🤖

Using the GRIB cluster

ITGrib info - Information for GRIB users

The Grib is a Research Program of two institutions: the IMIM-Hospital del Mar and the Medicine and Life Sciences (MELIS) department of the UPF, besides we are in the PRBB building... therefore in many aspects such as getting access to the internet, or the e.mail, the GRIB is dependant of these institutions.

http://nemo.upf.edu/info/

1. Connecting to `zeus.upf.edu`

1. Connecting from outside through a ssh server called `zeus.upf.edu`:

```
ssh vledesma@zeus.upf.edu
```

2. Connecting to `hydra`, the master node. We shouldn't run things here, as it is the node that assigns resources to other nodes so that things work nicely.

```
ssh hydra
```

🕉️ There is a new server called shiva that uses Ubuntu.

2. Basic SLURM

SLURM is a free and open-source job scheduler for Linux and Unix-like kernels, used by many of the world's supercomputers and computer clusters. A guide to use it and important commands can be found here:

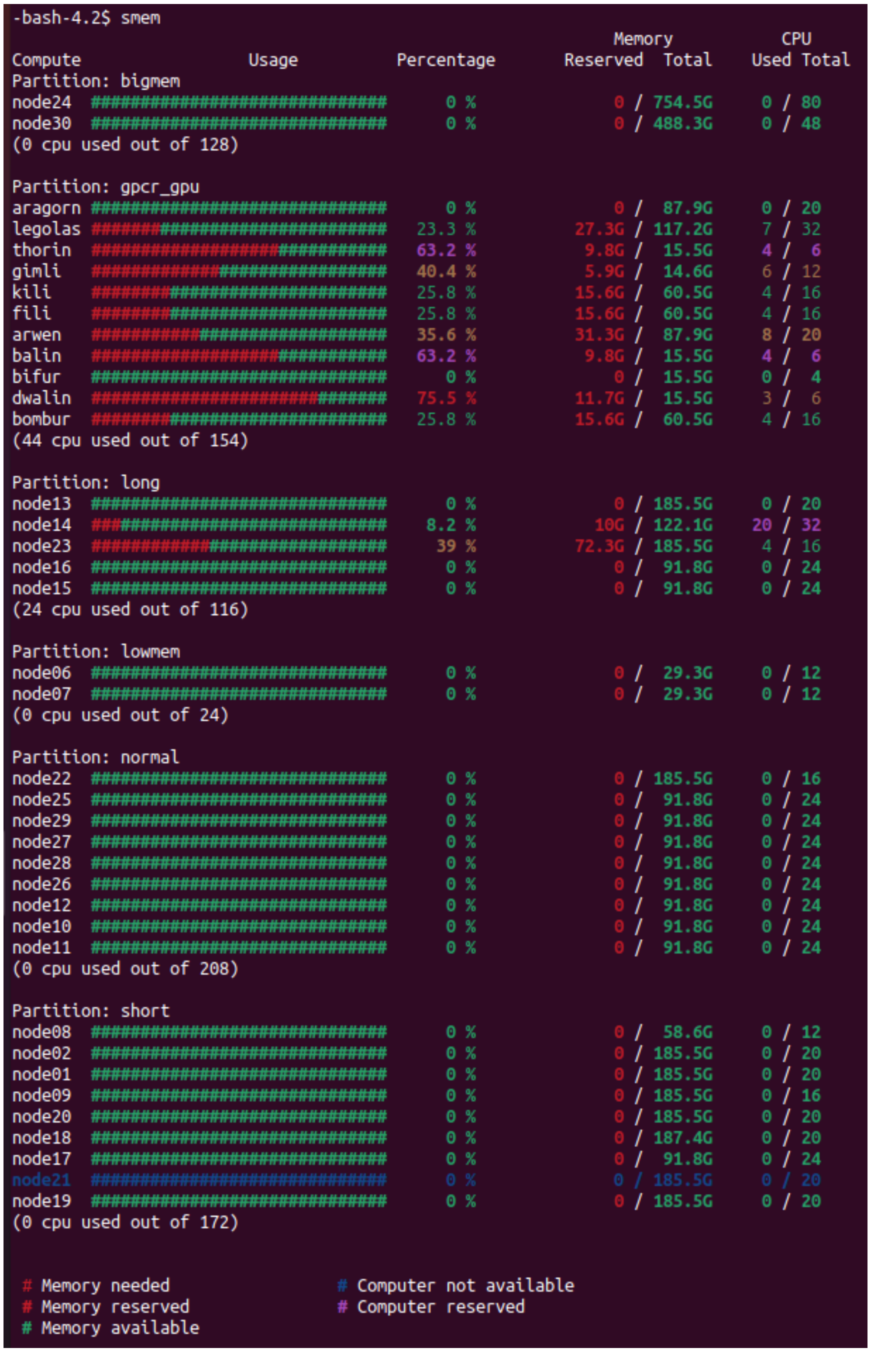
Slurm Workload Manager - Quick Start User Guide

