Xauthority
-rw-r--r--  1 cabezon physik      658 Jun 23  2012 .zshrc
[cabezon@login11 ~]$
```

Type: d (directory), - (ordinary file), l (symbolic link)

# More about files

```
cabezon@login11:~
-rw-r--r--  1 cabezon physik      121 Jun 22  2012 .kshrc
-rw-------  1 cabezon physik      234 Aug 19 16:26 .lesshst
drwxrwxr-x  3 cabezon physik       19 Jan 16  2015 .lmod.d/
drwx------  3 cabezon physik       18 Dec 11  2013 .local/
drwxr-xr-x  7 cabezon physik       74 Jan 22  2015 martin/
drwxr-xr-x  3 cabezon physik       19 Jan 24  2014 .matlab/
drwxr-xr-x  2 cabezon physik       27 May  2  2013 .matplotlib/
drwx------  2 cabezon physik       58 Jan 19  2015 .mc/
drwxr-xr-x  5 cabezon physik       51 May  2  2013 .mozilla/
-rw-r--r--  1 cabezon physik     2458 Sep 12  2014 new_absor_effect.f90
-rw-r--r--  1 cabezon physik    18240 May  8 11:55 nuprox3D.f90
drwxr-----  3 cabezon physik       18 May  2  2013 .pki/
-rw-r--r--  1 cabezon physik     6525 Jun 23 12:05 progenitor.f
-rw-------  1 cabezon physik      256 May  2  2013 .pulse-cookie
-rw-r--r--  1 cabezon physik     1092 Jan 22  2015 REPORT
drwxr-xr-x 21 cabezon physik     4096 Aug 25 09:33 research/
lrwxrwxrwx  1 root    root        13 Nov 16  2012 save -> /save/cabezon
drwxr-xr-x  3 cabezon physik       19 Jan 12  2015 software/
drwx------  2 cabezon physik       95 Jan 16  2015 .ssh/
drwxr-xr-x  6 cabezon physik     4096 Jan 22  2015 tmp/
-rw-------  1 cabezon physik     8260 Aug 20 17:18 .viminfo
-rw-------  1 cabezon physik    13390 Aug 26 14:53 .Xauthority
-rw-r--r--  1 cabezon physik      658 Jun 23  2012 .zshrc
[cabezon@login11 ~]$
```

Permissions: define the access rights.

# More about files

```
cabezon@login11:~
-rw-r--r--  1 cabezon physik      121 Jun 22  2012 .kshrc
-rw-------  1 cabezon physik      234 Aug 19 16:26 .lesshst
drwxrwxr-x  3 cabezon physik       19 Jan 16  2015 .lmod.d/
drwx------  3 cabezon physik       18 Dec 11  2013 .local/
drwxr-xr-x  7 cabezon physik       74 Jan 22  2015 martin/
drwxr-xr-x  3 cabezon physik       19 Jan 24  2014 .matlab/
drwxr-xr-x  2 cabezon physik       27 May  2  2013 .matplotlib/
drwx------  2 cabezon physik       58 Jan 19  2015 .mc/
drwxr-xr-x  5 cabezon physik       51 May  2  2013 .mozilla/
-rw-r--r--  1 cabezon physik     2458 Sep 12  2014 new_absor_effect.f90
-rw-r--r--  1 cabezon physik    18240 May  8 11:55 nuprox3D.f90
drwxr-----  3 cabezon physik       18 May  2  2013 .pki/
-rw-r--r--  1 cabezon physik     6525 Jun 23 12:05 progenitor.f
-rw-------  1 cabezon physik      256 May  2  2013 .pulse-cookie
-rw-r--r--  1 cabezon physik     1092 Jan 22  2015 REPORT
drwxr-xr-x 21 cabezon physik     4096 Aug 25 09:33 research/
lrwxrwxrwx  1 root    root        13 Nov 16  2012 save -> /save/cabezon
drwxr-xr-x  3 cabezon physik       19 Jan 12  2015 software/
drwx------  2 cabezon physik       95 Jan 16  2015 .ssh/
drwxr-xr-x  6 cabezon physik     4096 Jan 22  2015 tmp/
-rw-------  1 cabezon physik     8260 Aug 20 17:18 .viminfo
-rw-------  1 cabezon physik    13390 Aug 26 14:53 .Xauthority
-rw-r--r--  1 cabezon physik      658 Jun 23  2012 .zshrc
[cabezon@login11 ~]$
```

Links: number of links pointing to this file/directory.

# More about files

```
cabezon@login11:~
-rw-r--r--  1 cabezon physik     121 Jun 22  2012 .kshrc
-rw-------  1 cabezon physik     234 Aug 19 16:26 .lesshst
drwxrwxr-x  3 cabezon physik      19 Jan 16  2015 .lmod.d/
drwx------  3 cabezon physik      18 Dec 11  2013 .local/
drwxr-xr-x  7 cabezon physik      74 Jan 22  2015 martin/
drwxr-xr-x  3 cabezon physik      19 Jan 24  2014 .matlab/
drwxr-xr-x  2 cabezon physik      27 May  2  2013 .matplotlib/
drwx------  2 cabezon physik      58 Jan 19  2015 .mc/
drwxr-xr-x  5 cabezon physik      51 May  2  2013 .mozilla/
-rw-r--r--  1 cabezon physik    2458 Sep 12  2014 new_absor_effect.f90
-rw-r--r--  1 cabezon physik   18240 May  8 11:55 nuprox3D.f90
drwxr-----  3 cabezon physik      18 May  2  2013 .pki/
-rw-r--r--  1 cabezon physik    6525 Jun 23 12:05 progenitor.f
-rw-------  1 cabezon physik     256 May  2  2013 .pulse-cookie
-rw-r--r--  1 cabezon physik    1092 Jan 22  2015 REPORT
drwxr-xr-x 21 cabezon physik    4096 Aug 25 09:33 research/
lrwxrwxrwx  1 root    root       13 Nov 16  2012 save -> /save/cabezon
drwxr-xr-x  3 cabezon physik      19 Jan 12  2015 software/
drwx------  2 cabezon physik      95 Jan 16  2015 .ssh/
drwxr-xr-x  6 cabezon physik    4096 Jan 22  2015 tmp/
-rw-------  1 cabezon physik    8260 Aug 20 17:18 .viminfo
-rw-------  1 cabezon physik   13390 Aug 26 14:53 .Xauthority
-rw-r--r--  1 cabezon physik     658 Jun 23  2012 .zshrc
[cabezon@login11 ~]$
```

Owner: Usually the user who created the file.

# More about files

```
cabezon@login11:~
-rw-r--r--  1 cabezon physik     121 Jun 22  2012 .kshrc
-rw-------  1 cabezon physik     234 Aug 19 16:26 .lesshst
drwxrwxr-x  3 cabezon physik      19 Jan 16  2015 .lmod.d/
drwx------  3 cabezon physik      18 Dec 11  2013 .local/
drwxr-xr-x  7 cabezon physik      74 Jan 22  2015 martin/
drwxr-xr-x  3 cabezon physik      19 Jan 24  2014 .matlab/
drwxr-xr-x  2 cabezon physik      27 May  2  2013 .matplotlib/
drwx------  2 cabezon physik      58 Jan 19  2015 .mc/
drwxr-xr-x  5 cabezon physik      51 May  2  2013 .mozilla/
-rw-r--r--  1 cabezon physik    2458 Sep 12  2014 new_absor_effect.f90
-rw-r--r--  1 cabezon physik   18240 May  8 11:55 nuprox3D.f90
drwxr-----  3 cabezon physik      18 May  2  2013 .pki/
-rw-r--r--  1 cabezon physik    6525 Jun 23 12:05 progenitor.f
-rw-------  1 cabezon physik     256 May  2  2013 .pulse-cookie
-rw-r--r--  1 cabezon physik    1092 Jan 22  2015 REPORT
drwxr-xr-x 21 cabezon physik    4096 Aug 25 09:33 research/
lrwxrwxrwx  1 root    root       13 Nov 16  2012 save -> /save/cabezon
drwxr-xr-x  3 cabezon physik      19 Jan 12  2015 software/
drwx------  2 cabezon physik      95 Jan 16  2015 .ssh/
drwxr-xr-x  6 cabezon physik    4096 Jan 22  2015 tmp/
-rw-------  1 cabezon physik    8260 Aug 20 17:18 .viminfo
-rw-------  1 cabezon physik   13390 Aug 26 14:53 .Xauthority
-rw-r--r--  1 cabezon physik     658 Jun 23  2012 .zshrc
[cabezon@login11 ~]$
```

Group: Set of users that can access the file according to the permissions specified in the group field.

# More about files

```
cabezon@login11:~
-rw-r--r--  1 cabezon physik     121 Jun 22  2012 .kshrc
-rw-------  1 cabezon physik     234 Aug 19 16:26 .lesshst
drwxrwxr-x  3 cabezon physik      19 Jan 16  2015 .lmod.d/
drwx------  3 cabezon physik      18 Dec 11  2013 .local/
drwxr-xr-x  7 cabezon physik      74 Jan 22  2015 martin/
drwxr-xr-x  3 cabezon physik      19 Jan 24  2014 .matlab/
drwxr-xr-x  2 cabezon physik      27 May  2  2013 .matplotlib/
drwx------  2 cabezon physik      58 Jan 19  2015 .mc/
drwxr-xr-x  5 cabezon physik      51 May  2  2013 .mozilla/
-rw-r--r--  1 cabezon physik    2458 Sep 12  2014 new_absor_effect.f90
-rw-r--r--  1 cabezon physik   18240 May  8 11:55 nuprox3D.f90
drwxr-----  3 cabezon physik      18 May  2  2013 .pki/
-rw-r--r--  1 cabezon physik    6525 Jun 23 12:05 progenitor.f
-rw-------  1 cabezon physik     256 May  2  2013 .pulse-cookie
-rw-r--r--  1 cabezon physik    1092 Jan 22  2015 REPORT
drwxr-xr-x 21 cabezon physik    4096 Aug 25 09:33 research/
lrwxrwxrwx  1 root     root       13 Nov 16  2012 save -> /save/cabezon
drwxr-xr-x  3 cabezon physik      19 Jan 12  2015 software/
drwx------  2 cabezon physik      95 Jan 16  2015 .ssh/
drwxr-xr-x  6 cabezon physik    4096 Jan 22  2015 tmp/
-rw-------  1 cabezon physik    8260 Aug 20 17:18 .viminfo
-rw-------  1 cabezon physik   13390 Aug 26 14:53 .Xauthority
-rw-r--r--  1 cabezon physik     658 Jun 23  2012 .zshrc
[cabezon@login11 ~]$
```

Size: expressed in bytes. ( −h option is more ergonomic)

# More about files

```
cabezon@login11:~
-rw-r--r--  1 cabezon physik      121 Jun 22  2012 .kshrc
-rw-------  1 cabezon physik      234 Aug 19 16:26 .lesshst
drwxrwxr-x  3 cabezon physik       19 Jan 16  2015 .lmod.d/
drwx------  3 cabezon physik       18 Dec 11  2013 .local/
drwxr-xr-x  7 cabezon physik       74 Jan 22  2015 martin/
drwxr-xr-x  3 cabezon physik       19 Jan 24  2014 .matlab/
drwxr-xr-x  2 cabezon physik       27 May  2  2013 .matplotlib/
drwx------  2 cabezon physik       58 Jan 19  2015 .mc/
drwxr-xr-x  5 cabezon physik       51 May  2  2013 .mozilla/
-rw-r--r--  1 cabezon physik     2458 Sep 12  2014 new_absor_effect.f90
-rw-r--r--  1 cabezon physik    18240 May  8 11:55 nuprox3D.f90
drwxr-----  3 cabezon physik       18 May  2  2013 .pki/
-rw-r--r--  1 cabezon physik     6525 Jun 23 12:05 progenitor.f
-rw-------  1 cabezon physik      256 May  2  2013 .pulse-cookie
-rw-r--r--  1 cabezon physik     1092 Jan 22  2015 REPORT
drwxr-xr-x 21 cabezon physik     4096 Aug 25 09:33 research/
lrwxrwxrwx  1 root    root        13 Nov 16  2012 save -> /save/cabezon
drwxr-xr-x  3 cabezon physik       19 Jan 12  2015 software/
drwx------  2 cabezon physik       95 Jan 16  2015 .ssh/
drwxr-xr-x  6 cabezon physik     4096 Jan 22  2015 tmp/
-rw-------  1 cabezon physik     8260 Aug 20 17:18 .viminfo
-rw-------  1 cabezon physik    13390 Aug 26 14:53 .Xauthority
-rw-r--r--  1 cabezon physik      658 Jun 23  2012 .zshrc
[cabezon@login11 ~]$
```

Date: Date and time of last modification (write).

The −u option displays time of last access (read).

# More about files

