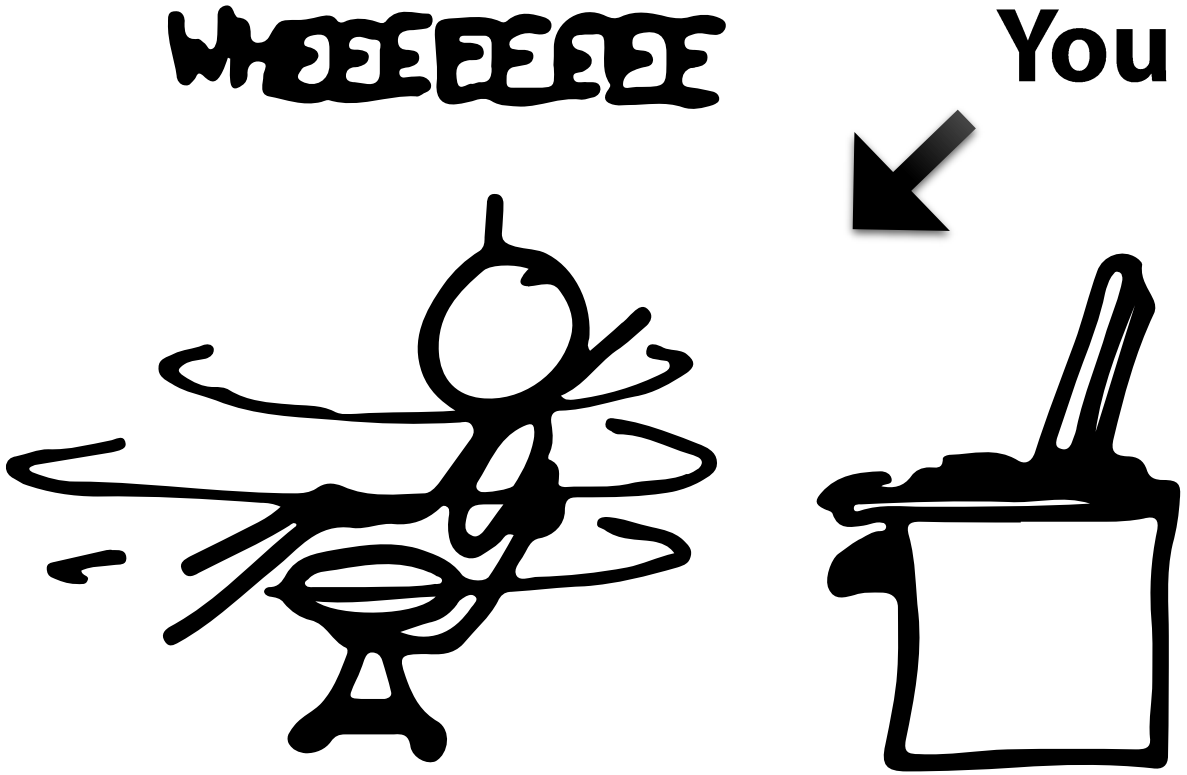
UNIVERSITEIT GENT

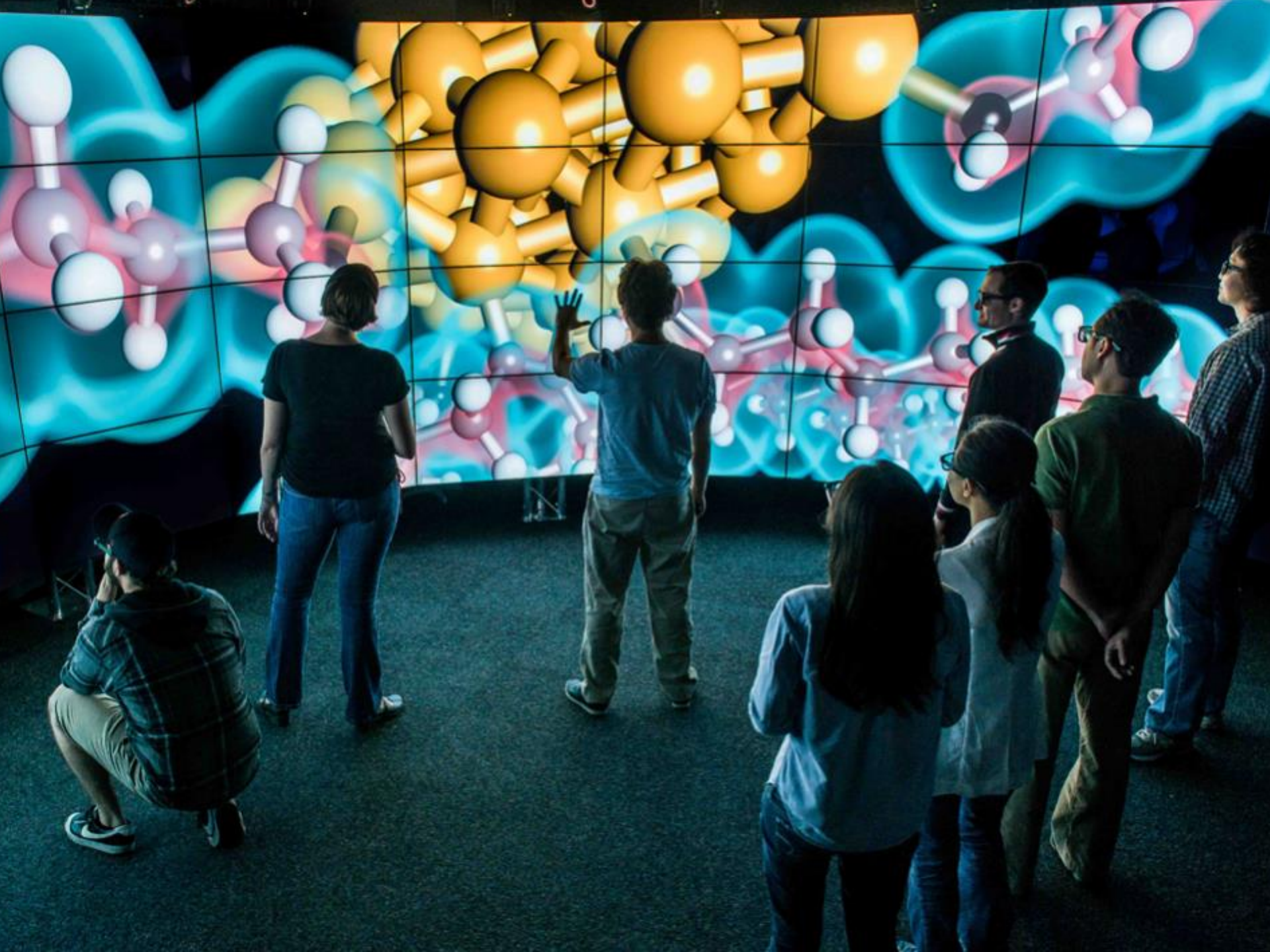
Intro to the dark arts

or supercomputer abuse for beginners

By Michael Sluydts

http://tiny.cc/l478bz


WHEEEEEEEE
You



The price you pay

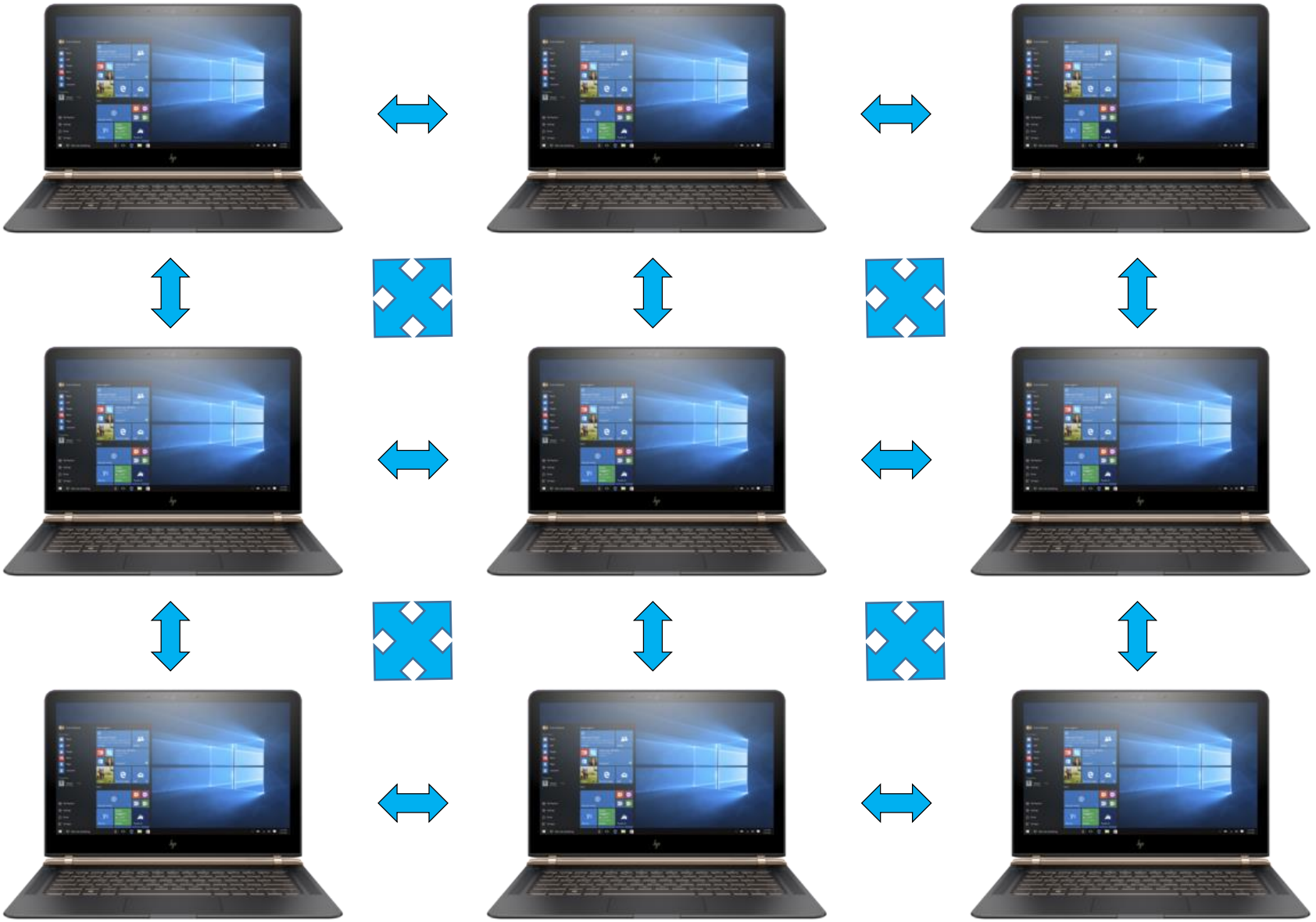
```
vsc40479@gligar01 /user/scratch/expert/#
```

A day in the life of a CMM researcher

1. What is a supercomputer?
2. How do I login?
3. How to feed it commands
4. Managing files
5. Submitting calculations
6. Backing up files

What makes a computer super?