https://slurm.schedmd.com/quickstart.html

2.1. Checking nodes and jobs

1. To see a list of all the nodes and how they are being used:

```
smem
```

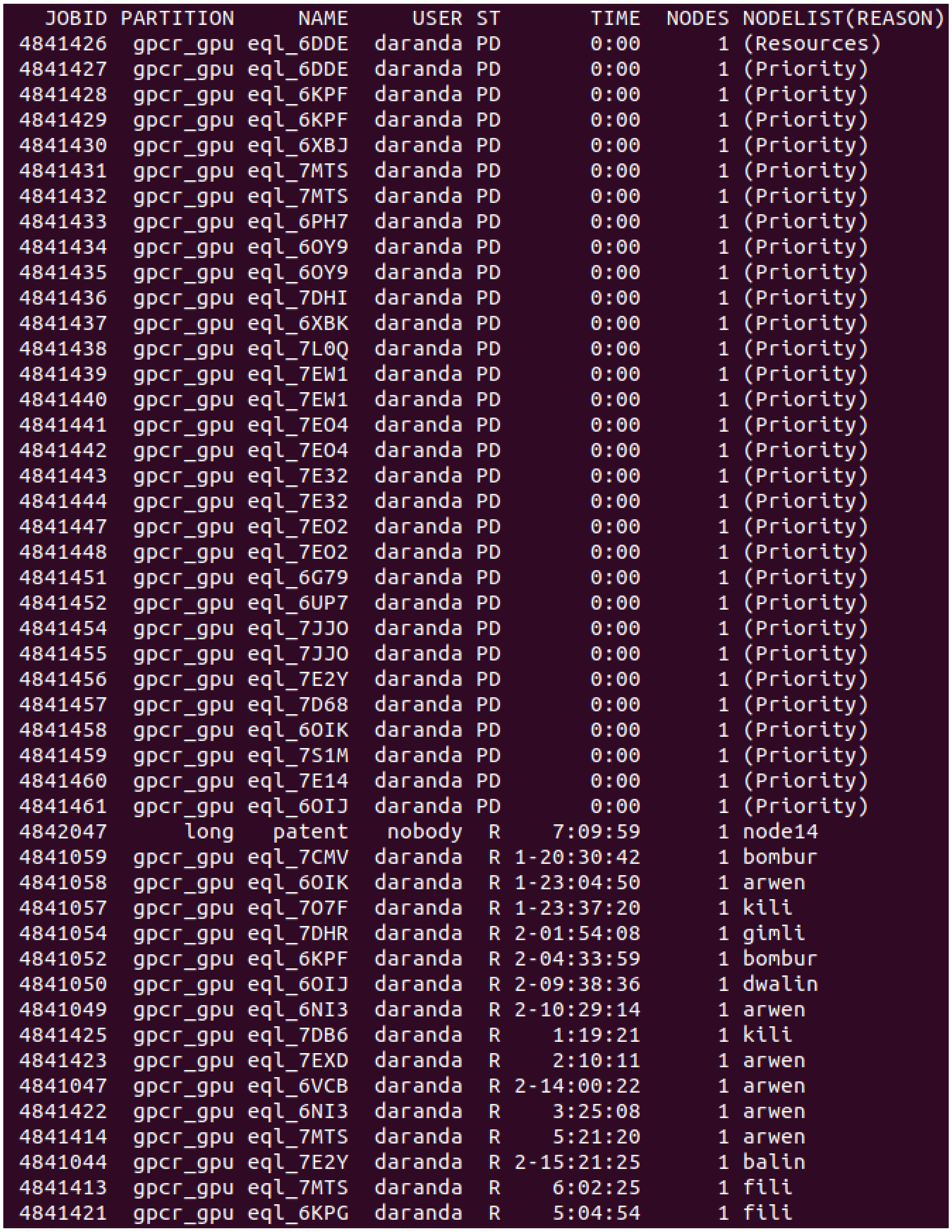


```
-bash-4.2$ smem
                                                             Memory           CPU
Compute                 Usage          Percentage       Reserved  Total    Used Total
Partition: bigmem
node24  ##############################      0 %              0 / 754.5G     0 / 80
node30  ##############################      0 %              0 / 488.3G     0 / 48
(0 cpu used out of 128)

Partition: gpcr_gpu
aragorn ##############################      0 %              0 /  87.9G     0 / 20
legolas ##############################   23.3 %          27.3G / 117.2G     7 / 32
thorin  ##############################   63.2 %           9.8G /  15.5G     4 /  6
gimli   ##############################   40.4 %           5.9G /  14.6G     6 / 12
kili    ##############################   25.8 %          15.6G /  60.5G     4 / 16
fili    ##############################   25.8 %          15.6G /  60.5G     4 / 16
arwen   ##############################   35.6 %          31.3G /  87.9G     8 / 20
balin   ##############################   63.2 %           9.8G /  15.5G     4 /  6
bifur   ##############################      0 %              0 /  15.5G     0 /  4
dwalin  ##############################   75.5 %          11.7G /  15.5G     3 /  6
bombur  ##############################   25.8 %          15.6G /  60.5G     4 / 16
(44 cpu used out of 154)

Partition: long
node13  ##############################      0 %              0 / 185.5G     0 / 20
node14  ##############################    8.2 %            10G / 122.1G    20 / 32
node23  ##############################     39 %          72.3G / 185.5G     4 / 16
node16  ##############################      0 %              0 /  91.8G     0 / 24
node15  ##############################      0 %              0 /  91.8G     0 / 24