```
cabezon@login11:~
-rw-r--r--  1 cabezon physik      121 Jun 22  2012 .kshrc
-rw-------  1 cabezon physik      234 Aug 19 16:26 .lesshst
drwxrwxr-x  3 cabezon physik       19 Jan 16  2015 .lmod.d/
drwx------  3 cabezon physik       18 Dec 11  2013 .local/
drwxr-xr-x  7 cabezon physik       74 Jan 22  2015 martin/
drwxr-xr-x  3 cabezon physik       19 Jan 24  2014 .matlab/
drwxr-xr-x  2 cabezon physik       27 May  2  2013 .matplotlib/
drwx------  2 cabezon physik       58 Jan 19  2015 .mc/
drwxr-xr-x  5 cabezon physik       51 May  2  2013 .mozilla/
-rw-r--r--  1 cabezon physik     2458 Sep 12  2014 new_absor_effect.f90
-rw-r--r--  1 cabezon physik    18240 May  8 11:55 nuprox3D.f90
drwxr-----  3 cabezon physik       18 May  2  2013 .pki/
-rw-r--r--  1 cabezon physik     6525 Jun 23 12:05 progenitor.f
-rw-------  1 cabezon physik      256 May  2  2013 .pulse-cookie
-rw-r--r--  1 cabezon physik     1092 Jan 22  2015 REPORT
drwxr-xr-x 21 cabezon physik     4096 Aug 25 09:33 research/
lrwxrwxrwx  1 root    root        13 Nov 16  2012 save -> /save/cabezon
drwxr-xr-x  3 cabezon physik       19 Jan 12  2015 software/
drwx------  2 cabezon physik       95 Jan 16  2015 .ssh/
drwxr-xr-x  6 cabezon physik     4096 Jan 22  2015 tmp/
-rw-------  1 cabezon physik     8260 Aug 20 17:18 .viminfo
-rw-------  1 cabezon physik    13390 Aug 26 14:53 .Xauthority
-rw-r--r--  1 cabezon physik      658 Jun 23  2012 .zshrc
[cabezon@login11 ~]$
```

Name: this is self-explaining…

# More about rights

```
cabezon@login11:~
-rw-r--r--  1 cabezon physik     121 Jun 22  2012 .kshrc
-rw-------  1 cabezon physik     234 Aug 19 16:26 .lesshst
drwxrwxr-x  3 cabezon physik      19 Jan 13  2015 .lmod.d/
drwx------  [illegible]
drwxr-xr-x  [illegible]in/
drwxr-xr-x  3 cabezon physik      19 Jan 24  2014 .matlab/
drwxr-xr-x  2 cabezon physik      27 May  2  2013 .matplotlib/
drwx------  2 cabezon physik      58 Jan 19  2015 .mc/
drwxr-xr-x  5 cabezon physik      51 May  2  2013 .mozilla/
-rw-r--r--  1 cabezon physik    2458 Sep 12  2014 new_absor_effect.f90
-rw-r--r--  1 cabezon physik   18240 May  8 11:55 nuprox3D.f90
drwxr-----  3 cabezon physik      18 May  2  2013 .pki/
-rw-r--r--  1 cabezon physik    6525 Jun 23 12:05 progenitor.f
-rw-------  1 cabezon physik     256 May  2  2013 .pulse-cookie
-rw-r--r--  1 cabezon physik    1092 Jan 22  2015 REPORT
drwxr-xr-x 21 cabezon physik    4096 Aug 25 09:33 research/
lrwxrwxrwx  1 root     root       13 Nov 16  2012 save -> /save/cabezon
drwxr-xr-x  3 cabezon physik      19 Jan 12  2015 software/
drwx------  2 cabezon physik      95 Jan 16  2015 .ssh/
drwxr-xr-x  6 cabezon physik    4096 Jan 22  2015 tmp/
-rw-------  1 cabezon physik    8260 Aug 20 17:18 .viminfo
-rw-------  1 cabezon physik   13390 Aug 26 14:53 .Xauthority
-rw-r--r--  1 cabezon physik     658 Jun 23  2012 .zshrc
[cabezon@login11 ~]$
```

There are 9 characters grouped in 3 sets of 3.
Each set defines permissions to 3 access types: r (read), w (write), x (execute)
An hyphen – means no permission to that access type.

# More about rights

```
cabezon@login11:~
-rw-r--r--  1 cabezon physik     121 Jun 22  2012 .kshrc
-rw-------  1 cabezon physik     234 Aug 19 16:26 .lesshst
drwxrwxr-x  3 cabezon physik      19 Jan 13  2015 .lmod.d/
drwx------  2 cabezon physik      95 Jan 16  2015 .ssh/
drwxr-xr-x  6 cabezon physik    4096 Jan 22  2015 tmp/
-rw-------  1 cabezon physik    8260 Aug 20 17:18 .viminfo
-rw-------  1 cabezon physik   13390 Aug 26 14:53 .Xauthority
-rw-r--r--  1 cabezon physik     658 Jun 23  2012 .zshrc
[cabezon@login11 ~]$
```

There are 9 characters grouped in 3 sets of 3.
Each set defines permissions to 3 access types: r (read), w (write), x (execute)
An hyphen – means no permission to that access type.

Each set refers to different users: **owner**, **group**, **other**

# More about rights

```
cabezon@login11:~
-rw-r--r--  1 cabezon physik     121 Jun 22  2012 .kshrc
-rw-------  1 cabezon physik     234 Aug 19 16:26 .lesshst
drwxrwxr-x  3 cabezon physik      19 Jan 8  2015 .lmod.d/
drwx------  2 cabezon physik      [illegible] [illegible] [illegible] [illegible] [illegible]in/
drwxr-xr-x  3 cabezon physik      19 Jan 24  2014 .matlab/
drwxr-xr-x  [illegible] [illegible] [illegible] [illegible] [illegible] [illegible]matplotlib/
drwx------  2 cabezon physik      58 Jan 19  2015 .mc/
drwxr-xr-x  5 cabezon physik      51 May  2  2013 .mozilla/
-rw-r--r--  1 cabezon physik    2458 Sep 12  2014 new_absor_effect.f90
-rw-r--r--  1 cabezon physik   18240 May  8 11:55 nuprox3D.f90
drwxr-----  3 cabezon physik      18 May  2  2013 .pki/
-rw-r--r--  1 cabezon physik    6525 Jun 23 12:05 progenitor.f
-rw-------  1 cabezon physik     256 May  2  2013 .pulse-cookie
-rw-r--r--  1 cabezon physik    1092 Jan 22  2015 REPORT
drwxr-xr-x 21 cabezon physik    4096 Aug 25 09:33 research/
lrwxrwxrwx  1 root     root       13 Nov 16  2012 save -> /save/cabezon
drwxr-xr-x  3 cabezon physik      19 Jan 12  2015 software/
drwx------  2 cabezon physik      95 Jan 16  2015 .ssh/
drwxr-xr-x  6 cabezon physik    4096 Jan 22  2015 tmp/
-rw-------  1 cabezon physik    8260 Aug 20 17:18 .viminfo
-rw-------  1 cabezon physik   13390 Aug 26 14:53 .Xauthority
-rw-r--r--  1 cabezon physik     658 Jun 23  2012 .zshrc
[cabezon@login11 ~]$
```

There are 9 characters grouped in 3 sets of 3.
Each set defines permissions to 3 access types: r (read), w (write), x (execute)
An hyphen – means no permission to that access type.

Each set refers to different users: **owner**, **group**, **other**

Example:

# More about rights

```
cabezon@login11:~
-rw-r--r--  1 cabezon physik     121 Jun 22  2012 .kshrc
-rw-------  1 cabezon physik     234 Aug 19 16:26 .lesshst
drwxrwxr-x  3 cabezon physik      19 Jan 8  2015 .lmod.d/
drwx------  3 cabezon physik      19 Jan 24 2014 .local/
drwxr-xr-x  3 cabezon physik      19 Jan 24  2014 .matlab/
drwxr-xr-x  3 cabezon physik      37 May  2  2013 .matplotlib/
drwxr-xr-x  3 cabezon physik      37 May  2  2013 .matplotlib/
drwx------  2 cabezon physik      58 Jan 19  2015 .mc/
drwxr-xr-x  5 cabezon physik      51 May  2  2013 .mozilla/
-rw-r--r--  1 cabezon physik    2458 Sep 12  2014 new_absor_effect.f90
-rw-r--r--  1 cabezon physik   18240 May  8 11:55 nuprox3D.f90
drwxr-----  3 cabezon physik      18 May  2  2013 .pki/
-rw-r--r--  1 cabezon physik    6525 Jun 23 12:05 progenitor.f
-rw-------  1 cabezon physik     256 May  2  2013 .pulse-cookie
-rw-r--r--  1 cabezon physik    1092 Jan 22  2015 REPORT
drwxr-xr-x 21 cabezon physik    4096 Aug 25 09:33 research/
lrwxrwxrwx  1 root     root       13 Nov 16  2012 save -> /save/cabezon
drwxr-xr-x  3 cabezon physik      19 Jan 12  2015 software/
drwx------  2 cabezon physik      95 Jan 16  2015 .ssh/
drwxr-xr-x  6 cabezon physik    4096 Jan 22  2015 tmp/
-rw-------  1 cabezon physik    8260 Aug 20 17:18 .viminfo
-rw-------  1 cabezon physik   13390 Aug 26 14:53 .Xauthority
-rw-r--r--  1 cabezon physik     658 Jun 23  2012 .zshrc
[cabezon@login11 ~]$
```

There are 9 characters grouped in 3 sets of 3.
Each set defines permissions to 3 access types: r (read), w (write), x (execute)
An hyphen – means no permission to that access type.

Each set refers to different users: **owner**, **group**, **other**

Example: REDACTED

# More about rights

```
cabezon@login11:~
-rw-r--r--
-rw-------
drwxrwxr-x
drwx------
drwxr-xr-x
drwxr-xr-x
drwxr-xr-x
drwx------
drwxr-xr-x
-rw-r--r--
-rw-r--r--
drwxr-----
-rw-r--r--
-rw-------
-rw-r--r--
drwxr-xr-x
lrwxrwxrwx
drwxr-xr-x
drwx------
drwxr-xr-x
-rw-------
-rw-------
-rw-r--r--
[cabezon@login11 ~]$
```

There are 9 characters grouped in 3 sets of 3.
Each set defines permissions to 3 access types: r (read), w (write), x (execute)
An hyphen – means no permission to that access type.

Each set refers to different users: **owner**, **group**, **other**

Example: REDACTED

chmod: used to change permissions.

**Symbolic:** uses **u**, **g**, **o**, **a** an operator **+**, **-** and a mode **r**, **w**, **x**

**Ex:** chmod g+w <file> : gives write permission to the group users

chmod –R a-wx <directory> : removes writing and executing rights to all users, recursively for the directory tree inside <directory>

**Octal:** uses three digits to define a combination of rights for each set. It's simply the decimal number corresponding to the binary resulting from the set of permissions.

**Ex:** - - x = 001 = 1, r - x = 101 = 5, rwx = 111 = 7

chmod 662 <file> =

# More about rights

cabezon@login11:~

```
-rw-r--r--
-rw-------
drwxrwxr-x
drwx------
drwxr-xr-x
drwxr-xr-x
drwxr-xr-x
drwx------
drwxr-xr-x
-rw-r--r--
-rw-r--r--
drwxr-----
-rw-r--r--
-rw-------
-rw-r--r--
drwxr-xr-x
lrwxrwxrwx
drwxr-xr-x
drwx------
drwxr-xr-x
-rw-------
-rw-------
-rw-r--r--
[cabezon@login11 ~]$
```

There are 9 characters grouped in 3 sets of 3.
Each set defines permissions to 3 access types: r (read), w (write), x (execute)
An hyphen – means no permission to that access type.

Each set refers to different users: **owner**, **group**, **other**

Example: REDACTED

chmod: used to change permissions.

**Symbolic:** uses **u**, **g**, **o**, **a** an operator **+**, **-** and a mode **r**, **w**, **x**

**Ex:** chmod g+w <file> : gives write permission to the group users

chmod –R a-wx <directory> : removes writing and executing rights to all users, recursively for the directory tree inside <directory>

**Octal:** uses three digits to define a combination of rights for each set. It's simply the decimal number corresponding to the binary resulting from the set of permissions.

**Ex:** - - x = 001 = 1, r - x = 101 = 5, rwx = 111 = 7

chmod 662 <file> = REDACTED

There are only 10 types of people in this world:
those who understand binary and those who don't.

# More about ownership

