Practical use of DelftBlue

Dennis Palagin
27 June 2022

TU Delft

Practical use of DelftBlue

- Recap: what is a cluster computer
- Accessing the system
- File systems and data transfer
- Queuing, accounting
- Module system (lmod)
- When (not to) use srun/mpirun/mpiexec
- Practical exercise

Practical use of DelftBlue

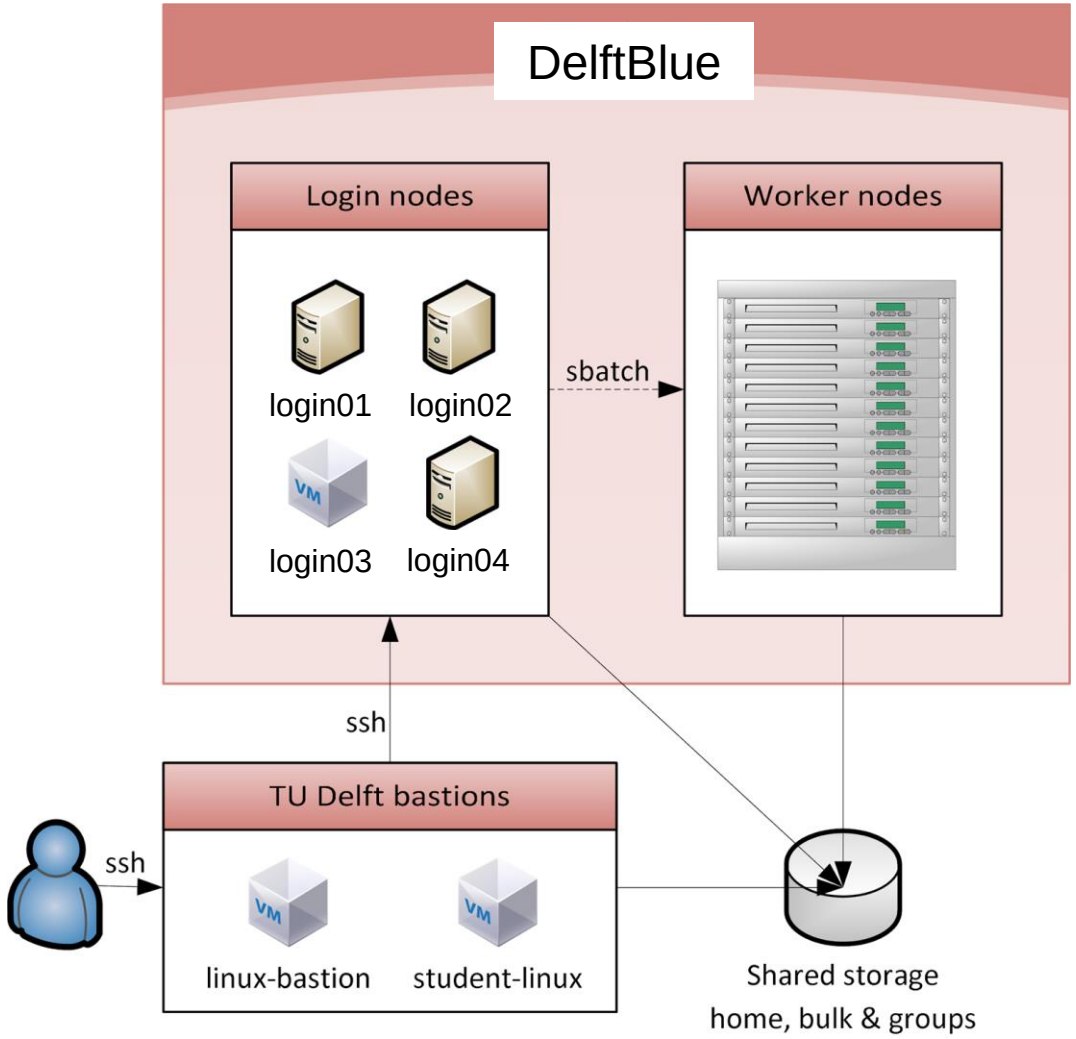
- Recap: what is a cluster computer
- Accessing the system
- File systems and data transfer
- Queuing, accounting
- Module system (lmod)
- When (not to) use srun/mpirun/mpiexec
- Practical exercise

TU Delft

What is a cluster?

- Collection of (large) computing resources:
  - Processors (CPUs)
  - Graphics processors (GPUs)
  - Memory
  - Storage
- Shared with other users
- Makes it possible to:
  - Schedule lots of jobs
  - Do a lot of computational work
    - Use a lot of threads
    - Run long computations
    - Use multiple computers together
  - Compute on big data sets
  - Use GPUs when your own computer doesn’t have one

What is a cluster?