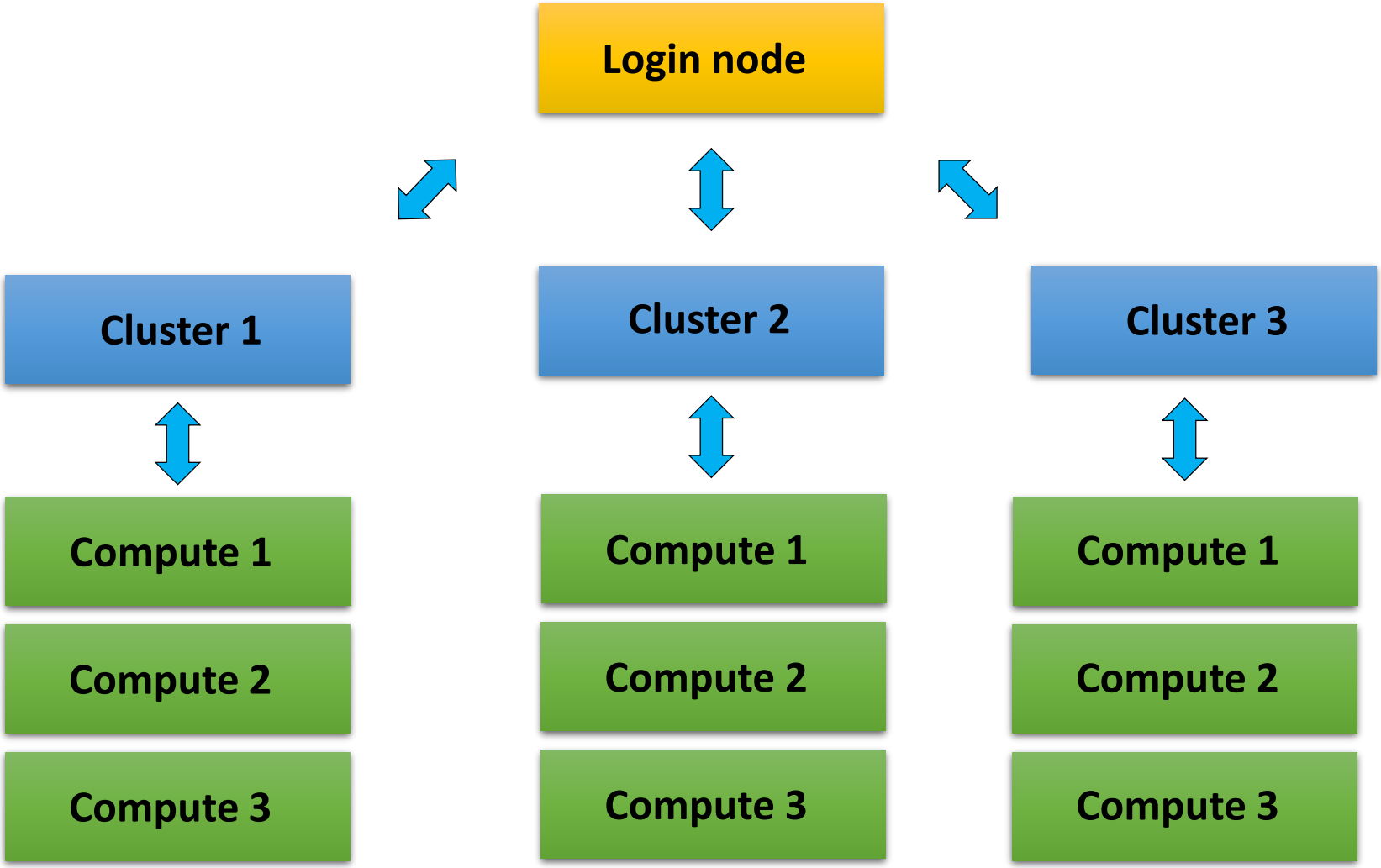
What makes a computer super?



Hurray, *a* supercomputer!

Really 6 clusters, named after…

Practical topology



So how do I login?

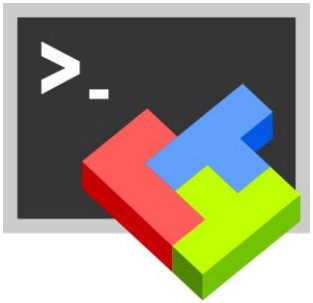
Setting up an account https://www.ugent.be/hpc/en/support/documentation.htm

1. Install software
2. Make key pair
3. Apply for account
4. Get welcome email
5. Configure software
6. Login

SSH Software (Secure shell)

Windows

MobaXTerm



http://mobaxterm.mobatek.net/

NOT PuTTY

Unix

Terminal

Android

JuiceSSH

Generate key pair

Public given to HPC (Upload to account.vscentrum.be)

Private kept locally

Need both keys to login, can add additional keys later

Password optional (noise is pass)

Welcome!

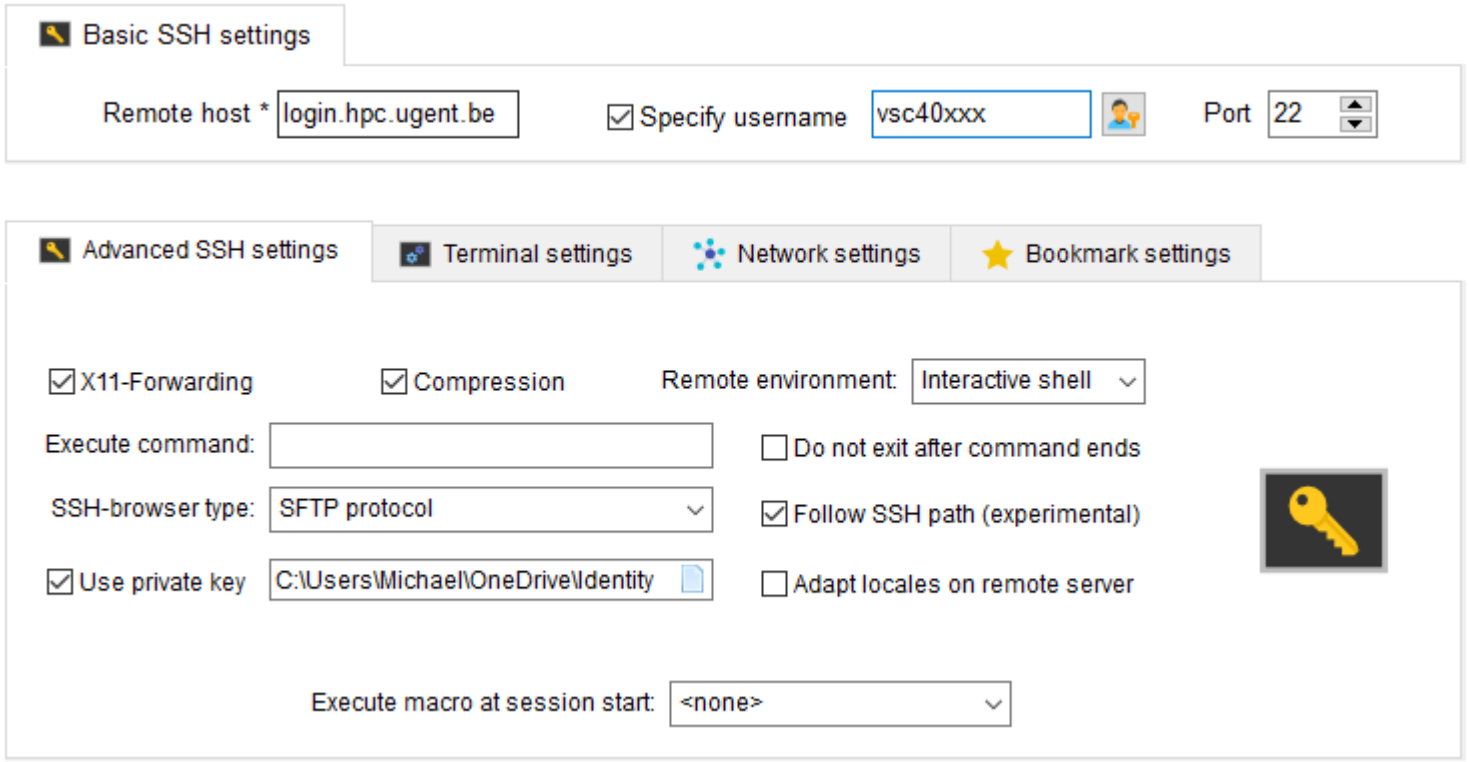
```
Dear (Username),
Your VSC-account has been approved by an administrator.
Your vsc-username is vsc40000

Your account should be fully active within one hour.

To check or update your account information please visit
https://account.vscentrum.be/

For further info please visit https://www.vscentrum.be/en/user-portal

Kind regards,
-- The VSC administrators
```