(24 cpu used out of 116)

Partition: lowmem
node06  ##############################      0 %              0 /  29.3G     0 / 12
node07  ##############################      0 %              0 /  29.3G     0 / 12
(0 cpu used out of 24)

Partition: normal
node22  ##############################      0 %              0 / 185.5G     0 / 16
node25  ##############################      0 %              0 /  91.8G     0 / 24
node29  ##############################      0 %              0 /  91.8G     0 / 24
node27  ##############################      0 %              0 /  91.8G     0 / 24
node28  ##############################      0 %              0 /  91.8G     0 / 24
node26  ##############################      0 %              0 /  91.8G     0 / 24
node12  ##############################      0 %              0 /  91.8G     0 / 24
node10  ##############################      0 %              0 /  91.8G     0 / 24
node11  ##############################      0 %              0 /  91.8G     0 / 24
(0 cpu used out of 208)

Partition: short
node08  ##############################      0 %              0 /  58.6G     0 / 12
node02  ##############################      0 %              0 / 185.5G     0 / 20
node01  ##############################      0 %              0 / 185.5G     0 / 20
node09  ##############################      0 %              0 / 185.5G     0 / 16
node20  ##############################      0 %              0 / 185.5G     0 / 20
node18  ##############################      0 %              0 / 187.4G     0 / 20
node17  ##############################      0 %              0 /  91.8G     0 / 24
node21  ##############################      0 %              0 / 185.5G     0 / 20
node19  ##############################      0 %              0 / 185.5G     0 / 20
(0 cpu used out of 172)


 # Memory needed                 # Computer not available
 # Memory reserved               # Computer reserved
 # Memory available
```

Our nodes are the `gpcr_gpu` ones.