TU Delft

DelftBlue

- Fast and flexible
- With a peak performance over 1 petaflop/s
- 10.000 CPU cores
- Over 200 compute nodes, 10 GPU nodes based on Nvidia Tesla v100 and 2 special nodes for interactive work.
- High-Speed Interconnect based on Mellanox InfiniBand
- And a 700TB high-speed parallel storage subsystem

TU Delft

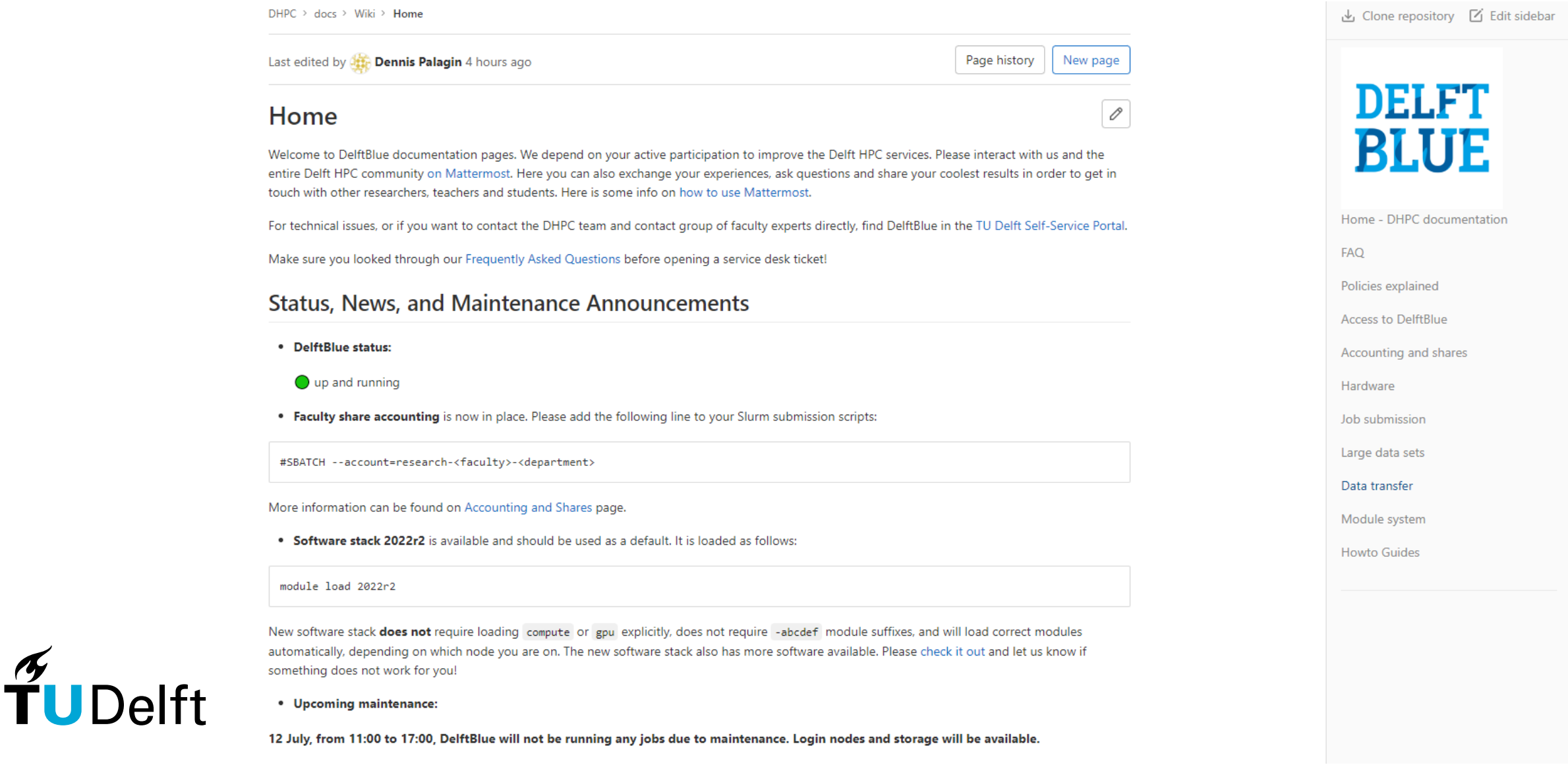
Practical use of DelftBlue

- Recap: what is a cluster computer
- Accessing the system
- File systems and data transfer
- Queuing, accounting
- Module system (lmod)
- When (not to) use srun/mpirun/mpiexec
- Practical exercise

TU Delft

Wiki:

- Website: https://www.tudelft.nl/dhpc/documentation



DHPC › docs › Wiki › Home

Last edited by **Dennis Palagin** 4 hours ago

Page history | New page

Home

Welcome to DelftBlue documentation pages. We depend on your active participation to improve the Delft HPC services. Please interact with us and the entire Delft HPC community on Mattermost. Here you can also exchange your experiences, ask questions and share your coolest results in order to get in touch with other researchers, teachers and students. Here is some info on how to use Mattermost.