Congratulations, you are now a number vsc40xxx

If your account has not been used, ask about the installer script when we login.

Exercise 1: Log in

Windows



Unix

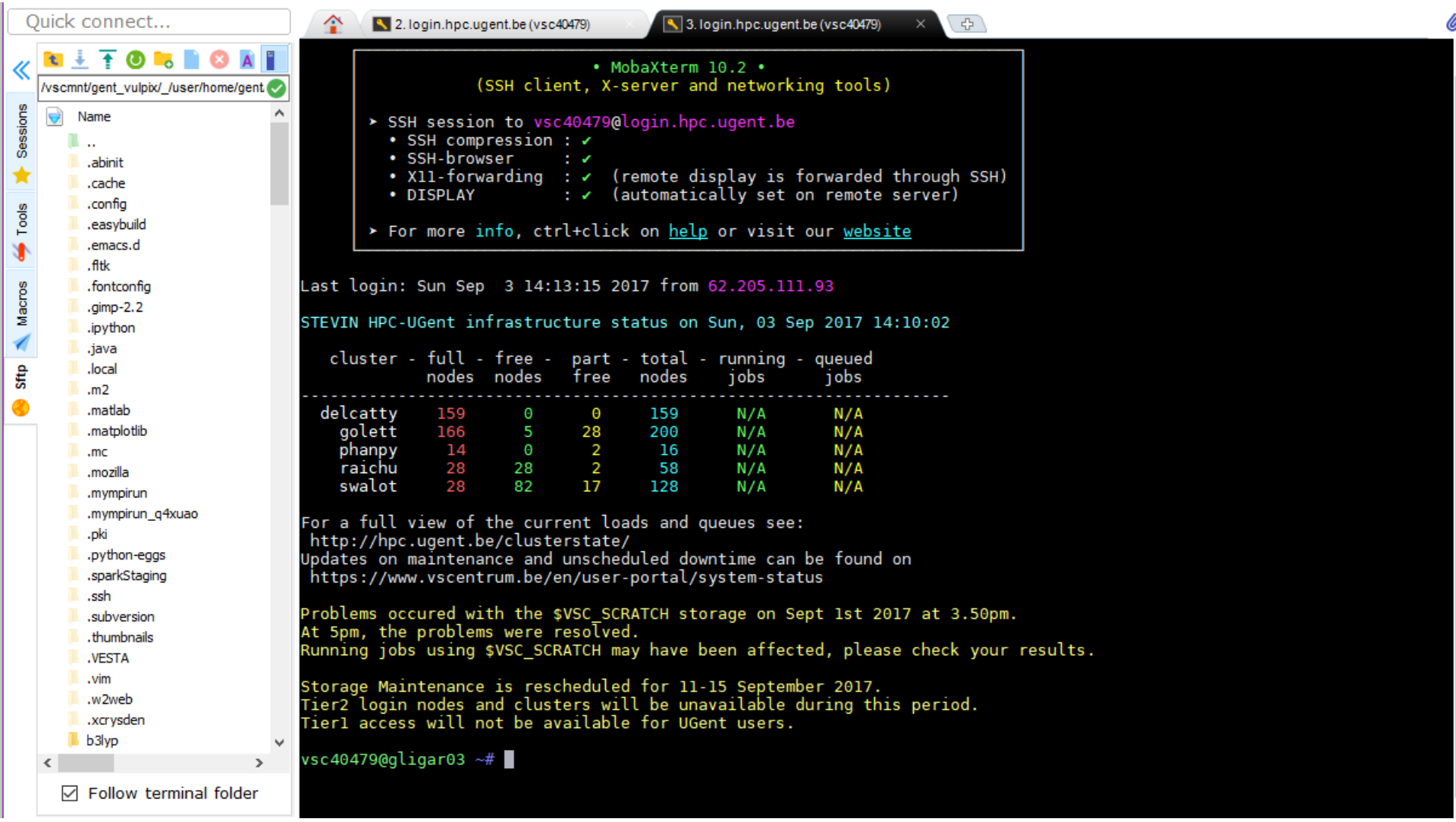
ssh –X vsc40000@login.hpc.ugent.be

Bootstrapping

```
vsc40xxx@gligar0x ~#
```

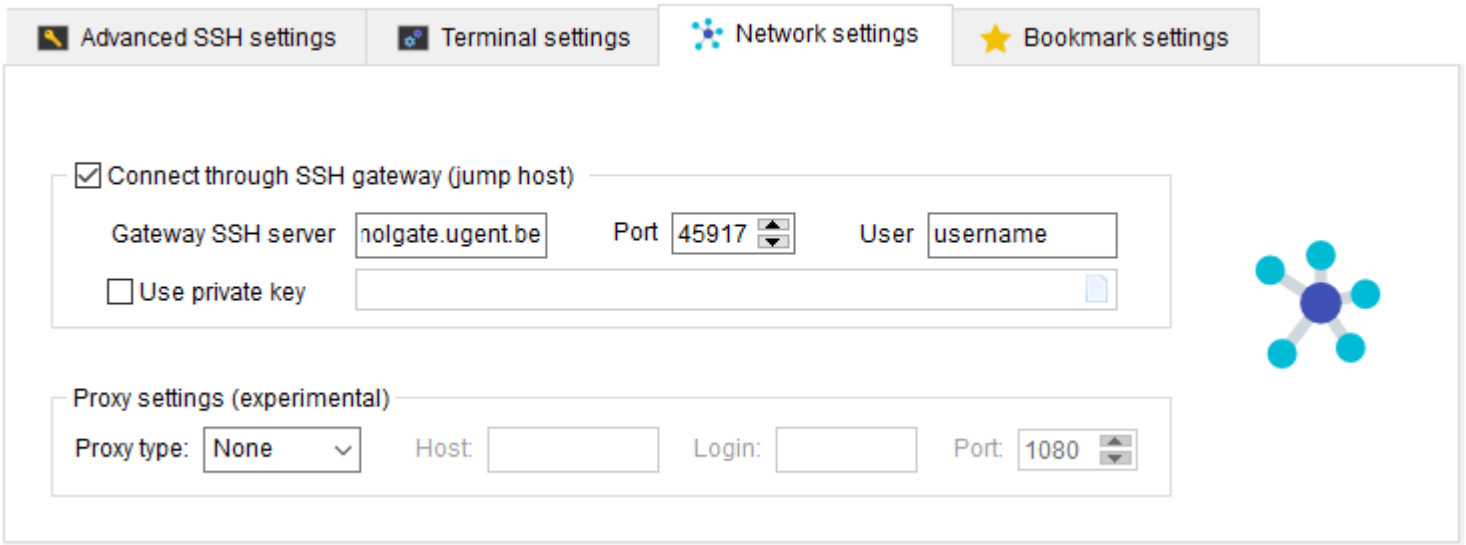
cd /data/gent/vo/000/gvo00003/shared/thesisbootstrap
./bootstrap

Success!


Quick connect...
2. login.hpc.ugent.be (vsc40479)
3. login.hpc.ugent.be (vsc40479)
/vscmnt/gent_vulpix/_/user/home/gent
Name
Sessions
Tools
Macros
Sftp
Follow terminal folder
• MobaXterm 10.2 •
(SSH client, X-server and networking tools)
➤ SSH session to vsc40479@login.hpc.ugent.be
• SSH compression : ✔
• SSH-browser : ✔
• X11-forwarding : ✔ (remote display is forwarded through SSH)
• DISPLAY : ✔ (automatically set on remote server)
➤ For more info, ctrl+click on help or visit our website
Last login: Sun Sep 3 14:13:15 2017 from 62.205.111.93
STEVIN HPC-UGent infrastructure status on Sun, 03 Sep 2017 14:10:02
cluster - full - free - part - total - running - queued
nodes nodes free nodes jobs jobs
delcatty 159 0 0 159 N/A N/A
golett 166 5 28 200 N/A N/A
phanpy 14 0 2 16 N/A N/A
raichu 28 28 2 58 N/A N/A
swalot 28 82 17 128 N/A N/A
For a full view of the current loads and queues see:
http://hpc.ugent.be/clusterstate/
Updates on maintenance and unscheduled downtime can be found on
https://www.vscentrum.be/en/user-portal/system-status
Problems occured with the $VSC_SCRATCH storage on Sept 1st 2017 at 3.50pm.
At 5pm, the problems were resolved.
Running jobs using $VSC_SCRATCH may have been affected, please check your results.
Storage Maintenance is rescheduled for 11-15 September 2017.
Tier2 login nodes and clusters will be unavailable during this period.
Tier1 access will not be available for UGent users.
vsc40479@gligar03 ~#

Outside of the Ugent network?

Windows



or use vpn

Unix

ssh –X username@molgate.ugent.be –p 45917

What next? Commands!

Let's learn to run simple commands:

```
vsc40xxx@gligar0x ~# command input1 input2
```

More complicated programs use flags:

```
vsc40xxx@gligar0x ~# command --flag longform -f shortform
```

Most commands feature a help flag!

- Long form: --help
- Short form: -h
- or by entering the command without flags

These can also be browsed using the man command:

```
vsc40xxx@gligar0x ~# man command
```

Finally shortforms without input can be combined

```
vsc40xxx@gligar0x ~# command -fgh
```

Remember this for when you get lost!

Navigating our files

```
vsc40xxx@gligar0x ~# pwd (print working directory)
```

/user/home/gent/vsc404/vsc40479

Home directory is not for working:

- **HOME: files for login and small scripts**
- **DATA: storing files after calculations**
- **SCRATCH: running actual calculations**

Let's change our working directory to scratch using the change directory command:

```
vsc40xxx@gligar0x ~# cd $VSC_SCRATCH  (a variable for the right path)
vsc40xxx@gligar0x ~# pwd
```

/user/scratch/gent/vsc404/vsc40479

Now to make a directory for our calculation

```
vsc40xxx@gligar0x ~# mkdir calculation1
```

Go in

```
vsc40xxx@gligar0x ~# cd calculation1
```

Go one level back up

```
vsc40xxx@gligar0x ~# cd ../
```

Listing files

```
vsc40xxx@gligar0x ~# ls

imap.py                 perturblog              test.py
inbox                   phononspure1.png        test.sh
INCAR                   pip                     user
intel                   Register                vasp
```

```
vsc40xxx@gligar0x ~# ls -l (or ll)

-rwx------  1 vsc40479 vsc40479      0 Jul 14  2014 output
-rwx------  1 vsc40479 vsc40479 524288 Aug 14  2014 PA2.chk
drwx------  2 vsc40479 vsc40479  32768 Jul  8  2015 pbpe
-rw-rw-r--  1 vsc40479 vsc40479      0 Jun  3  2016 PCDAT
-rwx------  1 vsc40479 vsc40479   2383 Jan 17  2014 perturblog
-rw-rw-r--  1 vsc40479 vsc40479 144306 Mar 15 12:38 phononspure1.png
```

No colors? Missing flag --color=auto

```
vsc40xxx@gligar0x ~# alias ll="ls -l --color=auto"
```

Save in ~/.bashrc for loading in all jobs, ~/.bash_profile for interactive sessions

Exercise 2: Setting up files

1. Navigate to scratch
2. Make a directory called calculations
3. Enter the directory
4. Make an input file called calc1 using touch.
Syntax: touch filename
5. Copy paste using the cp command to calc2 to 10
Syntax: cp oldfile newfile
6. Verify using ls
7. Time for a backup! Go up a level and recursively copy the entire directory ‘calculations’ to ‘calculations.bak’, verify.
8. Finally clean up. Remove both directories using the remove command rm (recursive and forced).

Checkout the help files for unknown commands!