```
cabezon@login11:~
-rw-r--r--  1 cabezon physik    121 Jun 22  2012 .kshrc
-rw-------  1 cabezon physik    234 Aug 19 16:26 .lesshst
drwxrwxr-x  3 cabezon physik      8 Jan 16  2015 .local/
drwx------  3 cabezon physik     74 Jan 22  2015 .mozilla/
drwxr-xr-x  7 cabezon physik     74 Jan 22  2015 martin/
drwxr-xr-x  3 cabezon physik     19 Jan 24  2014 .matlab/
drwxr-xr-x  2 cabezon physik     37 May  2  2013 .mozilla-extensions/
drwx------  2 cabezon physik     58 Jan 19  2015 .mc/
drwxr-xr-x  5 cabezon physik     51 May  2  2013 .mozilla/
-rw-r--r--  1 cabezon physik      [illegible] [illegible] [illegible] [illegible]
-rw-r--r--  1 cabezon physik  18240 May  8 11:55 nuprox3D.f90
drwxr-----  3 cabezon physik      [illegible] [illegible] [illegible] [illegible]
-rw-r--r--  1 cabezon physik      [illegible] [illegible] [illegible] [illegible]
-rw-------  1 cabezon physik     256 May  2  2013 .pulse-cookie
-rw-r--r--  1 cabezon physik    1092 Jan 22  2015 REPORT
drwxr-xr-x 21 cabezon physik    4096 Aug 25 09:33 research/
lrwxrwxrwx  1 root    root        13 Nov 16  2012 save -> /save/cabezon
drwxr-xr-x  3 cabezon physik      19 Jan 12  2015 software/
drwx------  2 cabezon physik      95 Jan 16  2015 .ssh/
drwxr-xr-x  6 cabezon physik    4096 Jan 22  2015 tmp/
-rw-------  1 cabezon physik    8260 Aug 20 17:18 .viminfo
-rw-------  1 cabezon physik   13390 Aug 26 14:53 .Xauthority
-rw-r--r--  1 cabezon physik     658 Jun 23  2012 .zshrc
[cabezon@login11 ~]$
```

Usually our home is blocked to other users, unless we have allowed read permission to users that are share a group with us.

This is a way of safely sharing data by allowing access.

chown: allows to change the ownership of a file/directory

**Ex:** chown alice <file> : changes ownership to alice
chown –R alice:physik <directory> : changes recursively ownership to alice and group to physik.

# 2. Exercise

1. Log in to the cluster
2. Check your path: pwd. That's your home directory.
3. Check the content of the environment variable $HOME: echo $HOME
4. Enter in linux_course/ and create a directory named Exercise2/
5. Create a file: cat > hello1.txt
6. Write hello world and exit (CTRL+D twice)
7. Create another file. Do emacs hello2.txt
8. Write something inside, save it (CTRL+X, CTRL+S) and exit (CTRL+X, CTRL+C)
9. Create another file: nano hello3.txt
10. Write hello world again and exit (CTRL+X). Save when asked.
11. Create another file: vim hello4.txt
12. Press i to go to insert mode. Write something. Press ESC and exit (:wq)
13. Write cp he and press TAB. What happens? Press TAB twice. Add a 4 and press TAB. What happens?
14. Finish the command to copy hello4.txt into hello5.txt
15. Go to your home/ with cd
16. Find where all hello files are: find linux_course/ –name 'hello*'
17. From your home directory and with one command move all hello files from linux_course/ to linux_course/Exercise2/
18. Go to Exercise2/ and check that all 5 hello files are there.

The final outcome should be a directory structure linux_course/Exercise2 that includes 5 files named hello1.txt, hello2.txt,…

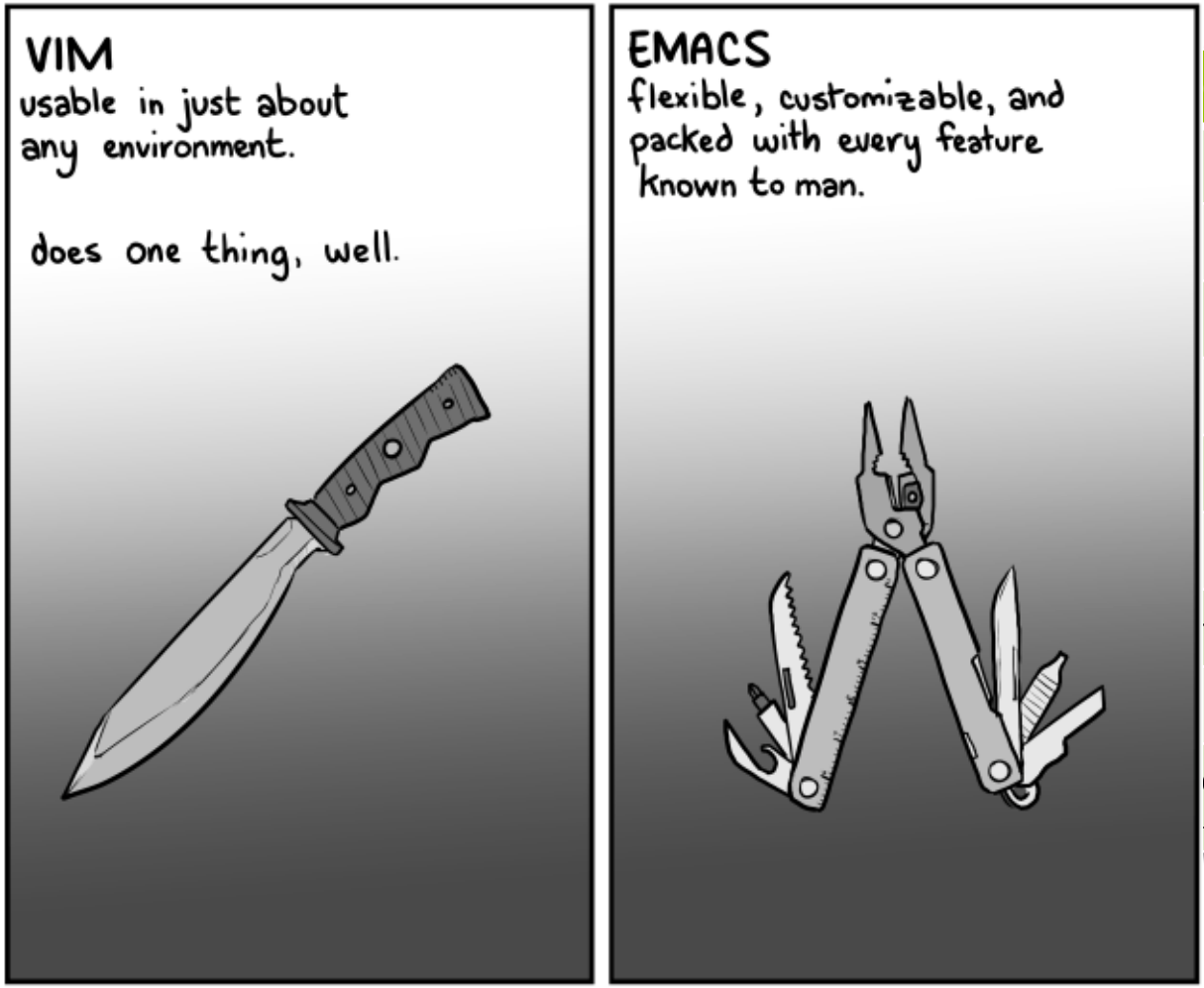
## Summary

| Command | Used to… | Common options |
|---|---|---|
| pwd | print current working directory | |
| echo | output text to screen | |
| cat | displays/create file contents | |
| emacs | displays/create file contents | -nw (no window) |
| nano | displays/create file contents | |
| vim | displays/create file contents | |
| find | search for a file | <path> -name '<file>' |
| <TAB> | auto completion | |
| $variable | dynamic values that can control how processes behave | $HOME, $PATH, $TMPDIR, $LD_LIBRARY_PATH, $PWD |

# 2. Exercise

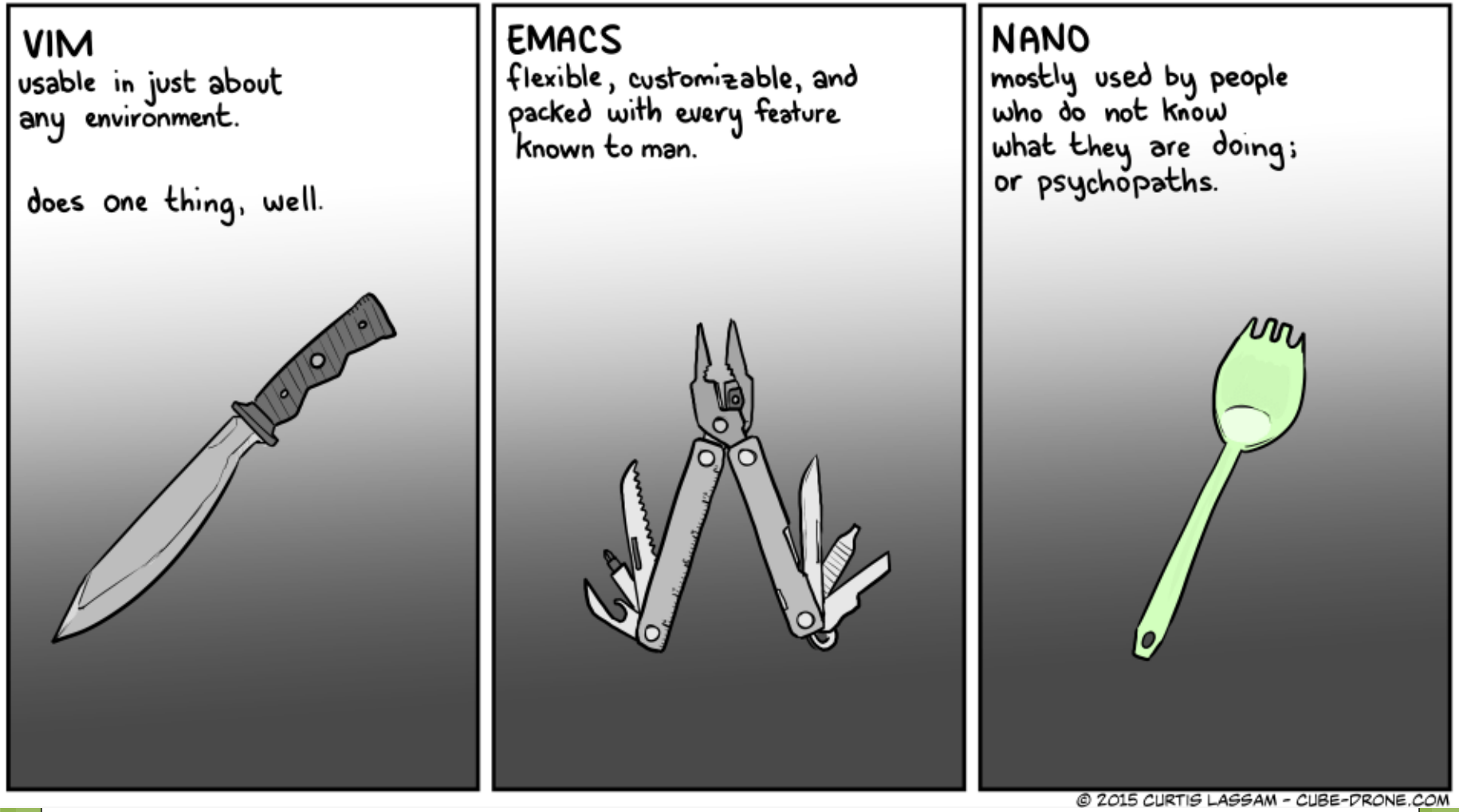

| sed to… | Common options |
|---|---|
| nt current working directory | |
| tput text to screen | |
| splays/create file contents | |
| splays/create file contents | -nw (no window) |
| splays/create file contents | |
| splays/create file contents | |
| arch for a file | <path> -name '<file>' |
| to completion | |
| namic values that can control w processes behave | $HOME, $PATH, $TMPDIR, $LD_LIBRARY_PATH, $PWD |

# 2. Exercise




| mmon options |
| --- |
| |
| |
| |
| (no window) |
| |
| |
| th> -name '<file>' |
| |
| ME, $PATH, $TMPDIR, _LIBRARY_PATH, $PWD |

# 2. Exercise




Want to learn vim? Check http://vim-adventures.com

# 3. Exercise

1. Log in to the cluster
2. Enter in linux_course/ and create a directory named Exercise3/
3. Find in which files from Exercise2/ is 'hello' written: grep 'hello' Exercise2/*
4. Compress the Exercise2/ directory: tar czvf exercise2.tar.gz Exercise2/
5. Move exercise2.tar.gz to Exercise3/
6. Enter in Exercise3/
7. Extract the contents of the tar ball: tar xvf exercise2.tar.gz
8. Check the contents of Exercise3/
9. Rename the newly created folder Exercise2/ to data/
10. Enter in data/ and check the size of all files: ll (these are two letters!)
11. Compress all txt files : gzip *.txt and check the size of the files. Comments?
12. Extract all compressed files: gunzip *.gz
13. Get this data file:

    cp /shared/data/linux/tau_nue70000.d .
14. Check its contents with less.
15. Check size of the data file, compress it and check it again.
16. Copy and rename the .gz file as data.d.gz with one single instruction.
17. Extract data.d.gz. Note that the uncompressed file has also the new name.
18. Extract tau_nue70000.d.gz
19. Check that both files are the same: diff -s data.d tau_nue70000.d
20. Edit data.d, change one number, save it, and diff again.
21. Print the second and third columns of data.d in a file named out.d:

    awk '{print $2 " " $3}' data.d > out.d
22. Check the contents of out.d and delete data.d
23. Go to your home and check the size of linux_course/: du –h linux_course

## Summary

| Command | Used to… | Common options |
|---|---|---|