For technical issues, or if you want to contact the DHPC team and contact group of faculty experts directly, find DelftBlue in the TU Delft Self-Service Portal.

Make sure you looked through our Frequently Asked Questions before opening a service desk ticket!

Status, News, and Maintenance Announcements

- **DelftBlue status:**

  🟢 up and running

- **Faculty share accounting** is now in place. Please add the following line to your Slurm submission scripts:

```
#SBATCH --account=research-<faculty>-<department>
```

More information can be found on Accounting and Shares page.

- **Software stack 2022r2** is available and should be used as a default. It is loaded as follows:

```
module load 2022r2
```

New software stack **does not** require loading `compute` or `gpu` explicitly, does not require `-abcdef` module suffixes, and will load correct modules automatically, depending on which node you are on. The new software stack also has more software available. Please check it out and let us know if something does not work for you!

- **Upcoming maintenance:**

12 July, from 11:00 to 17:00, DelftBlue will not be running any jobs due to maintenance. Login nodes and storage will be available.

Clone repository | Edit sidebar

DELFT BLUE

Home - DHPC documentation

FAQ

Policies explained

Access to DelftBlue

Accounting and shares

Hardware

Job submission

Large data sets

Data transfer

Module system

Howto Guides

TU Delft

ssh

Anyone with TU Delft <netid> should be able to SSH to DelftBlue:

```
user@laptop:~ $ ssh <netid>@login.delftblue.tudelft.nl
```

This will log you in into one of the four login nodes (login01, login02, login03, or login04).
Your home directory is directly accessible (via /home/<netid>/):

```
[<netid>@login01 ~]$ echo $HOME
/home/NetID

[<netid>@login01 ~]$ ls -l

drwxr-xr-x  3 <netid> domain users     1 Mar 17 13:51 calcs
drwxr-xr-x  3 <netid> domain users     1 Mar 17 13:27 codes
lrwxrwxrwx  1 <netid> domain users    21 Mar 17 14:27 scratch -> /scratch/NetID
drwxr-xr-x  3 <netid> domain users     1 Mar 17 13:27 tools
```

Jumping via bastion

Important note: a direct SSH to DelftBlue from outside of the university network is impossible!
For the access from outside of the university network, you have two options:

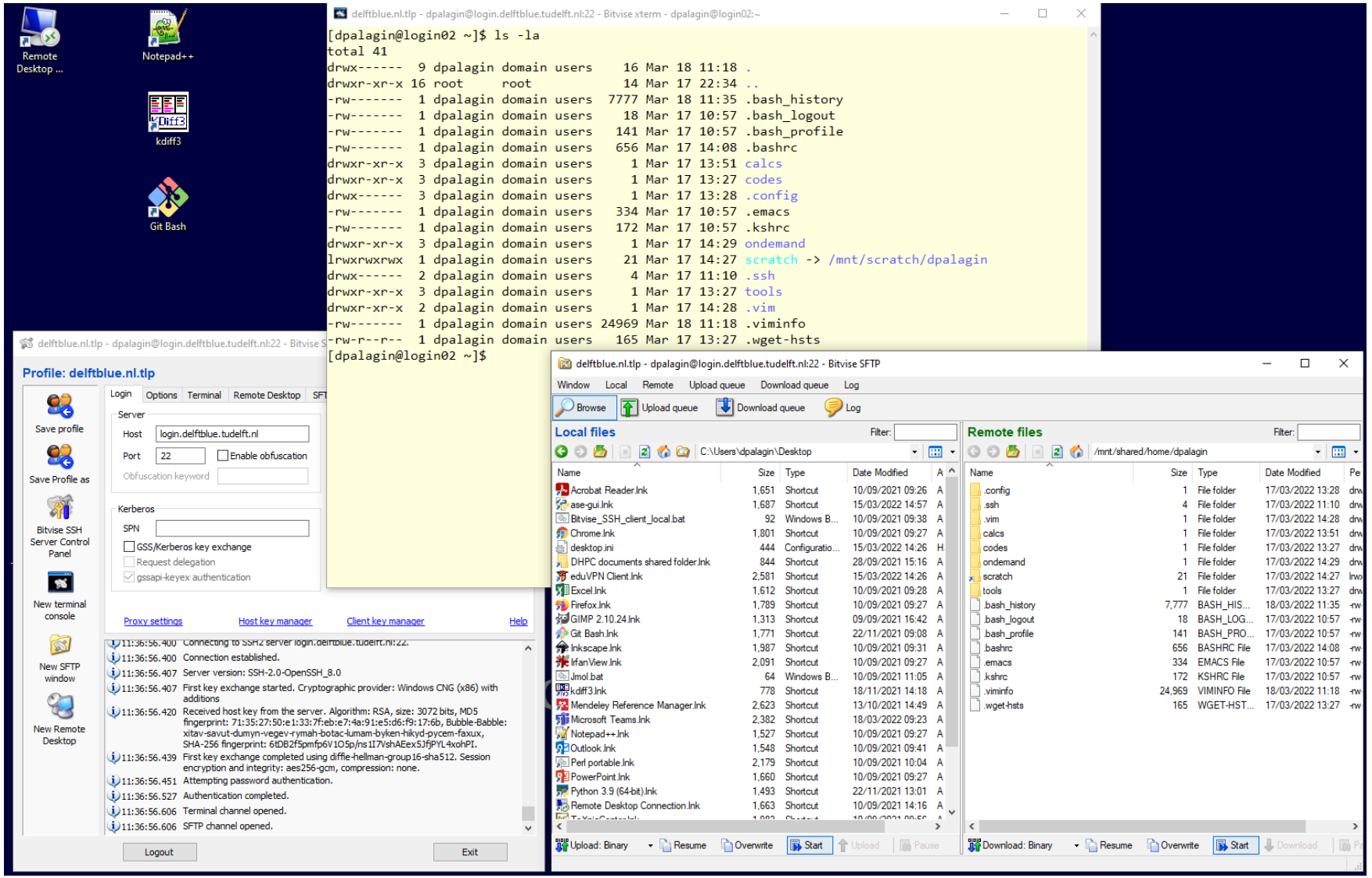
1. Use TU Delft's EduVPN. Once connected via VPN, you can SSH to DelftBlue directly.

2. It is also possible to access DelftBlue without VPN, via the so-called linux-bastion server.

```
user@laptop:~ $ ssh <netid>@linux-bastion.tudelft.nl
<netid>@linux-bastion.tudelft.nl's password:

                  ____  ____ _____
 ___ _ ____   _|___ \|___ \___  |
/ __| '__\ \ / / __) | __) | / /
\__ \ |   \ V / / __/ / __/ / /
|___/_|    \_/ |_____|_____/_/

Last login: Wed Mar 23 13:52:40 2022 from XXX.XXX.XXX.XXX

[<netid>@srv227 ~]$
```