Bash magic

Doing the same thing 10 times is boring.

```
vsc40xxx@gligar0x ~#
for i in {0..10};do touch calc${i};done
```

- {1..10} is the same as 1 2 3 4 5 6 7 8 9 10
- ; creates a linebreak, letting you type multiple commands on one line
- For condition;do [commands];done is a loop

An alternative:

```
vsc40xxx@gligar0x ~#
touch calc{1..10}
```

Can even use other commands

```
vsc40xxx@gligar0x ~#
for i in $(find ./ -name filename);do touch calc${i};done
```

Bash magic

Wildcards also exist:

```
vsc40xxx@gligar0x ~# rm calc*
```

will remove all files starting with calc**

```
vsc40xxx@gligar0x ~# for i in calc*;do rm $i;done
```

DANGEROUS, this will kill all (NO RECYCLE BIN):

```
vsc40xxx@gligar0x ~# rm *
```

An absolute path is of course safest:

```
vsc40xxx@gligar0x ~#
rm /user/scratch/gent/vsc40x/vsc40xxx/calculations/calc1
```

But too long… Relative paths can help

```
vsc40xxx@gligar0x ~#
rm ./calc1 (in current directory)
rm ../calc1 (one level up)
rm …/calc1 (two levels up)
rm ../calculations/calc1 (in the directory calculations one
level up)
```

**check out regular expressions for more advanced pattern matching

Bash magic

A few more useful things:

- When typing a command or path, hit tab to autocomplete
- Bash has a history, navigate with up or down
- If you have bash properly configured you can complete partial commands with history

Exercise 3: more files

1. Make a directory called calculations

2. Make 10 subdirectories named project1 to 10

3. In each folder make files foldername.1 to foldername.10

4. Finally clean up. Remove all directories using the remove command rm

Editing files

We can display text with the echo command

```
vsc40xxx@gligar0x ~# echo ‘Hello world’
```

And we can store it into a file by redirecting the output of echo to the file:

```
vsc40xxx@gligar0x ~# echo ‘Hello world’ > file1
```

> filename: Overwrites file

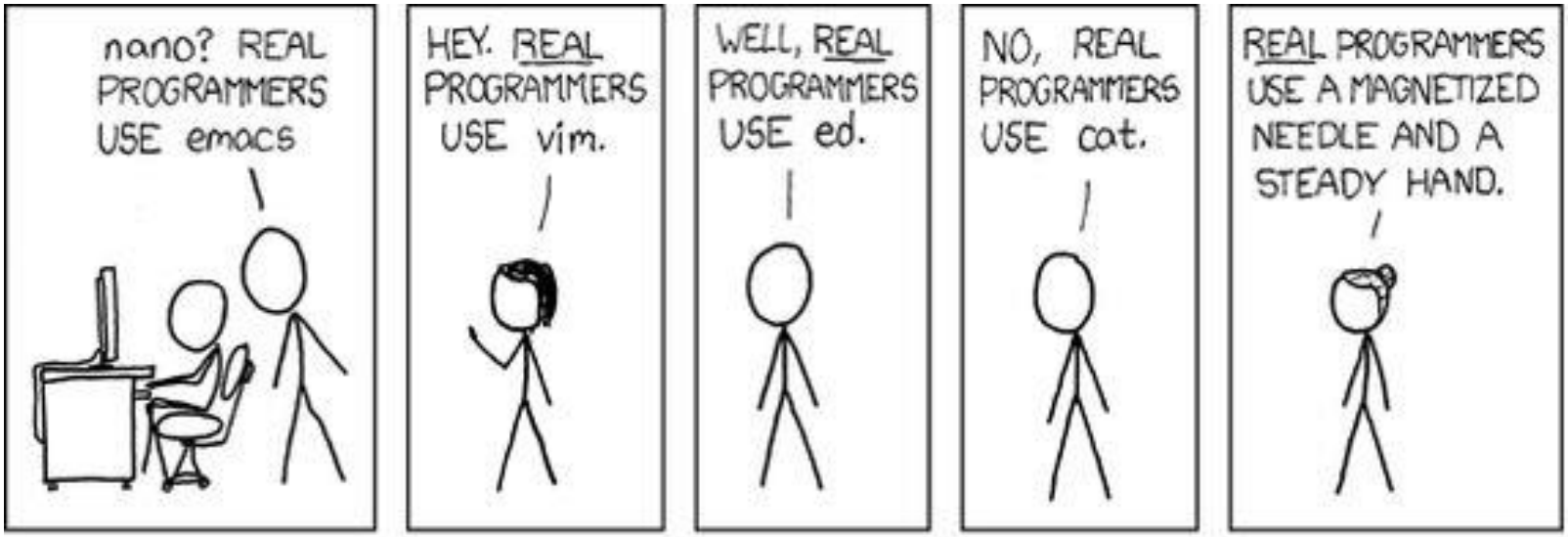
>> filename: appends to file

| command: sends output as input file to next command (piping)

There are even other flavors to just redirect output (stdout) and errors (stderr)!
https://workaround.org/linuxtip/pipes

This is useful to store the output of other commands for later usage,
but not to write a page.

The editor wars



CMM battlefield

```
                  :::
  iLE88Dj.  :jD88888Dj:
.LGitE888D.f8GjjjL8888E;
iE   :8888Et.     .G8888.
;i    E888,        ,8888,
      D888,        :8888:
      D888,        :8888:
      D888,        :8888:
      D888,        :8888:
      888W,        :8888:
      W88W,        :8888:
      W88W:        :8888:
      DGGD:        :8888:
                   :8888:
                   :W888:
                   :8888:
                    E888i
                    tW88D
```

```
 .d8888b.  888b     888 888      888
d88P  Y88b 8888b    888 888      888
888    888 88888b   888 888      888
888        888Y88b 888 888      888
888  88888 888 Y88b888 888      888
888    888 888  Y88888 888      888
Y88b  d88P 888   Y8888 Y88b. .d88P
 "Y8888P88 888    Y888  "Y88888P"

88888b.   8888b.  88888b.   .d88b.
888 "88b     "88b 888 "88b d88""88b
888  888 .d888888 888  888 888  888
888  888 888  888 888  888 Y88..88P
888  888 "Y888888 888  888  "Y88P"
```

Vim is cool (undofile!), but use whatever you like. (see demo)

Exercise 4: editing files

1. Make a file called file1 with vim
2. Enter the text: “This file was made with vim” (i for insert, type, escape for commands, :x for write and exit)
3. Save and close the file
4. Make a file called file2 with nano (^ is ctrl)
5. Enter the text: “This file was made with nano”
6. Save and close the file
7. View both files with cat (wildcards!)
8. Compare the files with the diff command
9. Compare the files with the vimdiff command