| grep | search file (or stdin) for lines matching a pattern | -i (case insensitive), -w (only) |
| tar | back up entire directories and file into a single container file | -c (create), -v (verbose),<br>-f (file), -z (zip), -x (extract),<br>-t (list) |
| gzip | compress individual files | -d (decompress) |
| gunzip | decompress individual files | |
| diff | display differences between two files | -s (report if files are identical) |
| awk | manipulate files with data in columns | |
| du | show the disk usage of a directory | -h (human readable)<br>-s (disply only the sum) |

# Extra tips

You can use & at the end of a command to keep using the command line and allow the execution to proceed in the background. Ex: emacs datafile &

time in front of any command will provide the time it takes to execute. Ex: time /bin/sleep 5

history shows a numbered list of the commands that have been used. With !<number> that command is re-used. With CTRL+R it performs a reverse search in the history.

top gives you information about running processes, memory and CPU consumption in realtime.

htop is a enhanced version of top, which also gives a pseudo-graphical load / CPU.

ps aux lists all processes, like top, but as a snapshot.

$HOME is an environment variable that points to your home/ directory.

cd without arguments will always lead you to cd $HOME

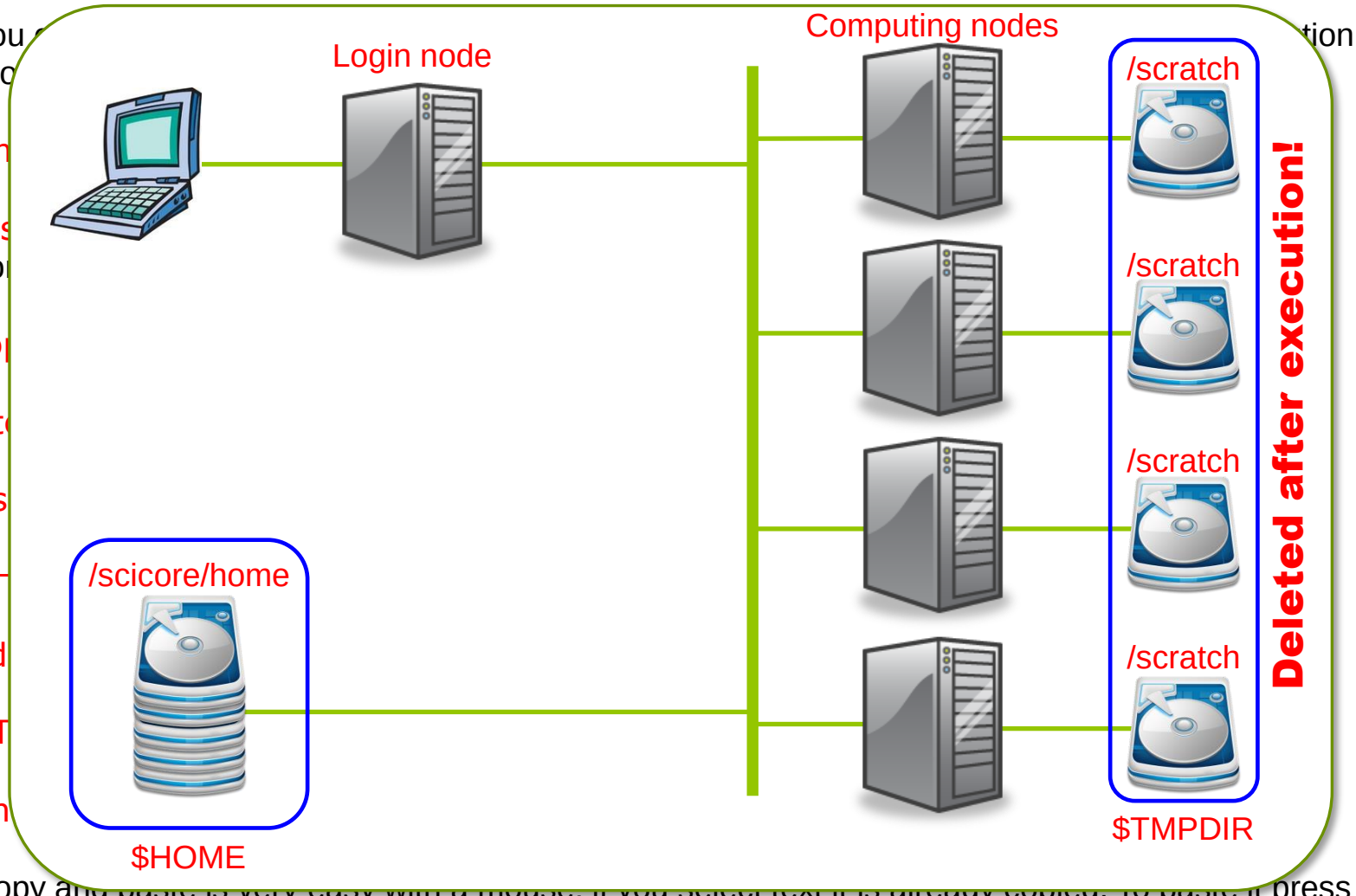
$TMPDIR points to a temporary directory

env will show all enviroment variables

Copy and paste is very easy with a mouse: if you select text it is already copied. To paste it press the central button of the mouse (if you have one…) or right-click and paste.

# Extra tips

You
pro
tin
his
co
to
ht
ps
$H
cd
$T
en




tion to

Copy and paste is very easy with a mouse. If you select text it is already copied. To paste it press the central button of the mouse (if you have one…) or right-click and paste.

# Extra tips

| Event | Latency | Scaled |
|---|---|---|
| 1 CPU cycle | 0.3 ns | 1 s |
| Level 1 cache access | 0.9 ns | 3 s |
| Level 2 cache access | 2.8 ns | 9 s |
| Level 3 cache access | 12.9 ns | 43 s |
| Main memory access (DRAM, from CPU) | 120 ns | 6 min |
| Solid-state disk I/O (flash memory) | 50–150 μs | 2–6 days |
| Rotational disk I/O | 1–10 ms | 1–12 months |
| Internet: San Francisco to New York | 40 ms | 4 years |
| Internet: San Francisco to United Kingdom | 81 ms | 8 years |
| Internet: San Francisco to Australia | 183 ms | 19 years |
| TCP packet retransmit | 1–3 s | 105–317 years |
| OS virtualization system reboot | 4 s | 423 years |
| SCSI command time-out | 30 s | 3 millennia |
| Hardware (HW) virtualization system reboot | 40 s | 4 millennia |
| Physical system reboot | 5 m | 32 millennia |

# Extra tips

You can use grep attached to any command with the pipe | This will allow you to show only the information you are interested in. Ex: history vs history | grep ls

In fact, pipes can be used with almost all commands. They transfer the output of the first command as input for the second one, and so on.
Ex: ps aux | grep -w grep | awk '{print $2}' Can you explain what does this command do?

file shows the file type of a file.

wc shows the number of lines, words and characters of a file.

cat prints the content of a file in the screen and it also concatenates files.

head prints the first 10 lines of a file (or the first N lines, with the option -n <N>)

tail prints the last 10 lines of a file (or the last N lines, with the option -n <N>)

Can you write a command that prints the $N^{th}$ line of a text file?
**HINT:** use head, tail and |

# Extra tips

REDACTED

In fact, pipes can be used with almost all commands. They transfer the output of the first command as input for the second one, and so on.
Ex: ps au ... ommand do?

copy-past ... already copied. To

file shows

wc shows

cat prints

head print

REDACTED

Check http://tldp.org/LDP/abs/html/special-chars.html for a summary of all special characters used in scripts.

Can you write a command that prints the N-line of a text file?
**HINT:** use cat, head, tail and |

# Outline


09:30 – Course starts
09:30 – 11:00: Introduction to Linux. Part 1
11:00 – 11:15: Coffee break
11:15 – 12:30: Introduction to Linux. Part 2

12:30 – 14:00: Lunch

14:30 – 16:00: Cluster usage. Part 1
16:00 – 16:15: Coffee break
16:15 – 17:30: Cluster usage. Part 2
17:30 – 18:00: Feedback + Q&A

# Clusters

Cluster are HPC infrastructures:

- Parallel computing servers
- Large-memory servers
- Large-storage capacity

Using HPC facilities <u>does not</u> mean to do the same as in your laptop but in bigger machines!

- Planning
- Knowing your needs
- Being familiar with the code
- Respecting the rules


vs.

# People in sciCORE

Who is who in sciCORE?

scicore-admin@unibas.ch

Steering board


Torsten


Michael

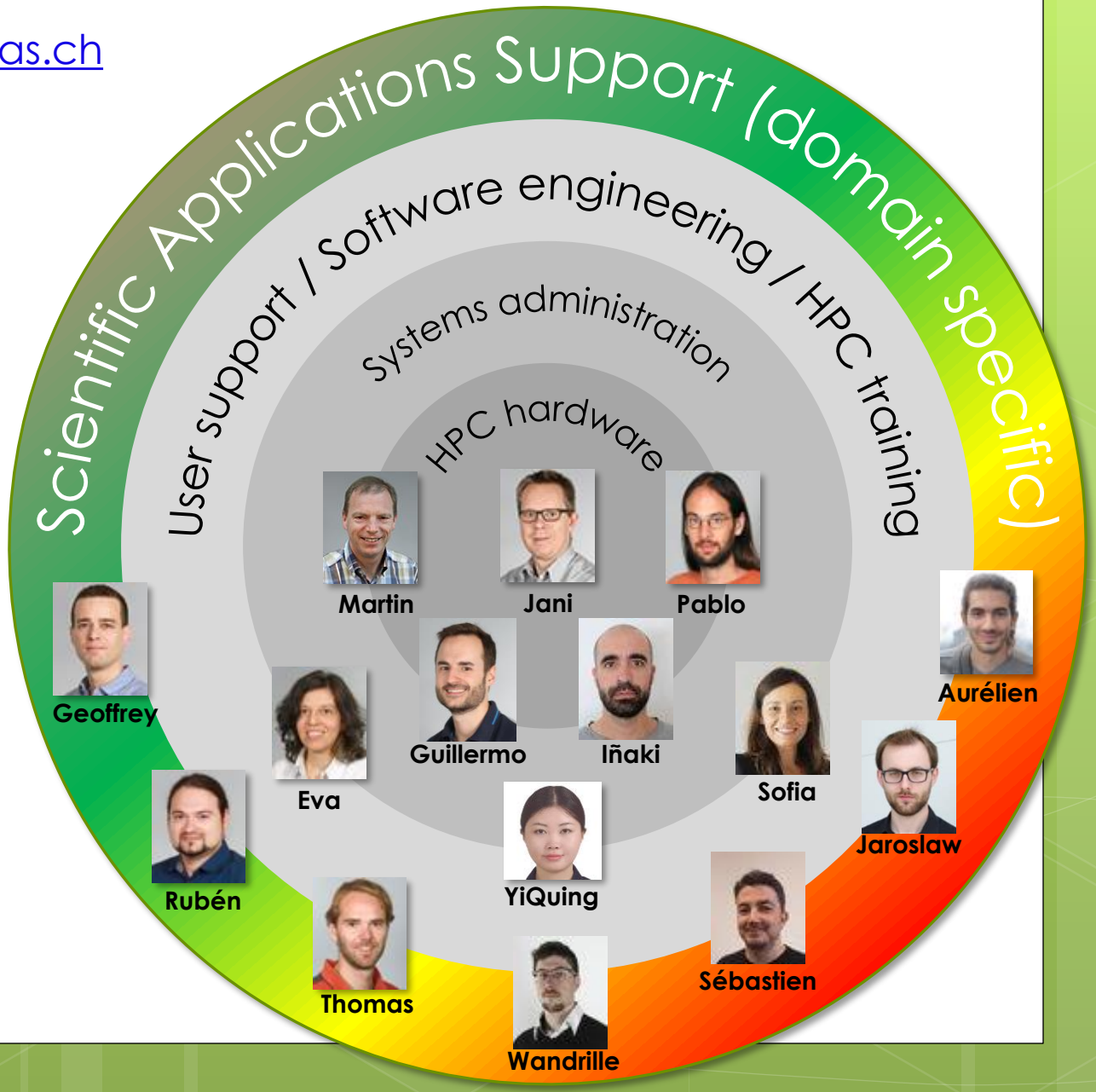

Timm

Management & Administration


Thierry

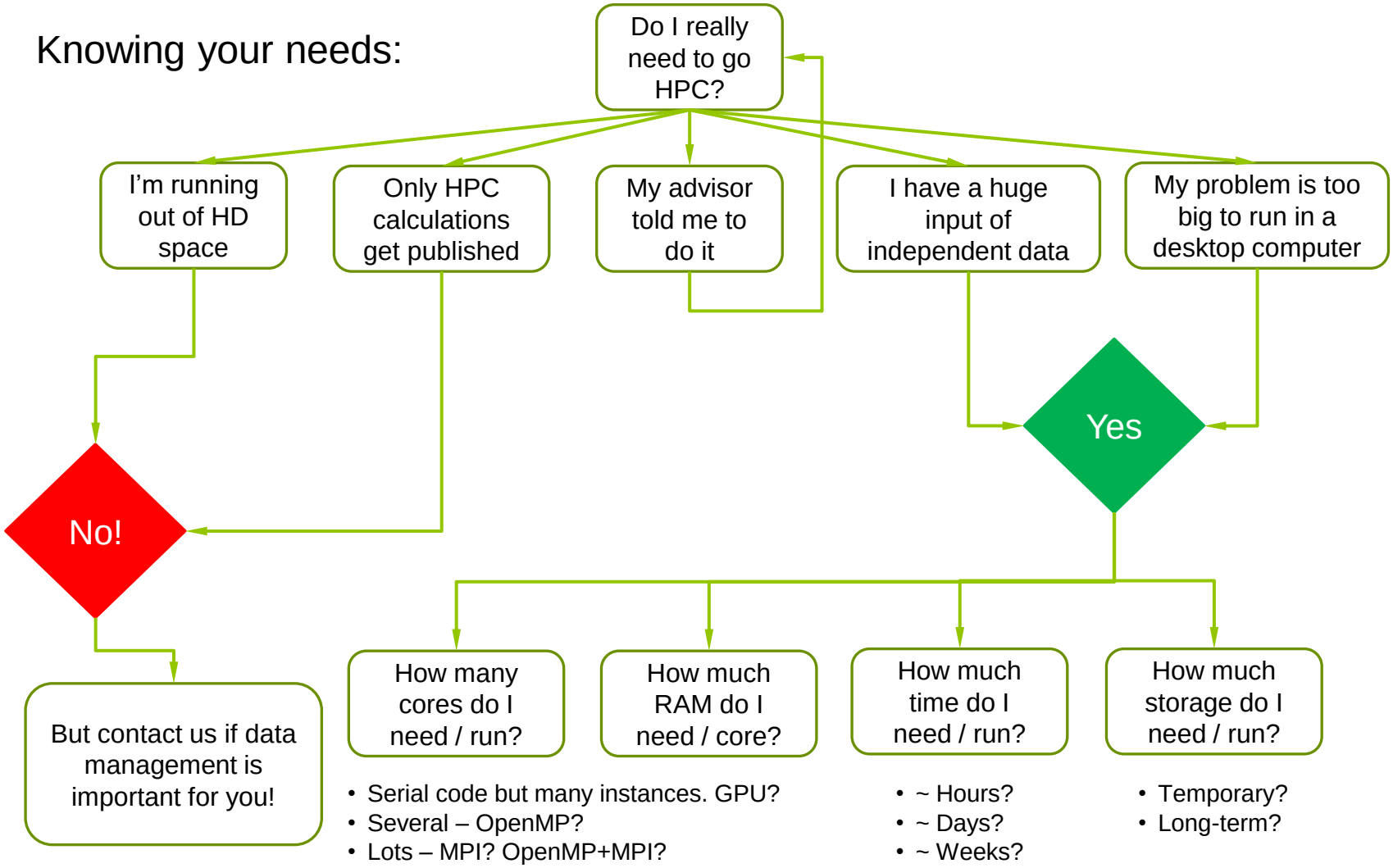

Lorenza


Ariadna

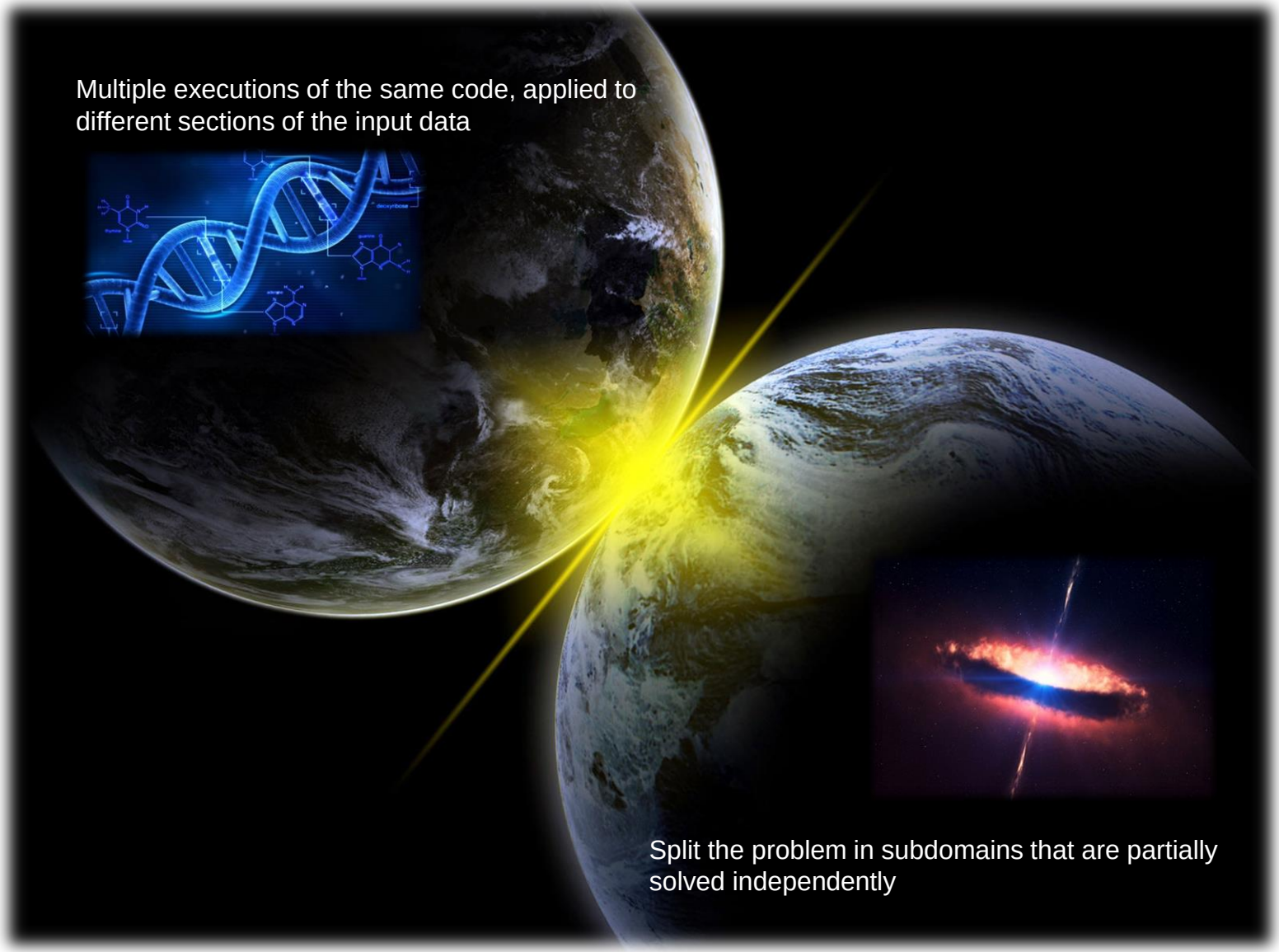


# Using a cluster

Knowing your needs:



# Using a cluster



# Using sciCORE

2 login nodes (front-end to access cluster)

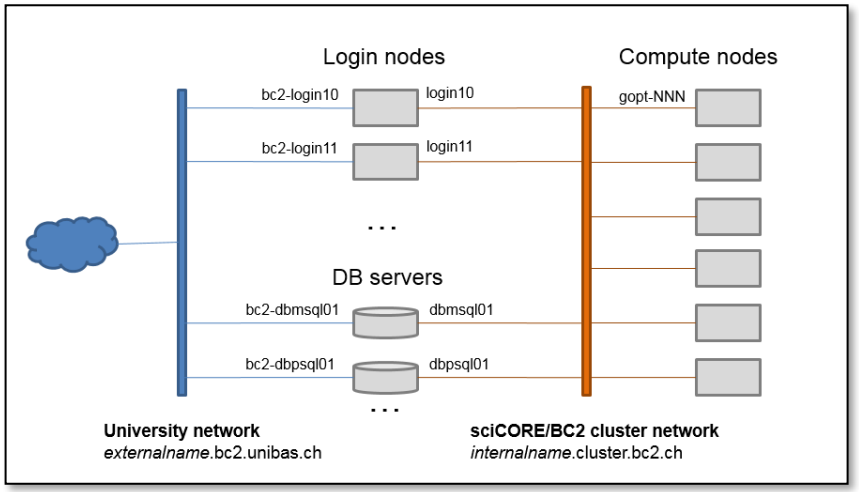
375 computation nodes:
- 8000 cores
- 57 TB RAM (distributed) / 2 node (2 TB) !
- Infiniband interconnected

High-performance storage (GPFS):
- 11 PB
- Hierarchical Storage Management (HSM)

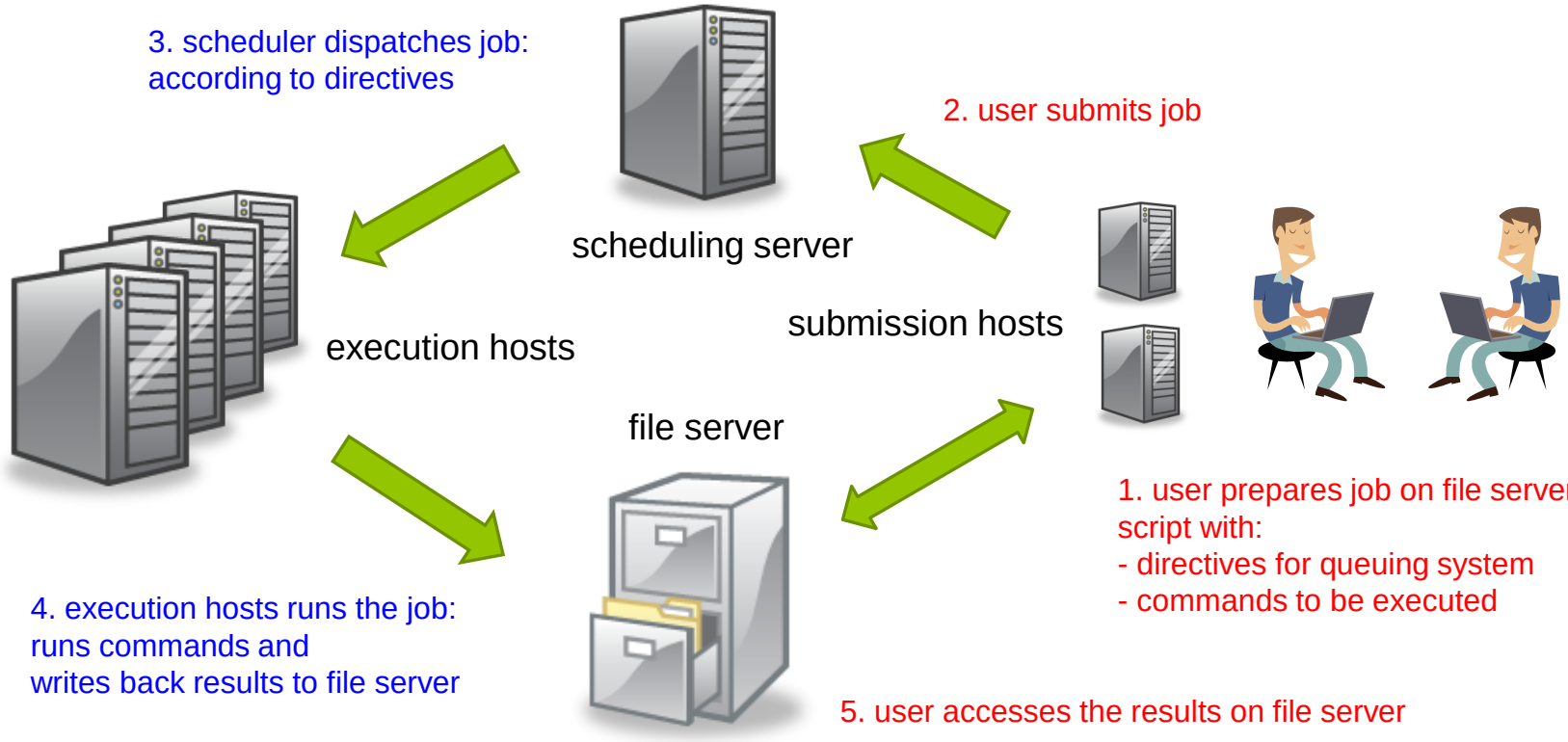
Database and web servers

Private and public services

| Purpose | Uni network external name | sciCORE network internalname | Login |
|---|---|---|---|
| Centos 7 – Login node | login | login10<br>login20 | General SLURM |
| PostgreSQL | bc2-dbpsql01 | dbpsql01 | No login |
| MySQL | bc2-dbmsql01 | dbmspl01 | No login |
| Web Server | bc2.unibas.ch | www | No login |



# Using SLURM




User takes care

Cluster takes care

# Using SLURM

## Script for submitting a job