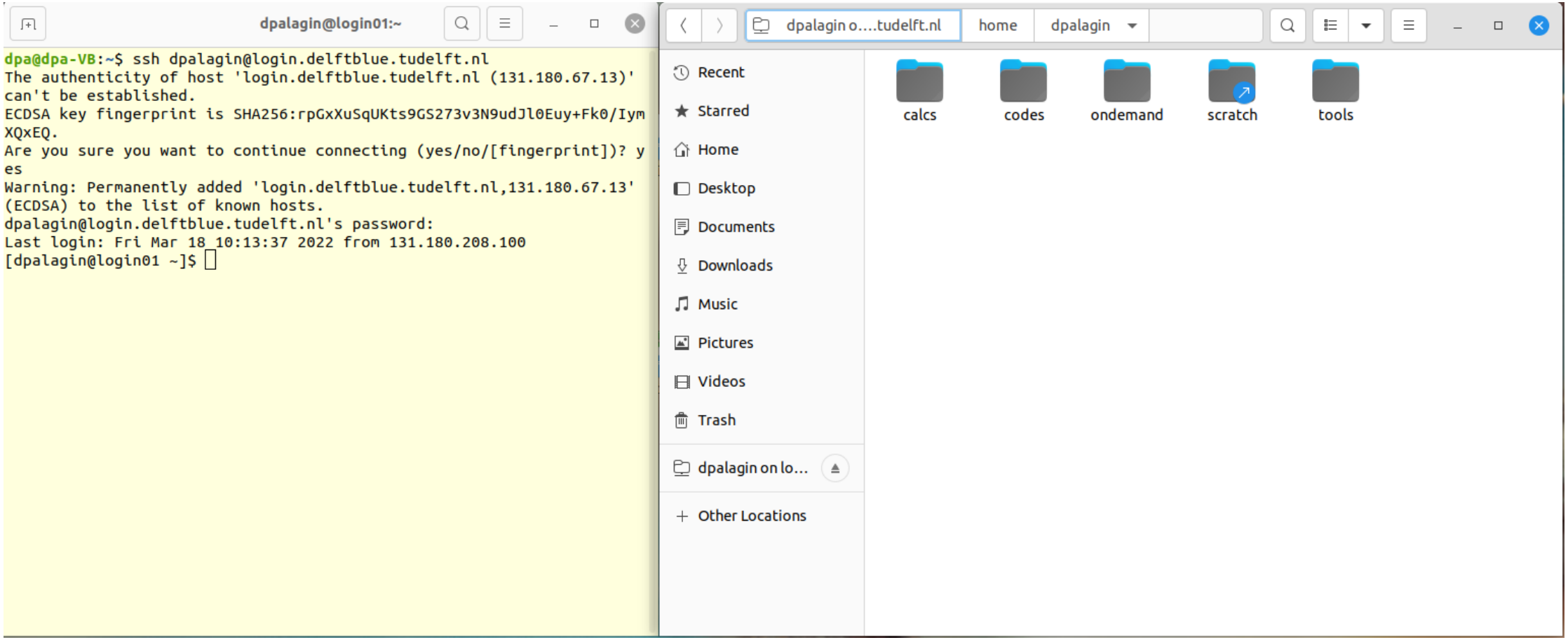
TU Delft

Graphical tools Windows: PuTTy, Bitvise SSH, MobaXterm, etc…



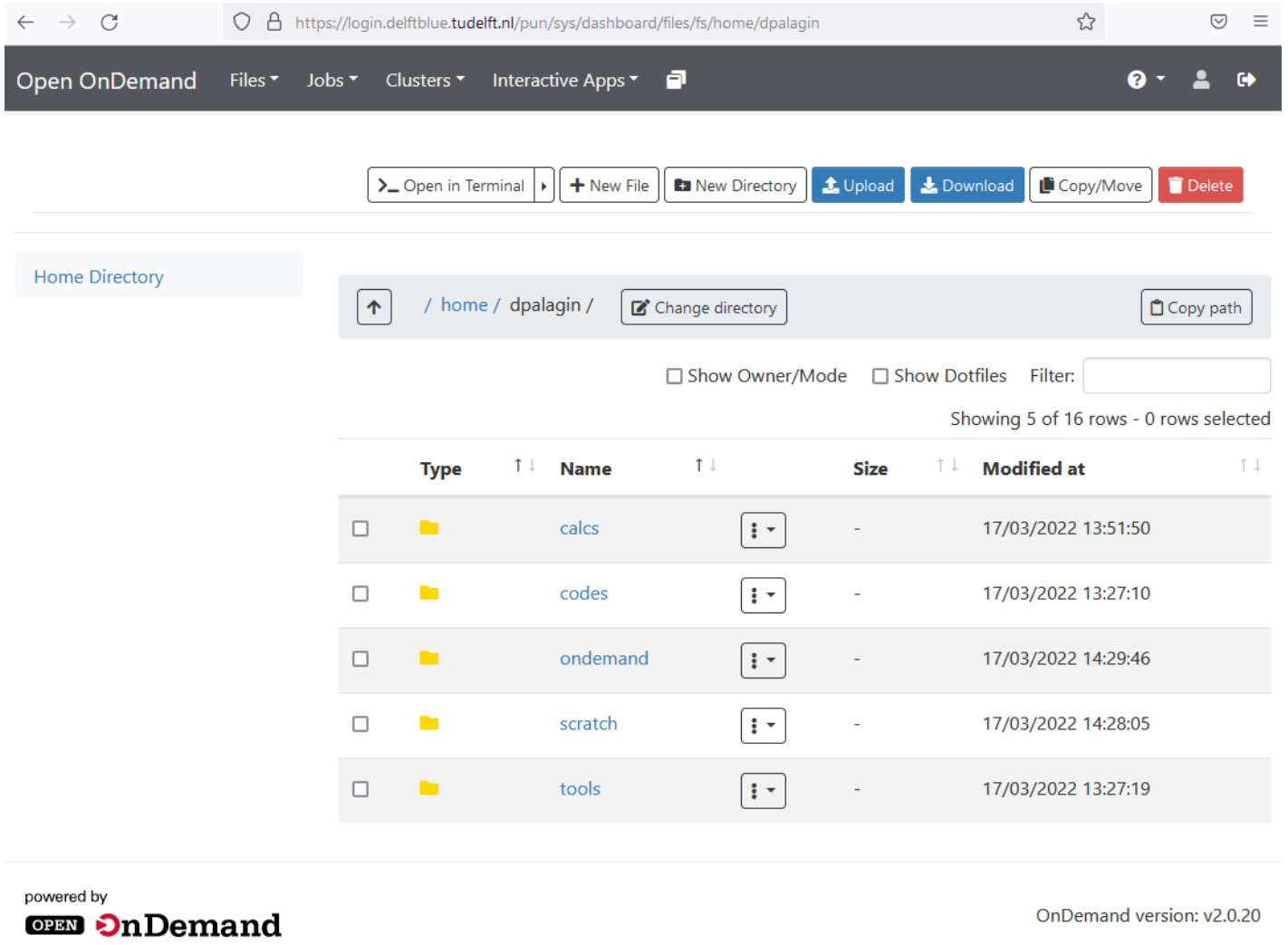
TU Delft

Graphical tools Linux: ssh in terminal + sftp via file manager



TU Delft

Web tools: OpenOnDemand



TU Delft

Practical use of DelftBlue

- Recap: what is a cluster computer
- Accessing the system
- File systems and data transfer
- Queuing, accounting
- Module system (lmod)