2. To see the queue of jobs to be run:

```
squeue
```



```
   JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
 4841426  gpcr_gpu eql_6DDE  daranda PD       0:00      1 (Resources)
 4841427  gpcr_gpu eql_6DDE  daranda PD       0:00      1 (Priority)
 4841428  gpcr_gpu eql_6KPF  daranda PD       0:00      1 (Priority)
 4841429  gpcr_gpu eql_6KPF  daranda PD       0:00      1 (Priority)
 4841430  gpcr_gpu eql_6XBJ  daranda PD       0:00      1 (Priority)
 4841431  gpcr_gpu eql_7MTS  daranda PD       0:00      1 (Priority)
 4841432  gpcr_gpu eql_7MTS  daranda PD       0:00      1 (Priority)
 4841433  gpcr_gpu eql_6PH7  daranda PD       0:00      1 (Priority)
 4841434  gpcr_gpu eql_6OY9  daranda PD       0:00      1 (Priority)
 4841435  gpcr_gpu eql_6OY9  daranda PD       0:00      1 (Priority)
 4841436  gpcr_gpu eql_7DHI  daranda PD       0:00      1 (Priority)
 4841437  gpcr_gpu eql_6XBK  daranda PD       0:00      1 (Priority)
 4841438  gpcr_gpu eql_7L0Q  daranda PD       0:00      1 (Priority)
 4841439  gpcr_gpu eql_7EW1  daranda PD       0:00      1 (Priority)
 4841440  gpcr_gpu eql_7EW1  daranda PD       0:00      1 (Priority)
 4841441  gpcr_gpu eql_7EO4  daranda PD       0:00      1 (Priority)
 4841442  gpcr_gpu eql_7EO4  daranda PD       0:00      1 (Priority)
 4841443  gpcr_gpu eql_7E32  daranda PD       0:00      1 (Priority)
 4841444  gpcr_gpu eql_7E32  daranda PD       0:00      1 (Priority)
 4841447  gpcr_gpu eql_7EO2  daranda PD       0:00      1 (Priority)
 4841448  gpcr_gpu eql_7EO2  daranda PD       0:00      1 (Priority)
 4841451  gpcr_gpu eql_6G79  daranda PD       0:00      1 (Priority)
 4841452  gpcr_gpu eql_6UP7  daranda PD       0:00      1 (Priority)
 4841454  gpcr_gpu eql_7JJO  daranda PD       0:00      1 (Priority)
 4841455  gpcr_gpu eql_7JJO  daranda PD       0:00      1 (Priority)
 4841456  gpcr_gpu eql_7E2Y  daranda PD       0:00      1 (Priority)
 4841457  gpcr_gpu eql_7D68  daranda PD       0:00      1 (Priority)
 4841458  gpcr_gpu eql_6OIK  daranda PD       0:00      1 (Priority)
 4841459  gpcr_gpu eql_7S1M  daranda PD       0:00      1 (Priority)
 4841460  gpcr_gpu eql_7E14  daranda PD       0:00      1 (Priority)
 4841461  gpcr_gpu eql_6OIJ  daranda PD       0:00      1 (Priority)
 4842047      long   patent   nobody  R    7:09:59      1 node14
 4841059  gpcr_gpu eql_7CMV  daranda  R 1-20:30:42      1 bombur
 4841058  gpcr_gpu eql_6OIK  daranda  R 1-23:04:50      1 arwen
 4841057  gpcr_gpu eql_7O7F  daranda  R 1-23:37:20      1 kili
 4841054  gpcr_gpu eql_7DHR  daranda  R 2-01:54:08      1 gimli
 4841052  gpcr_gpu eql_6KPF  daranda  R 2-04:33:59      1 bombur
 4841050  gpcr_gpu eql_6OIJ  daranda  R 2-09:38:36      1 dwalin
 4841049  gpcr_gpu eql_6NI3  daranda  R 2-10:29:14      1 arwen
 4841425  gpcr_gpu eql_7DB6  daranda  R    1:19:21      1 kili
 4841423  gpcr_gpu eql_7EXD  daranda  R    2:10:11      1 arwen
 4841047  gpcr_gpu eql_6VCB  daranda  R 2-14:00:22      1 arwen
 4841422  gpcr_gpu eql_6NI3  daranda  R    3:25:08      1 arwen
 4841414  gpcr_gpu eql_7MTS  daranda  R    5:21:20      1 arwen
 4841044  gpcr_gpu eql_7E2Y  daranda  R 2-15:21:25      1 balin
 4841413  gpcr_gpu eql_7MTS  daranda  R    6:02:25      1 fili
 4841421  gpcr_gpu eql_6KPG  daranda  R    5:04:54      1 fili
```

```
# Only my jobs
squeue -u vledesma
```

2.2. Queuing jobs

This should be through SLURM, writing a script and executing it with `sbatch`.