```
#!/bin/bash
```

This line is mandatory

```
#SBATCH --job-name=myJob
#SBATCH --cpus-per-task=1
#SBATCH --mem=1G
#SBATCH –-time=06:00:00
#SBATCH --qos=6hour
#SBATCH --output=/path/to/stdout/folder
#SBATCH --error=/path/to/stderr/folder
#SBATCH --mail-type=END,FAIL,TIME_LIMIT
#SBATCH --mail-user=<useremail>@unibas.ch
```

All this is optional, but you need to know what is for!

```
# load your required modules
##########################
ml Java/1.8.0_92

# and here goes your command line
sleep 30
```

Your command is mandatory

# Using SLURM

## Script for submitting a job

```
#!/bin/bash

#SBATCH --job-name=myJob
#SBATCH --cpus-per-task=1
#SBATCH --mem=1G
#SBATCH --time=06:00:00
#SBATCH --qos=6hour
#SBATCH --output=/path/to/stdout/folder
#SBATCH --error=/path/to/stderr/folder
#SBATCH --mail-type=END,FAIL,TIME_LIMIT
#SBATCH --mail-user=<useremail>@unibas.ch
```

- Give your job a name that will appear in the queue
- Number of processors (default 1 CPU, max: 64)
- Total RAM. Can also be requested per CPU with --mem-per-cpu (default1 GB/CPU)
- How long will your job run? (default 06:00:00). Scheduler more efficient if this is accurate!
- Select queue to run
- stdout and stderr (default $HOME)
- Send an email when the specified events occur. Caution!
- Define email

```
# load your required modules
##########################
ml Java/1.8.0_92

# and here goes your command line
sleep 30
```

- Load required software
- Execute the program

OMP_NUM_THREADS=$SLURM_CPUS_PER_TASK
OPENBLAS_NUM_THREADS=$SLURM_CPUS_PER_TASK

| Queue name | Max. runtime | Limit TOTAL (cores) | Limit ACCOUNT (cores) | Limit USER (cores) | RAM |
|---|---|---|---|---|---|
| 30min.q | 30 min | 1200 | 1000 | 400 | 256 GB/node |
| 6hours.q | 6 h | 1300 | 800 | 300 | |
| 1d | | | | | |
| 1week.q | 7 days | 400 | 60 | 40 | |
| 2weeks.q | 14 days | 300 | 60 | 40 | |
| infinite.q | ∞ | 100 | 40 | 20 | |

Use command **usage** to know the current status

# Using SLURM

## Script for submitting a job

```
#!/bin/bash

#SBATCH --job-name=myJob
#SBATCH --cpus-per-task=1
#SBATCH --mem=1G
#SBATCH –-time=06:00:00
#SBATCH --qos=6hour
#SBATCH --output=/path/to/stdout/folder
#SBATCH --error=/path/to/stderr/folder
#SBATCH --mail-type=END,FAIL,TIME_LIMIT
#SBATCH --mail-user=<useremail>@unibas.ch

# load your required modules
##########################
ml Java/1.8.0_92

# and here goes your command line
sleep 30
```

- Give your job a name that will appear in the queue
- Number of processors (default 1 CPU, max: 64)
- Total RAM. Can also be requested per CPU with --mem-per-cpu (default1 GB/CPU)
- How long will your job run? (default 06:00:00). Scheduler more efficient if this is accurate!
- Select queue to run
- stdout and stderr (default $HOME)
- Send an email when the specified events occur. Caution!
- Define email
- Load required software
- Execute the program

```
OMP_NUM_THREADS=$SLURM_CPUS_PER_TASK
OPENBLAS_NUM_THREADS=$SLURM_CPUS_PER_TASK
```

## Working with SLURM

| | |
|---|---|
| **sbatch launch.sh** | Sends the script to the queue system, that will schedule the job and assign the required resources. |
| **squeue** | Shows the status of the jobs in the queue. |
| **squeue –j <job ID>** | Shows detailed data about a specific job. |
| **scancel <job ID>** | Cancels a submitted job. |
| **scancel –u <username>** | Cancels all submitted jobs owned by **<username>** |
| **sacct –j <job ID>** | Report and account of usage. **Ex: sacct –j 25120 –o JobID,AllocCPUS,MaxRSS,State,ExitCode** Useful for benchmarking memory needs. |
| **seff <job ID>** | Report and account of usage more user friendly. Info given in email by scheduler. |

# Using SLURM

## More about modules

Usually we need additional software to program / compile / debug / launch / process…
This is handled with a software stack that offers a list of programs/libraries that can be loaded if needed.