- When (not to) use srun/mpirun/mpiexec
- Practical exercise

TU Delft

scp

1. Transfer a file from your computer to DelftBlue:

```
user@laptop:~ $ scp localfile <netid>@login.delftblue.tudelft.nl:~/destination_on_DelftBlue/
```

2. Transfer a folder from your computer to DelftBlue:

```
user@laptop:~ $ scp -r localfile <netid>@login.delftblue.tudelft.nl:~/destination_on_DelftBlue/
```

3. Transfer a file from DelftBlue to your computer:

```
user@laptop:~ $ scp <netid>@login.delftblue.tudelft.nl:~/folder_on_DelftBlue/remotefile ./
```

4. Transfer a folder from DelftBlue to your computer:

```
user@laptop:~ $ scp -r <netid>@login.delftblue.tudelft.nl:~/folder_on_DelftBlue ./
```

rsync

rsync does not just blindly copy files, but instead synchronizes the source with the destination:

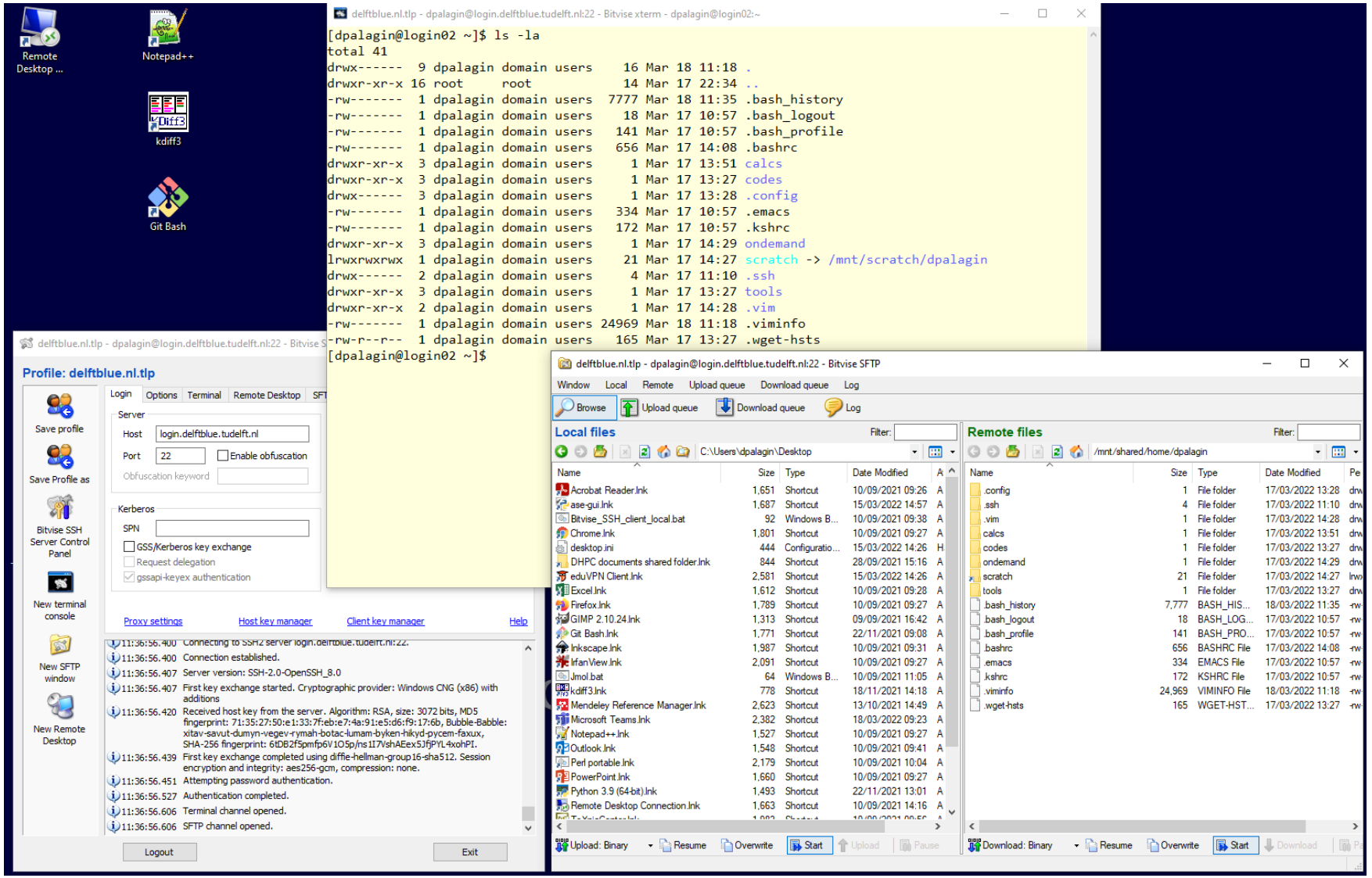
```
user@laptop:~ $ rsync -av ${source} ${target}
```