This is an example script that runs ACEMD. The scripts must be written in bash.

run_acemd.sh

To run it, the command is:

```
sbatch run_acemd.sh
```

More about how to write these scripts:

Sample SLURM Scripts

Below are a number of sample scripts that can be used as a template for building your own SLURM submission scripts for use on HiPerGator 2.0. These scripts are also located at: /data/training/SLURM/, and can be copied from there.

https://help.rc.ufl.edu/doc/Sample_SLURM_Scripts

In case of doubt as to the number of GPUs, nodes, MBs… leave it blank and they will be automatically assigned.

The scripts should be written in such a way that the program is execute inside a node (`home/vledesma`), **NOT** in `gpcr/users/vledesma`. The example script contains code to automatically move the results back to `gpcr/users/vledesma`.

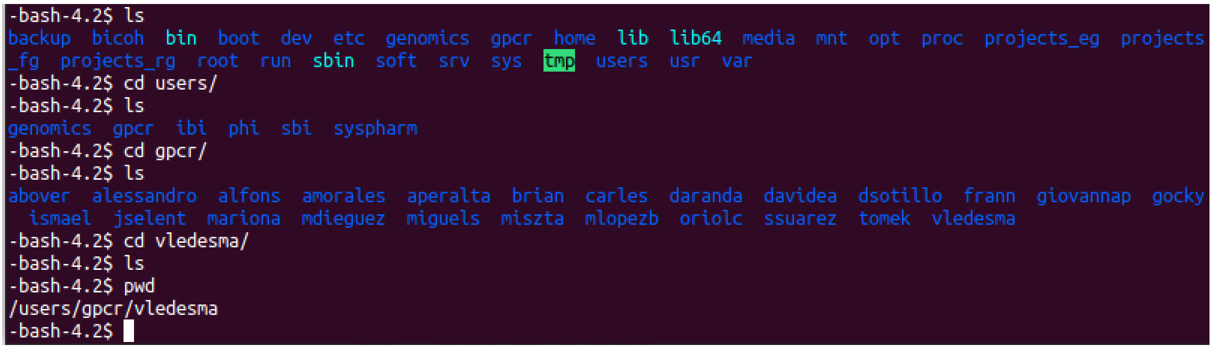
```
# Once the simulations is completed, store it on the STORE_FOLDER path (usually /gpcr/path)
mkdir -p ${STORE_FOLDER}
cp ${LOCAL_RUN_FOLDER}/* ${STORE_FOLDER}/
```

2.3. Cancelling a queued process

```
# Cancel a specific process
scancel <jobid>

# Cancel all my processes
scancel -u vledesma
```

3. My files

- Everything that is mine and that I don't want deleted, should go to `/gpcr/users/vledesma`.



```
-bash-4.2$ ls
backup  bicoh  bin  boot  dev  etc  genomics  gpcr  home  lib  lib64  media  mnt  opt  proc  projects_eg  projects
_fg  projects_rg  root  run  sbin  soft  srv  sys  tmp  users  usr  var
-bash-4.2$ cd users/
-bash-4.2$ ls
genomics  gpcr  ibi  phi  sbi  syspharm
-bash-4.2$ cd gpcr/
-bash-4.2$ ls
abover  alessandro  alfons  amorales  aperalta  brian  carles  daranda  davidea  dsotillo  frann  giovannap  gocky
  ismael  jselent  mariona  mdieguez  miguels  miszta  mlopezb  oriolc  ssuarez  tomek  vledesma
-bash-4.2$ cd vledesma/
-bash-4.2$ ls
-bash-4.2$ pwd
/users/gpcr/vledesma
-bash-4.2$ 
```

3.1. Creating a tunnel

To copy files to and from the server, a tunnel must be created.

```
# Create the tunnel
ssh -f -X vledesma@zeus.upf.edu -N -L 5555:hydra.prib.upf.edu:22

# Connect to hydra using the tunnel
ssh -p 5555 vledesma@localhost
```