Checkout the Vim cheat sheet https://vim.rtorr.com/ **!**

Some more commands

Cat was meant to concatenate files, (less is similar to cat but paged)

```
vsc40xxx@gligar0x ~# cat file1 file2 > file 3
```

Display only top 10 lines

```
vsc40xxx@gligar0x ~# head 10 file
```

Display only bottom lines (tail –f to follow a file which is being written)

```
vsc40xxx@gligar0x ~# tail –n10  file
```

Show only lines containing word or pattern *important*

```
vsc40xxx@gligar0x ~# grep word file
```

Count lines

```
vsc40xxx@gligar0x ~# wc –l file
```

Count lines with word (piping lets you send the output from one command to another)

```
vsc40xxx@gligar0x ~# grep word file | wc –l
```

Download a file:

```
vsc40xxx@gligar0x ~# wget http://url –O file
```

Exercise 5: advanced

1. Download the bible as bible.txt
http://www.gutenberg.org/cache/epub/30/pg30.txt

2. Download the complete collections of shakespeare as shakespeare.txt
https://ocw.mit.edu/ans7870/6/6.006/s08/lecturenotes/files/t8.shakespeare.txt

3. Find out how many times the word bash occurs in both of these works using a single line (commands needed are on the previous slide)

Note: the output files of certain molecular modeling programs are of similar length

Checkout the help files for unknown commands!

Can I submit a job yet?

Each cluster runs queuees using the PBS software:

- **Short queue: up to 12 hours**
- **Long queue: up to 72 hours**

How do we interact with the queue? Commands!

- **qstat –q, : shows info on the queues**
- **qstat, qstat -a, qstat -f : shows specific node numbers**

```
Job ID                    Name             User            Time Use S Queue
------------------------- ---------------- --------------- -------- - -----
40125959.tier1-p-moab-1.tier1 ...elvolkp4ISPIN vsc40479        1538:34: C q72h
40126263.tier1-p-moab-1.tier1 ...llvolrel-spin vsc40479        1668:21: C q72h
40126445.tier1-p-moab-1.tier1 .../IM/Garef     vsc40479        670:11:4 C q24h
40126446.tier1-p-moab-1.tier1 ....0HSEspecific vsc40479        141:40:5 C q24h
```

Can I submit a job yet?

- **Pbsmon shows occupation of cluster**

```
vsc40479@gligar02 ~# pbsmon
 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020
    j    J    j    J    j    J    J    J    J    J    J    j    j    j    j    J    J    J    j    J

 2021 2022 2023 2024 2025 2026 2027 2028 2029 2030 2031 2032 2033 2034 2035 2036 2037 2038 2039 2040
    J    J    J    j    J    J    J    J    j    J    J    J    J    j    j    J    J    J    J    J

 2041 2042 2043 2044 2045 2046 2047 2048 2049 2050 2051 2052 2054 2055 2056 2057 2058 2059 2060 2061
    J    J    J    J    J    _    J    j    j    J    J    J    j    J    J    J    J    J    J    J

 2062 2063 2064 2065 2066 2067 2068 2069 2070 2071 2072 2073 2074 2075 2076 2077 2078 2079 2080 2081
    j    j    j    j    J    J    J    J    j    j    j    J    j    j    j    J    j    j    j    j

 2082 2083 2084 2085 2086 2087 2088 2089 2090 2091 2092 2093 2094 2095 2096 2097 2098 2099 2100 2101
    J    J    J    J    J    J    j    j    j    j    j    j    J    j    j    j    J    J    J    j

 2102 2103 2104 2105 2106 2107 2108 2109 2110 2111 2112 2113 2114 2115 2116 2117 2118 2119 2120 2121
    j    J    J    J    j    j    j    j    j    J    J    j    j    J    J    J    J    J    j    j

 2122 2123 2124 2125 2126 2127 2128 2129 2130 2131 2132 2133 2134 2135 2136 2137 2138 2139 2140 2141
    J    j    j    j    j    j    J    j    j    J    j    J    j    J    j    J    j    J    j    j

 2142 2143 2144 2145 2146 2147 2148 2149 2150 2151 2152 2153 2154 2155 2156 2157 2158 2159 2160
    J    j    j    j    J    j    j    j    j    J    J    J    J    J    J    J    J    j    j

   J full                : 87  |   . offline             : 0   |
   j partial             : 71  |   X down                : 0   |
   _ free                : 1   |   x down_on_error       : 0   |
                               |   o other               : 0   |

Node type:
 ppn=16, physmem=62.8GB, swap=20.0GB, vmem=82.8GB, local disk=876.0GB
```

Switching clusters? Modules!

The module system is used to load software, clusters are also listed as different versions of 'cluster' software

- **module spider cluster: find all cluster modules**

There are modules for all kinds of things such as Python and MATLAB (case-sensitive!)

- **module swap cluster/golett: swaps the cluster to golett**

If a module isn't loaded yet use module load, else use swap

- **module list: shows all currently loaded modules**

- **module purge: resets the modules**

Exercise 6: find the right cluster






	#nodes	CPU	Mem/node	Diskspace/node	Network
Delcatty	160	2 x 8-core Intel E5-2670 (Sandy Bridge @ 2.6 GHz)	64 GB	400 GB	FDR InfiniBand
Phanpy	16	2 x 12-core Intel E5-2680v3 (Haswell-EP @ 2.5 GHz)	512 GB	3x 400 GB (SSD, striped)	FDR InfiniBand
Golett	200	2 x 12-core Intel E5-2680v3 (Haswell-EP @ 2.5 GHz)	64 GB	500 GB	FDR-10 InfiniBand
Swalot	128	2 x 10-core Intel E5-2660v3 (Haswell-EP @ 2.6 GHz)	128 GB	1 TB	FDR InfiniBand
skitty	72	2 x 18-core Intel Xeon Gold 6140 (Skylake @ 2.3 GHz)	192 GB	1 TB 240 GB SSD	EDR InfiniBand
victini	96	2 x 18-core Intel Xeon Gold 6140 (Skylake @ 2.3 GHz)	96 GB	1 TB 240 GB SSD	10 GbE

For:

- **A job using 8 cores, 1GB RAM per core, 10 GB disk**
- **The same, but at least 72 cores, little communication**
- **The same, but fast communication**
- **A job needing 10 GB RAM per core**

Wasting will just lower your priority!