```
ABySS/1.3.6-goolf-1.4.10-Python-2.7.5
ABySS/1.9.0-goolf-1.4.10-Python-2.7.5                (D)
ALLPATHS-LG/49835-goolf-1.4.10_20140514
AMOS/3.1.0-goolf-1.4.10
APBS/1.4-linux-static-x86_64
ATSAS/2.6.0-1.el6.x86_64
Augustus/3.0.2-goolf-1.4.10
AutoDock Vina/1.1.2_linux x86
AutoDockSuite/4.2.5.1-goolf-1.4.10
BamTools/2.2.3-goolf-1.4.10
BayesTraits/1.0-linux32
BayesTraits/2.0-Beta-Linux64                         (D)
BCFtools/1.1-goolf-1.4.10
Beast/1.8.0
Beast/1.8.2
Beast/2.1.3                                          (D)
BEDOPS/2.4.1-GCC-4.8.2
BEDTools/2.18.1-goolf-1.4.10
Biopython/1.65-goolf-1.4.10-Python-2.7.5
BitSeq/0.7.0-goolf-1.4.10
BLAST/2.2.16
BLAST/2.2.23
BLAST/2.2.26                                         (D)
BLAST+/2.2.28-goolf-1.4.10
BLAST+/2.2.29-goolf-1.4.10
BLAST+/2.2.30-goolf-1.4.10                           (D)
Bowtie/1.0.0-goolf-1.4.10
Bowtie2/2.0.5-goolf-1.4.10
Bowtie2/2.0.6-goolf-1.4.10
Bowtie2/2.1.0-goolf-1.4.10
Bowtie2/2.2.0-goolf-1.4.10
Bowtie2/2.2.2-goolf-1.4.10                           (D)
BWA/0.6.2-goolf-1.4.10
BWA/0.7.5a-goolf-1.4.10
BWA/0.7.7-goolf-1.4.10
BWA/0.7.8-goolf-1.4.10
BWA/0.7.10-goolf-1.4.10
BWA/0.7.12-goolf-1.4.10                              (D)
CAD-score/1633_86b5fbe9aa36-GCC-4.7.2
CAP3/20071221-pgcc
CCP4/6.4.0.1-ARP-wARP-7.4
CCP4/6.5-ARP-wARP-7.5                                (D)
CD-HIT/4.6.1-goolf-1.4.10-2012-08-27
cdbfasta/0.99-goolf-1.4.10
Celera Assembler/8.1-goolf-1.4.10
CellProfiler/2.1.1-goolf-1.4.10-Python-2.7.5
CEM/0.9.1-goolf-1.4.10
Chimera/1.10-linux x86_64
ClonalFrame/1.2-goolf-1.4.10
Clustal-Omega/1.2.0-goolf-1.4.10
ClustalW2/2.1-goolf-1.4.10
CONTRAfold/2.02-goolf-1.4.10
CONTRAlign/2.01-rna-goolf-1.4.10
CONTRAlign/2.01-proteins-goolf-1.4.10                (D)
Coot/0.8.1-binary-Linux-x86_64-rhel-6-python-gtk2
Cufflinks/2.0.2-goolf-1.4.10
Cufflinks/2.1.1-goolf-1.4.10                         (D)
cutadapt/1.3-goolf-1.4.10-Python-2.7.5
cutadapt/1.4.1-goolf-1.4.10-Python-2.7.5             (D)
Desmond Maestro/2014.2_Linux-x86_64
dssp/2.2.1-goolf-1.4.10
dssp/2003-May-30-goolf-1.4.10                        (D)
DSX/0.89_Linux-x86_64
EMAN2/2.11-linux64
EMBOSS/6.2.0-goolf-1.4.10
EMBOSS/6.6.0-goolf-1.4.10                            (D)
Ensembl-API/20150615-Perl-5.16.3-goolf-1.4.10
ExaML/2.0.4-goolf-1.4.10-mpi-avx
Exonerate/2.2.0-goolf-1.4.10
eXpress/1.5.1-goolf-1.4.10
FASTA/36.3.5e-goolf-1.4.10
FastQC/0.11.3-Java-1.7.0_21
FastTree/2.1.7-goolf-1.4.10
FASTX-Toolkit/0.0.14-goolf-1.4.10
freebayes/0.9.18-1-g4233a23-dirty-goolf-1.4.10
FreeSurfer/5.3.0-centos6_x86_64
FSA/1.15.8-goolf-1.4.10
FSL/5.0.6-centos6_64
GMAP-GSNAP/2014-01-21-goolf-1.4.10
GMAP-GSNAP/2014-06-10-goolf-1.4.10
GMAP-GSNAP/2014-12-21-GCC-4.9.2
GMAP-GSNAP/2014-12-21-goolf-1.4.10                   (D)
GROMACS/4.6.5-goolf-1.4.10-hybrid
GROMACS/4.6.5-goolf-1.4.10-mt                        (D)
GTOOL/0.7.5_x86_64_dynamic
GTOOL/0.7.5_x86_64_static                            (D)
HBPLUS/3.0-GCC-4.7.2
HH-suite/2.0.16-goolf-1.4.10
HMMER/3.1b1-goolf-1.4.10
HOLE/2.2_002
HotspotsX/0.61_Linux-x86_64
HTSeq/0.5.4p5-goolf-1.4.10-Python-2.7.5
HTSeq/0.6.1-goolf-1.4.10-Python-2.7.5                (D)
HTSlib/1.1-goolf-1.4.10
IGV/2.3.59-Java-1.7.0_21
IMPUTE2/2.3.1_x86_64_dynamic
IMPUTE2/2.3.1_x86_64_static                          (D)
IPMF/v4
IQSeq/1.0-goolf-1.4.10_5d2B2d
IsoInfer/0.9.1-goolf-1.4.10
IsoLasso/2.6.1-goolf-1.4.10
khmer/1.1-goolf-1.4.10-Python-2.7.5
KING/1.4_Linux-x86_64
KvarQ/0.12.2-goolf-1.4.10-Python-2.7.5
LGA/v10-2009
libgtextutils/0.6.1-goolf-1.4.10
MACS/1.4.2-1-goolf-1.4.10-Python-2.7.5
MAFFT/7.130-goolf-1.4.10-with-extensions
MAMMOTH/1.0
Mauve/2.3.1
MEME/4.8.0-goolf-1.4.10
MGLTools/1.5.6
Minia/1.608B-goolf-1.4.10
Minimac/2013.7.17-goolf-1.4.10-beta
MM-align/1.0-goolf-1.4.10
MMSEQ/1.0.8-linux64-static
Modeller/9.13-Python-2.7.5
Modeller/9.14-goolf-1.4.10-Python-2.7.5              (D)
MOE/2013 0802
MOE/2014 0901                                        (D)
MolScript/2.1.2-goolf-1.4.10
MotEvo/1.02-goolf-1.4.10
mpiBLAST/1.6.0-goolf-1.4.10
MrBayes/3.2.2-goolf-1.4.10-mpi
MrBayes/3.2.2-ictce-6.2.5-mpi
MrBayes/3.2.2-ictce-6.2.5-serial                     (D)
ms/20071014-goolf-1.4.10
msms/2.6.1-linux-x86_64
MUMmer/3.23-goolf-1.4.10
MUSCLE/3.8.31-goolf-1.4.10
MView/1.49-goolf-1.4.10-Perl-5.16.3
MView/1.57-goolf-1.4.10-Perl-5.16.3                  (D)
Naccess/2.1.1
NAMD/2.9_Linux-x86_64-multicore
NNScore/2.01
NVBIO/0.9.7-gcccuda-2.6.10
OpenMM/6.1-Linux64
OpenStructure/1.4.0-goolf-1.4.10_20141024
OpenStructure/1.4.0-goolf-1.4.10_20150123
OpenStructure/1.4.0-goolf-1.4.10_20150306 (D)
PAML/4.8-goolf-1.4.10
PAML/4.8-ictce-6.2.5                                 (D)
PeakSeq/1.3-goolf-1.4.10
PhyloGibbs/1.2-goolf-1.4.10
PhyML/20120412-goolf-1.4.10
PhyML/20131016-goolf-1.4.10devel                     (D)
picard/1.107
picard/1.135                                         (D)
Pindel/0.2.5a7-goolf-1.4.10
PLINK/1.07-goolf-1.4.10
PRANK/140110-goolf-1.4.10
PROCSE/2.0-goolf-1.4.10
ProModII/3.70
PSIPRED/2.61-goolf-1.4.10-BLAST-2.2.26
PSIPRED/3.5-goolf-1.4.10-BLAST-2.2.26                (D)
pyRAD/3.0.5-goolf-1.4.10-Python-2.7.5
QMEAN/2.1-goolf-1.4.10_ost_20150306
QMEAN/2.1-goolf-1.4.10_20141201                      (D)
RAD/06-2014-goolf-1.4.10
RADtools/1.2.4
Raster3D/3.0-2-goolf-1.4.10
RaxML/8.1.1-goolf-1.4.10-mpi-avx
RaxML/8.1.1-goolf-1.4.10-mt-avx                      (D)
ResMap/1.1.4-linux64
RNASkim/aa7fad3-goolf-1.4.10-20140630
RNASkim/cc0441c-goolf-1.4.10-20140630                (D)
Rosetta/3.5-goolf-1.4.10
rSeq/0.2.0-goolf-1.4.10
Sailfish/0.6.3-goolf-1.4.10-beta
SAMtools/0.1.18-goolf-1.4.10
SAMtools/0.1.19-goolf-1.4.10                         (D)
SCANMS/2.02-goolf-1.4.10-Perl-5.16.3
Schrodinger/2014-1_Linux-x86_64
Schrodinger/2014-2_Linux-x86_64
Schrodinger/2014-3_Linux-x86_64
Schrodinger/2015-1_Linux-x86_64                      (D)
segemehl/0.1.7-goolf-1.4.10
SHAPEIT/v2.r790.RHEL5_5.4.dynamic
SHAPEIT/v2.r790.RHEL5_5.4.static                     (D)
SHELX/2014.2_linux64
sickle/1.210-goolf-1.4.10-bab15f7
smalt/0.7.5-goolf-1.4.10
SOAPaligner/2.21_Linux-x86_64
splitRef/0.0.1
splitRef/0.0.2
SRA-Toolkit/2.3.5-centos_linux64
SSpro/4.03
Stacks/1.18-goolf-1.4.10
Stacks/1.19-goolf-1.4.10
Stacks/1.20-goolf-1.4.10
Stacks/1.27-goolf-1.4.10
Stacks/1.32-goolf-1.4.10                             (D)
Stampy/1.0.23-goolf-1.4.10
SwitchSeq/1.0-goolf-1.4.10
tabix/0.2.6-goolf-1.4.10
TopHat/2.0.10-goolf-1.4.10
Trimmomatic/0.32-Java-1.7.0_21
Trinity/2.0.4-goolf-1.4.10
USEARCH/7.0.1090_i86linux32
VCFtools/0.1.12-goolf-1.4.10-Perl-5.16.3
Velvet/1.2.10-goolf-1.4.10
ViennaRNA/2.1.8-goolf-1.4.10
VIGRA/1.10.0-goolf-1.4.10
VMD/1.9.1-GCC-4.7.2
VSEARCH/1.1.3
XDS/3Nov2014-INTEL64_Linux_x86_64
Xplor-NIH/2.39
------------------------------------------ /scicore/soft/modules/chem ------------------------------------------
CHARMM/36b1-goolf-1.4.10-mpi        gaussian/g09-d.01       OpenBabel/2.3.2-goolf-1.4.10
CHARMM/39a1-goolf-1.4.10-mpi (D)    Molcas/v8.0.15-05-06    vasp/5.3-intel-2015.02
------------------------------------------ /scicore/soft/modules/compiler ------------------------------------------
Clang/3.3-GCC-4.8.2     GCC/4.9.2                (D)    icc/2015.2.164-GCC-4.9.2   (D)    ifort/2015.2.164-GCC-4.9.2 (D)
GCC/4.7.2               icc/2013_sp1.2.144              ifort/2013_sp1.2.144
GCC/4.8.2               icc/2015.1.133-GCC-4.9.2        ifort/2015.1.133-GCC-4.9.2
GCC/4.8.4               icc/2015.1.133                  ifort/2015.1.133
------------------------------------------ /scicore/soft/modules/devel ------------------------------------------
ant/1.9.0-Java-1.7.0_21                 Boost/1.53.0-goolf-1.4.10-Python-2.7.5      JUnit/4.11-Java-1.7.0_21
argtable/2.13-goolf-1.4.10              Boost/1.53.0-goolf-1.4.10                (D)libharu/2.2.0-goolf-1.4.10
Autoconf/2.69-GCC-4.8.4                 Check/0.9.12-goolf-1.4.10                   M4/1.4.17-GCC-4.8.4
Automake/1.15-GCC-4.8.4                 CMake/2.8.4-GCC-4.7.2                       protobuf/2.5.0-goolf-1.4.10