3.2. Using the tunnel to see the server files in my local computer (doesn’t work)

```
sudo sshfs -o allow_other -p 5555 vledesma@localhost:/gpcr/users/vledesma /gpcr/users/vledesma
```

3.3. Copying files to and from the server

```
# Copy files through tunnel
scp -P 5555 <source> <destination>

# Example
scp -P 5555 vledesma@localhost:path/to/file path/to/file/local
```

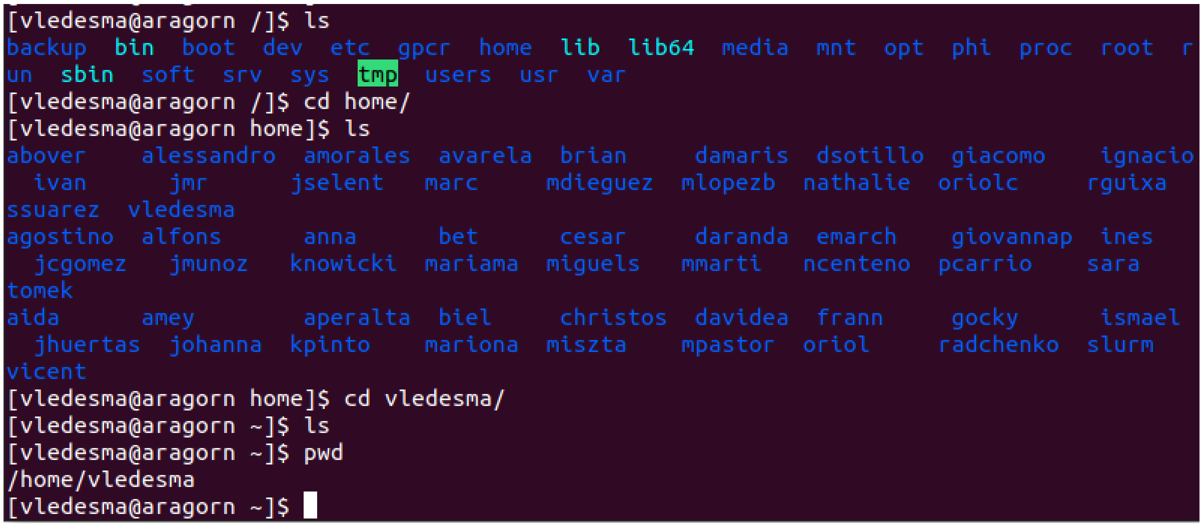
4. Running a job inside a node

> ⏰ Ideally, jobs should be queued as seen before.

1. Choose a node that is not being used. For example:

```
ssh aragorn
```

2. Remember everything should be run on `home/vledesma`.



```
[vledesma@aragorn /]$ ls
backup  bin  boot  dev  etc  gpcr  home  lib  lib64  media  mnt  opt  phi  proc  root  r
un  sbin  soft  srv  sys  tmp  users  usr  var
[vledesma@aragorn /]$ cd home/
[vledesma@aragorn home]$ ls
abover    alessandro  amorales  avarela  brian     damaris  dsotillo  giacomo    ignacio
  ivan      jmr       jselent   marc     mdieguez  mlopezb  nathalie  oriolc     rguixa
ssuarez  vledesma
agostino  alfons      anna      bet      cesar     daranda  emarch    giovannap  ines
  jcgomez   jmunoz    knowicki  mariama  miguels   mmarti   ncenteno  pcarrio    sara
tomek
aida      amey        aperalta  biel     christos  davidea  frann     gocky      ismael
  jhuertas  johanna   kpinto    mariona  miszta    mpastor  oriol     radchenko  slurm
vicent
[vledesma@aragorn home]$ cd vledesma/
[vledesma@aragorn ~]$ ls
[vledesma@aragorn ~]$ pwd
/home/vledesma
[vledesma@aragorn ~]$ 
```

3. Check other processes being run with `htop` and the GPU usage with `nvidia-smi` to avoid collapsing the computer. To exit htop press `<q>`.

5. Loading a module

For example, loading Miniconda or VMD:

```
module load <package> (e.g. Miniconda3/4.12.0 or VMD)
```

> 🔎 If you don't know where a specific package is, ask Alfons.