Building a submit script

```
#!/bin/bash
#PBS -N name
#PBS -m ae
#PBS -q long
#PBS -l walltime=71:59:00
#PBS -l nodes=1:ppn=20

ulimit -s unlimited

cd /scratch/leuven/404/vsc40479/IM/results/WP1/511volrel2+
module load cluster
module load scripts
module load VASP/5.4.1-intel-2016a-VTST
mympirun --output tempout vasp_std
exit 0
```

You can execute any script as a command:

1. **Set permissions as executable with chmod 744 filename (look up chmod, chown)**
2. **If you want to use it out of the folder put it in a folder which is in your PATH variable**

See demo

Exercise 7: Submitting a job

1. Make a job file which submits a job in the debug queue
2. Ensure the job only lasts 59 mins and uses 1 core
3. Add instructions make the job print ‘hello world’ (using echo)

4. Add instructions to make the job print ‘hello file’ to a file
5. Add an exit to the job file
6. Submit with qsub
7. Check with ‘watch qstat’
8. Check the .oJOBID file for output, .eJOBID for errors

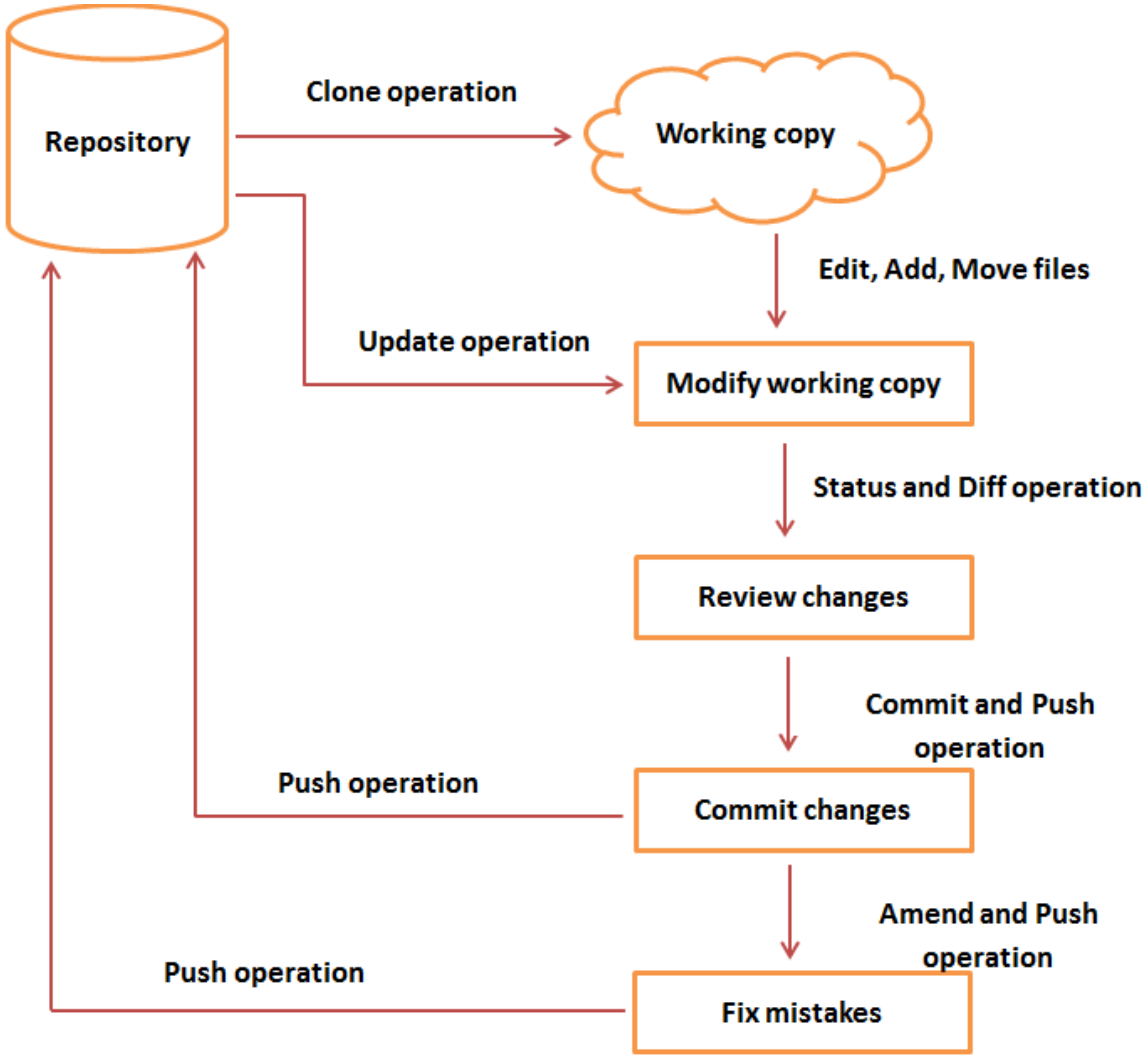
Backups

1. Level 1: cp –R source dest, wastes storage
2. Level 2: scp source user@remote:folder
3. Level 3: rsync –uav source user@remote:folder, can update only new files and show progress (lots of files are slow)
4. Compressing: tar cvf output.tar files (similar zip, more common on linux)
5. Further compress: gzip file.tar (or other files)
6. Fast compress in job: Pigz –rR files.tar: parallel gzip
7. Git: versioning!

Note: you can use moldyn58 for remote backups

Checkout the help files for unknown commands!

Git: a summary



Git: a summary

1. Create a repository on github.com or get url for existing
2. git clone url: creates a local working copy
3. git add file: If you add new files in the directory, add them
4. git commit: Add the changes as a new version to the repo
5. git push: upload changes
6. git pull: download changes

See demo

More: http://rogerdudler.github.io/git-guide/

Other useful commands

- **find: find files in directory**
- **sort: sort input (-n for numeric)**
- **cut: cut out a column of input**
- **set and export: set variables**
- **xargs: apply a command to multiple files**
- **sed: find and replace in text (character-based)**
- **awk: fancier sed (aimed at columns)**
- **Check out bash regular expressions for advanced pattern matching**

More help:

- https://molmod.ugent.be/computer-introductions
- https://www.ugent.be/hpc/en/support
- http://www.stackoverflow.com

The end