Boost/1.47.0-goolf-1.4.10-Python-2.7.5  CMake/2.8.4-goolf-1.4.10                   QCachegrind/0.7.4-goolf-1.4.10
Boost/1.47.0-goolf-1.4.10               CMake/2.8.11-GCC-4.8.2                      Qt/4.8.4-goolf-1.4.10
Boost/1.49.0-goolf-1.4.10-Python-2.7.5  CMake/2.8.12-GCC-4.7.2                      QuadProg++/1.2.1-goolf-1.4.10
Boost/1.51.0-goolf-1.4.10-Python-2.7.5  CMake/2.8.12-goolf-1.4.10                (D)SCons/2.3.0-goolf-1.4.10-Python-2.7.5
Boost/1.51.0-goolf-1.4.10               Intel-TBB/4.2-update3-goolf-1.4.10          Valgrind/3.8.1-goolf-1.4.10
------------------------------------------ /scicore/soft/modules/lang ------------------------------------------
Java/1.7.0_21      NASM/2.11.06-goolf-1.4.10    Python/2.7.5-goolf-1.4.10       R/3.1.0-goolf-1.4.10
Java/1.7.0_75 (D)  Perl/5.16.3-goolf-1.4.10     Python/2.7.5-ictce-6.2.5  (D)   rpy2/2.4.4-goolf-1.4.10-Python-2.7.5
------------------------------------------ /scicore/soft/modules/lib ------------------------------------------
GDAL/1.9.2-ictce-6.2.5     google-glog/0.3.3-goolf-1.4.10     libtool/2.4.5-GCC-4.8.4     Xerces-C/3.1.2-intel-2015a
gflags/2.0-goolf-1.4.10    google-sparsehash/2.0.2-goolf-1.4.10    libunwind/1.1-goolf-1.4.10    zlib/1.2.8-goolf-1.4.10
------------------------------------------ /scicore/soft/modules/math ------------------------------------------
Eigen/3.1.4-ictce-6.2.5        Mathematica/10.0.2                      numpy/1.8.2-goolf-1.4.10-Python-2.7.5
Gurobi/5.6.3 linux64           MATLAB/r2014a                           numpy/1.9.1-goolf-1.4.10-Python-2.7.5  (D)
Gurobi/6.0.0 linux64    (D)    MCL/14.137-goolf-1.4.10                 scipy/0.14.0-goolf-1.4.10-Python-2.7.5
JAGS/3.4.0-goolf-1.4.10        numpy/1.7.1-goolf-1.4.10-Python-2.7.5
------------------------------------------ /scicore/soft/modules/mpi ------------------------------------------
impi/4.1.3.049                          impi/5.0.2.044-iccifort-2015.1.133          OpenMPI/1.6.4-GCC-4.7.2
impi/5.0.2.044-iccifort-2015.1.133-GCC-4.9.2    impi/5.0.3.048-iccifort-2015.2.164-GCC-4.9.2 (D)    OpenMPI/1.8.4-GCC-4.8.4 (D)
------------------------------------------ /scicore/soft/modules/numlib ------------------------------------------
Armadillo/2.4.4-goolf-1.4.10-Python-2.7.5       GSL/1.12-goolf-1.4.10
ATLAS/3.10.1-goolf-1.4.10-LAPACK-3.4.2          GSL/1.16-goolf-1.4.10                       (D)
beagle-lib/20140228-goolf-1.4.10-r1248-no-gpu   imkl/11.1.2.144
beagle-lib/20140320-ictce-6.2.5-r1253      (D)  imkl/11.2.1.133-iimpi-7.2.3-GCC-4.9.2
CGAL/4.0-goolf-1.4.10-Python-2.7.5              imkl/11.2.1.133-iimpi-7.2.3
Eigen/2.0.17-goolf-1.4.10                       imkl/11.2.2.164-iimpi-7.2.5-GCC-4.9.2       (D)
------------------------------------------ /scicore/soft/modules/math ------------------------------------------
Eigen/3.1.4-ictce-6.2.5        Mathematica/10.0.2                      numpy/1.8.2-goolf-1.4.10-Python-2.7.5
Gurobi/5.6.3 linux64           MATLAB/r2014a                           numpy/1.9.1-goolf-1.4.10-Python-2.7.5  (D)
Gurobi/6.0.0 linux64    (D)    MCL/14.137-goolf-1.4.10                 scipy/0.14.0-goolf-1.4.10-Python-2.7.5
JAGS/3.4.0-goolf-1.4.10        numpy/1.7.1-goolf-1.4.10-Python-2.7.5
------------------------------------------ /scicore/soft/modules/mpi ------------------------------------------
impi/4.1.3.049                          impi/5.0.2.044-iccifort-2015.1.133          OpenMPI/1.6.4-GCC-4.7.2
impi/5.0.2.044-iccifort-2015.1.133-GCC-4.9.2    impi/5.0.3.048-iccifort-2015.2.164-GCC-4.9.2 (D)    OpenMPI/1.8.4-GCC-4.8.4 (D)
------------------------------------------ /scicore/soft/modules/numlib ------------------------------------------
Armadillo/2.4.4-goolf-1.4.10-Python-2.7.5       GSL/1.12-goolf-1.4.10
ATLAS/3.10.1-goolf-1.4.10-LAPACK-3.4.2          GSL/1.16-goolf-1.4.10                       (D)
beagle-lib/20140228-goolf-1.4.10-r1248-no-gpu   imkl/11.1.2.144
beagle-lib/20140320-ictce-6.2.5-r1253      (D)  imkl/11.2.1.133-iimpi-7.2.3-GCC-4.9.2
CGAL/4.8-goolf-1.4.10-Python-2.7.5              imkl/11.2.1.133-iimpi-7.2.3
Eigen/2.0.17-goolf-1.4.10                       imkl/11.2.2.164-iimpi-7.2.5-GCC-4.9.2       (D)
Eigen/3.1.1-goolf-1.4.10                        MPFR/3.1.8-goolf-1.4.10
Eigen/3.1.4-goolf-1.4.10                        OpenBLAS/0.2.6-gompi-1.4.10-LAPACK-3.4.2
Eigen/3.2.1-goolf-1.4.10                   (D)  OpenBLAS/0.2.13-GCC-4.8.4-LAPACK-3.5.0      (D)
FFTW/3.3.3-gompi-1.4.10                         ScaLAPACK/2.0.2-gompi-1.4.10-OpenBLAS-0.2.6-LAPACK-3.4.2
FFTW/3.4-gompi-1.7.20                      (D)  ScaLAPACK/2.0.2-gompi-1.7.20-OpenBLAS-0.2.13-LAPACK-3.5.0 (D)
GMP/5.0.5-goolf-1.4.10
------------------------------------------ /scicore/soft/modules/system ------------------------------------------
CUDA/5.5.22    hwloc/1.6.2-GCC-4.7.2    hwloc/1.8.1-iccifort-2013_sp1.2.144    hwloc/1.10.1-GCC-4.8.4 (D)
------------------------------------------ /scicore/soft/modules/toolchain ------------------------------------------
gcccuda/2.6.10      goolf/1.4.10                iccifort/2015.1.133                 impi/7.2.3-GCC-4.9.2        intel/2014.11
gompi/1.4.10        iccifort/2013_sp1.2.144     iccifort/2015.2.164-GCC-4.9.2 (D)   impi/7.2.3                  intel/2015a
gompi/1.7.20   (D)  iccifort/2015.1.133-GCC-4.9.2  ictce/6.2.5                     impi/7.2.5-GCC-4.9.2 (D)    intel/2015.02 (D)
------------------------------------------ /scicore/soft/modules/tools ------------------------------------------
Coreutils/8.22-goolf-1.4.10         git/1.8.5.6-goolf-1.4.10                (D)    Mercurial/2.5.2-goolf-1.4.10-Python-2.7.5
cURL/7.29.0-goolf-1.4.10            GLPK/4.53-goolf-1.4.10                         numactl/2.0.10-GCC-4.8.4
cURL/7.40.0                         gperftools/2.1-goolf-1.4.10                    pigz/2.3.1-goolf-1.4.10
cURL/7.43.0-intel-2015a        (D)  Grace/5.1.23-goolf-1.4.10                      subversion/1.7.19-goolf-1.4.10
EasyBuild/2.2.0                     HPL/2.1-intel-2015.02                          Szip/2.1-goolf-1.4.10
EasyBuild/2.3.0                (D)  IOR/2.10.3-goolf-1.4.10-posix-and-mpiio        Vim/7.4-goolf-1.4.10
git/1.8.3.1-goolf-1.4.10            mdtest/1.9.3-goolf-1.4.10
------------------------------------------ /scicore/soft/modules/vis ------------------------------------------
frealign/9.10-intel-2015a-avx-mp    freetype/2.4.11-goolf-1.4.10    Xmipp/3.1-goolf-1.4.10
```

## More about modules

Usually we need additional software to program / compile / debug / launch / process…
This is handled with a software stack that offers a list of programs/libraries that can be loaded if needed.
We control what is loaded via the command ml

| | |
|---|---|
| ml | :shows the list of loaded modules. |
| ml av | :(available) shows the list of all modules. |
| ml av <keyword> | :shows available modules that contain <keyword> in their name. |
| ml spider <keyword> | :shows available modules that contain <keyword> in their name with extra info. |
| ml <module> | :loads <module> |
| ml –<module> | :unloads <module> |
| ml swap <module1> <module2> | :unloads <module> and loads <module2> (but better to do purge!) |
| ml purge | :unloads all modules |
| ml show <module> | :shows detailed information about <module> |

## More about toolchains

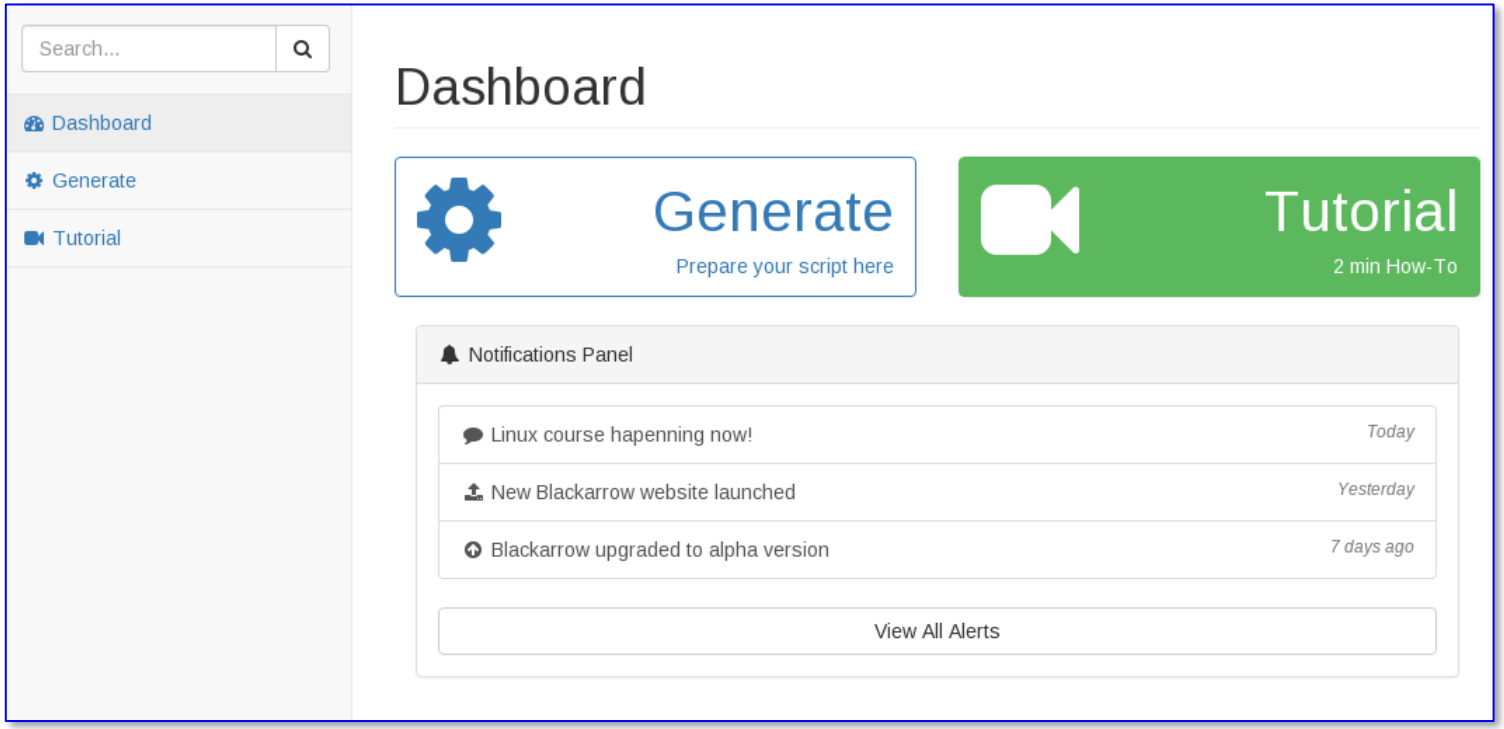
Some software has encoded in its name how it was compiled and with which libraries: toolchains

| | |
|---|---|
| goolf /foss | GNU compiler + OpenMPI + OpenBLAS + FFTW and ScaLAPACK |
| intel | ifort + icc + MKL + intel MPI |
| iomkl | ifort + icc + MKL + OpenMPI |

You can always do **ml show <keyword>** to know more about it.

# Using SLURM

We designed a user-friendly web-based script generator, that will help you to familiarize with launching scripts.




**https://scriptgen.scicore.unibas.ch**

# 4. Exercise

1. Log in to the cluster
2. Enter in linux_course/ , create a directory named Exercise4/ and enter in it.
3. Use scriptgen to write a script named launch.sh that contains the following:

```
#!/bin/bash

#SBATCH --job-name=test.<yourname>
#SBATCH --cpus-per-task=1
#SBATCH --mem-per-cpu=1G
#SBATCH --time=00:03:00
#SBATCH --output=out.o
#SBATCH --error=err.o

time sleep 60
```

4. Save the file and submit it with sbatch launch.sh
5. Check the status with watch squeue –u <username>
6. Check how long was the code running (less err.o)
7. Copy this two files:

   cp /shared/data/linux/200kheger.relax4 .
   cp /shared/data/linux/findneighbors.run .

9. Additionally, you can copy and read this pdf:

   cp /shared/data/linux/findneighbors.pdf .

   to learn a bit more about what does findneighbors.run calculate.
10. Create a script to launch findneighbors.run for 10 minutes with 300Mb and 1 core.

    And add the commands

    ulimit –s unlimited
    export OMP_NUM_THREADS=$SLURM_CPUS_PER_TASK
    time ./findneighbors.run in your script.

    **Tip:** Copy launch.sh to launch2.sh and modify it correspondingly.
11. Check that the code is running and note the job ID.
12. Get extra info with sacct –o ALL –j <job ID>. Try now seff <job ID>.
13. Check how long was the code running.
14. Re-launch the program now with 3 cores changing to --cpus-per-task=3

    How long did it take now? Check that it indeed used 3 cores.

# 4. Exercise

1. Log in to the cluster
2. Enter
3. Use s
4. Save
5. Chec
6. Chec
7. Copy

9. Addi

to le
10. Cre

11. Che
12. Get
13. Che
14. Re-l
Hov

in it.
ing:

<yourname>
1
1G

1 core.

=3

You can check the results of the calculation doing the following:

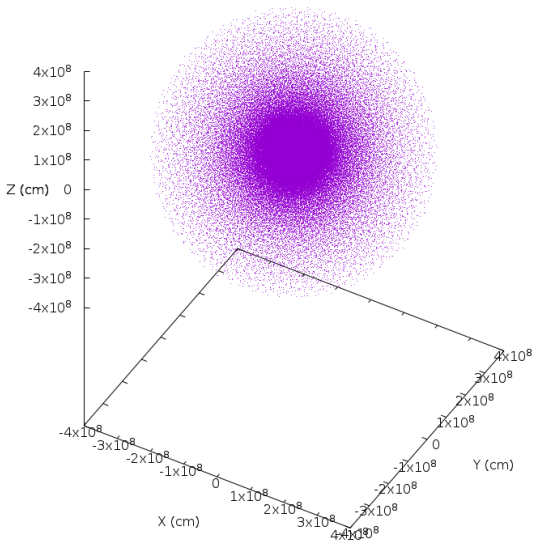
cp /shared/data/linux/plot.gpl .

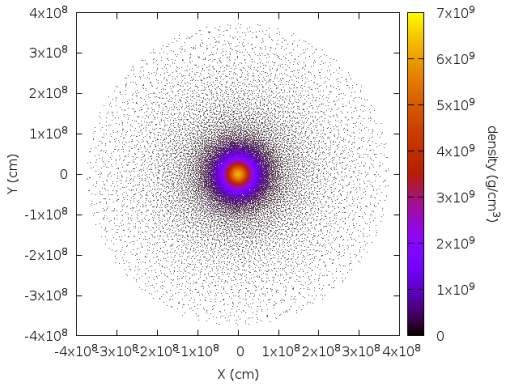
gnuplot plot.gpl

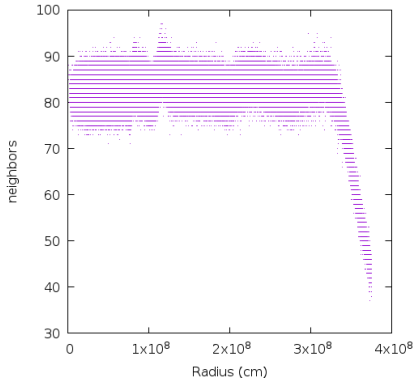
(if you get asked to select a module just select #1)

display results.png

You should see an image similar to this one:



# Using SLURM

## That's fine, but…

What happens when I have to submit the same job with different data sets 1,000 times, or 10,000 times?

You should use array jobs:

- You only write one script
- You don't have to worry about deleting thousands of scripts
- If you submit and array job, and realize that you made a mistake, you only have one jod id to cancel, instead of 100s.
- You put less burden on the head node.

## Script for submitting an array job

```
#!/bin/bash

#SBATCH --job-name=myrun
#SBATCH --cpus-per-task=1
#SBATCH --mem-per-cpu=1G
#SBATCH --time=06:00:00
#SBATCH --qos=6hours
#SBATCH --output=myrun.o%A-%3a
#SBATCH --error=myrun.e%A-%3a
#SBATCH --mail-type=END,FAIL,TIME_LIMIT
#SBATCH --mail-user=mailaddress@unibas.ch


# Tell SLURM that this is an array of jobs
#########################
#SBATCH --array=1-100%5

# load your required modules
#######################
ml Java

# and here goes your command line
/bin/sleep 30
```

This how the output files can be formatted. Here %A stands for the job array's master ID and %a for the job array index. The 3 is for using 3 digits:

myrun.o1234-000, myrun.o1234-001, …

This will launch 100 tasks to be numbered from 1 to 100.

Optionally, we can limit the amount of tasks running simultaneously.

When a task in the array job is sent to a compute node, its task number is stored in the variable SLURM_ARRAY_TASK_ID, so we can use it to select the input and output commands/data that we want.

# 5. Exercise

```
#!/bin/bash

#SBATCH --job-name=myrun
#SBATCH --cpus-per-task=1
#SBATCH --mem-per-cpu=1G
#SBATCH --time=06:00:00
#SBATCH --output=myrun.o%A-%3a
#SBATCH --error=myrun.e%A-%3a

# Tell SLURM that this is an array of jobs
#######################
#SBATCH --array=1-100%10

# and here goes your command line
sleep 30
```

This is just an example to remind you the options of a script.

1. Log in to the cluster
2. Enter in linux_course/ and create a directory named Exercise5/ and enter in it.
3. Create a file named commands.cmd that contains 20 lines with the command sleep N, with N taking values between 5 and 20.
4. Create a script that launches an array job of 20 jobs in chunks of 5 simultaneous jobs.
5. The script should read the commands from the file commands.cmd and submit them in the way specified above.
6. When done submit the script and monitorize the submission.

TIP: Remember that the variable $SLURM_ARRAY_TASK_ID runs from 1 up the number of jobs in the array.

TIP: $(<command>) executes <command>

# Using SLURM

## Interesting tip:

If you have a pipeline of different processes that depend on the results of the previous step, you can use the option --dependency

## Script for submitting an array of dependant jobs

```
#!/bin/bash

jobID=$(sbatch job1.sh | awk `{print $4}`)
sbatch --dependency=afterok:$jobID job2.sh
```

Get the job ID of the first job of the pipeline.

Hold the second job until $jobID finishes.

---

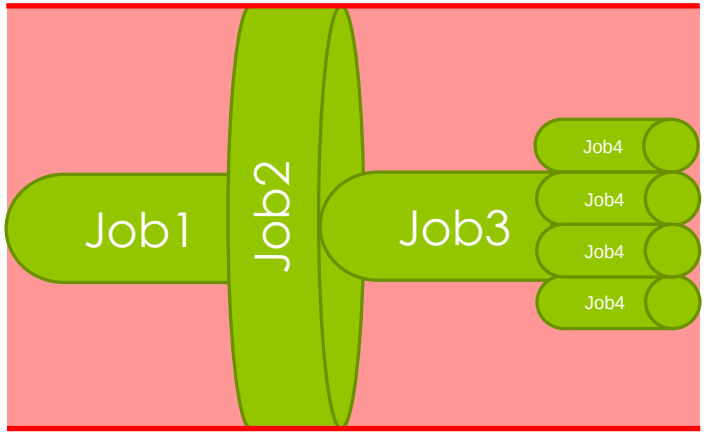
```
#!/bin/bash

sbatch  --job-name=myjob1 job1.sh
sbatch  --dependency=singleton --job-name=myjob1 job2.sh
```

Submit first job.

Hold the second job until myjob1 finishes.

Nevertheless, be cautious with very inhomogeneus requirements at different stages of the pipeline.



## Files processing:

**Best practices for Transferring files to/from/on the cluster (and scripts)**
**Moving large number of files, and/or large files (and scripts)**

For remote transfer, using scp is good enough for most cases:

scp <options> <file_origin> <file_destination>

You can specify a path in a remote machine as: <username>@<host>:<path>/<filename>
Most typical option is –rv (recursive and verbose) to transfer full content of a directory.

Examples:

scp myfile.txt cabezon@login.scicore.unibas.ch:/scicore/home/scicore/cabezon/research/
scp cabezon@login.scicore.unibas.ch:research/myfile.txt .
scp -rv cabezon@login.scicore.unibas.ch:research/folder1/ ruben@lapalma1.iac.es:simulation/

NOTE: scp works over ssh, so you need a working ssh access to the machines involved.

## Files processing:

To transfer many and/or large files, using rsync is the best option:

rsync <options> <file_origin> <file_destination>

You can specify a path in a remote machine in the same way as with scp: <username>@<host>:<path>/<filename>

| | |
|---|---|
| Copy/Sync a File on a Local Computer<br>Copy/Sync a Directory on Local Computer | rsync -zvh backup.tar /tmp/backups/<br>rsync -avzh /home/rpmpkgs /tmp/backups/ |
| Copy a File from a Local Server to a Remote Server with SSH | rsync -avzhe ssh backup.tar ruben@login.scicore.unibas.ch:/backups/ |
| Show Progress While Transferring Data | rsync -avzhe ssh --progress backup.tar ruben@login.scicore.unibas.ch:/backups/ |
| Automatically Delete source Files after successful Transfer | rsync --remove-source-files -zvh backup.tar /tmp/backups/ |
| Do a Dry Run to test the command before doing any changes | rsync --dry-run --remove-source-files -zvh backup.tar /tmp/backups/ |

## Files processing:

To transfer many and/or large files, using rsync is the best option:

rsync <options> <file_origin> <file_destination>

You can specify a path in a remote machine in the same way as with scp: <username>@<host>:<path>/<filename>

| | |
|---|---|
| Copy/Sync a File on a Local C<br>Copy/Sync a Directory on Loca | ups/ |
| Copy a File from a Local Serve<br>Server with SSH | ps/ |
| Show Progress While Transfer | ar<br>ps/ |
| Automatically Delete source Fi<br>successful Transfer | up.tar /tmp/backups/ |
| Do a Dry Run to test the comm<br>any changes | zvh backup.tar |

| Common options | |
|---|---|
| -v | verbose |
| -r | copies data recursively (but don't preserve timestamps and permission while transferring data |
| -a | archive mode allows copying files recursively and it also preserves symbolic links, file permissions, user & group ownerships and timestamps |
| -z | compress file data |
| -h | output numbers in a human-readable format |
| -e | Specify the transfer protocol name you want to use |

# Final messages

## Good praxis!

- Keep your storage low in home/
- Add the string “nobackup” to directories that do not need to be backed up from your home/
- Your programs should not access more than 10s-100s files in a single process.
- When submitting your job, check that it is running as expected with the desired parameters.
- Do not run jobs in the login nodes.
- Read the documentation!

**Thank you for your attention**


Resume at:
14:00


Coffee
break