Please note: some file/folder permissions might not be compatible with project drives, e.g. on staff-umbrella. If you get rsync: failed to set permissions error when e.g. rsync'ing to your project drive, please use the --no-perms flag, e.g.:

```
user@laptop:~ $ rsync -av --no-perms ${source} ${target-on-network-drive}
```

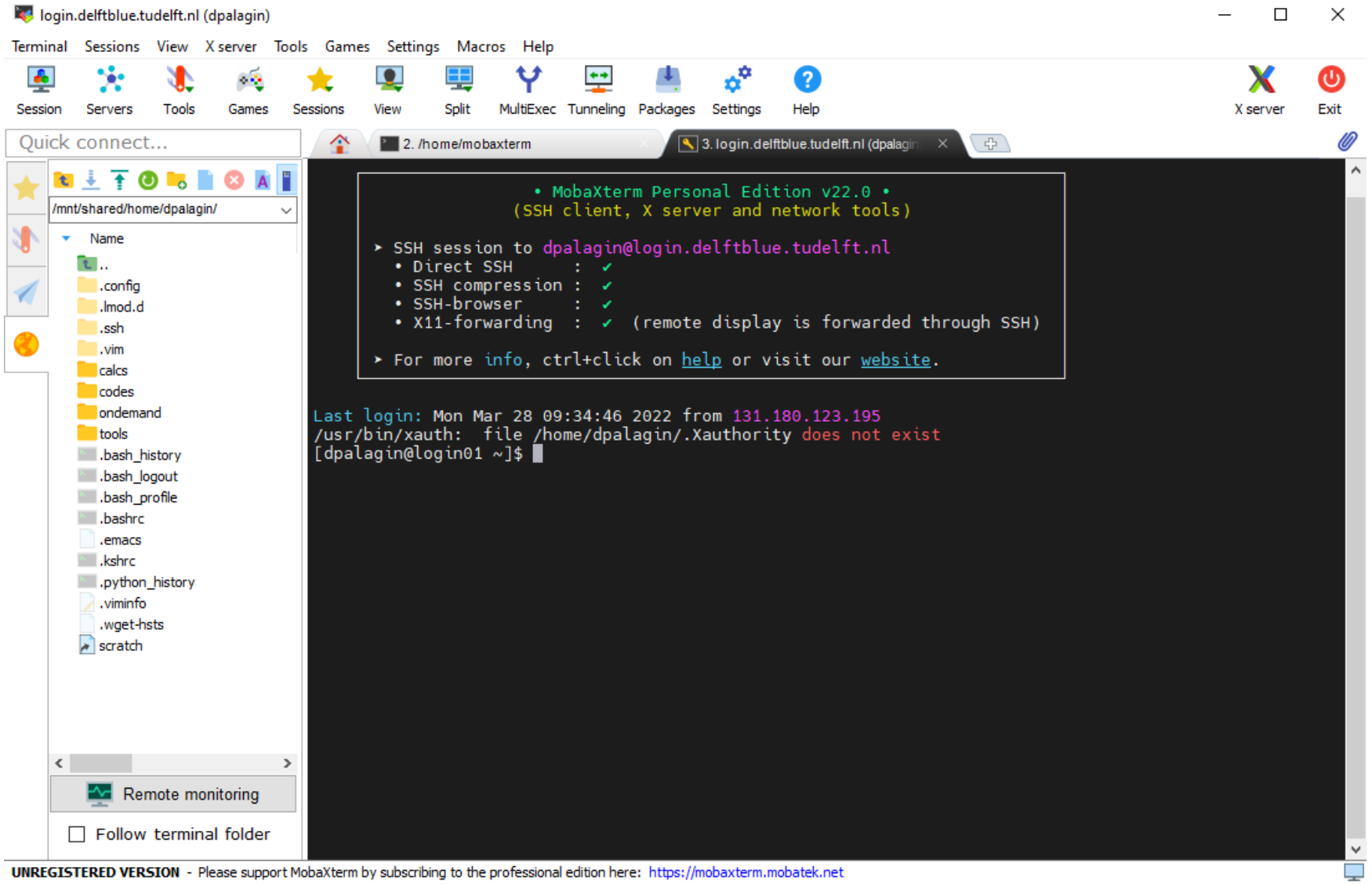
TU Delft

Graphical tools Windows: Bitvise SSH client



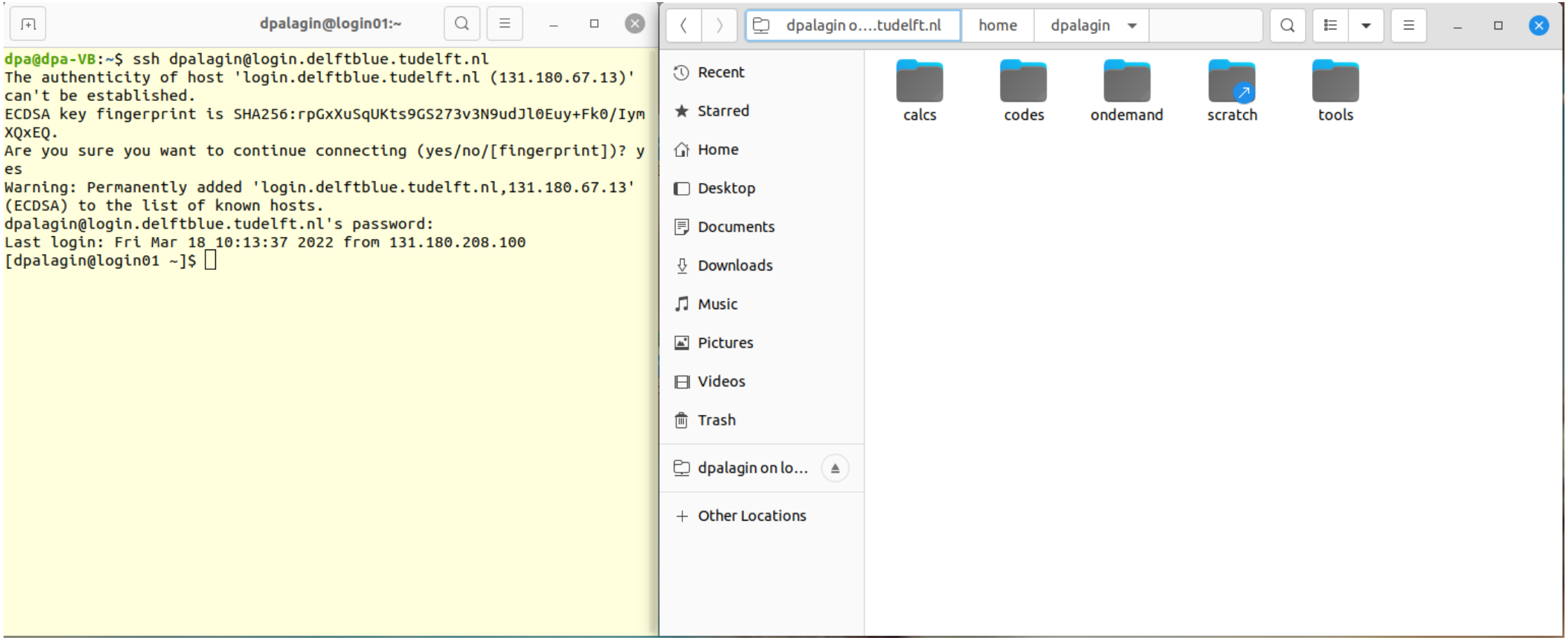
TU Delft

Graphical tools Windows: MobaXterm



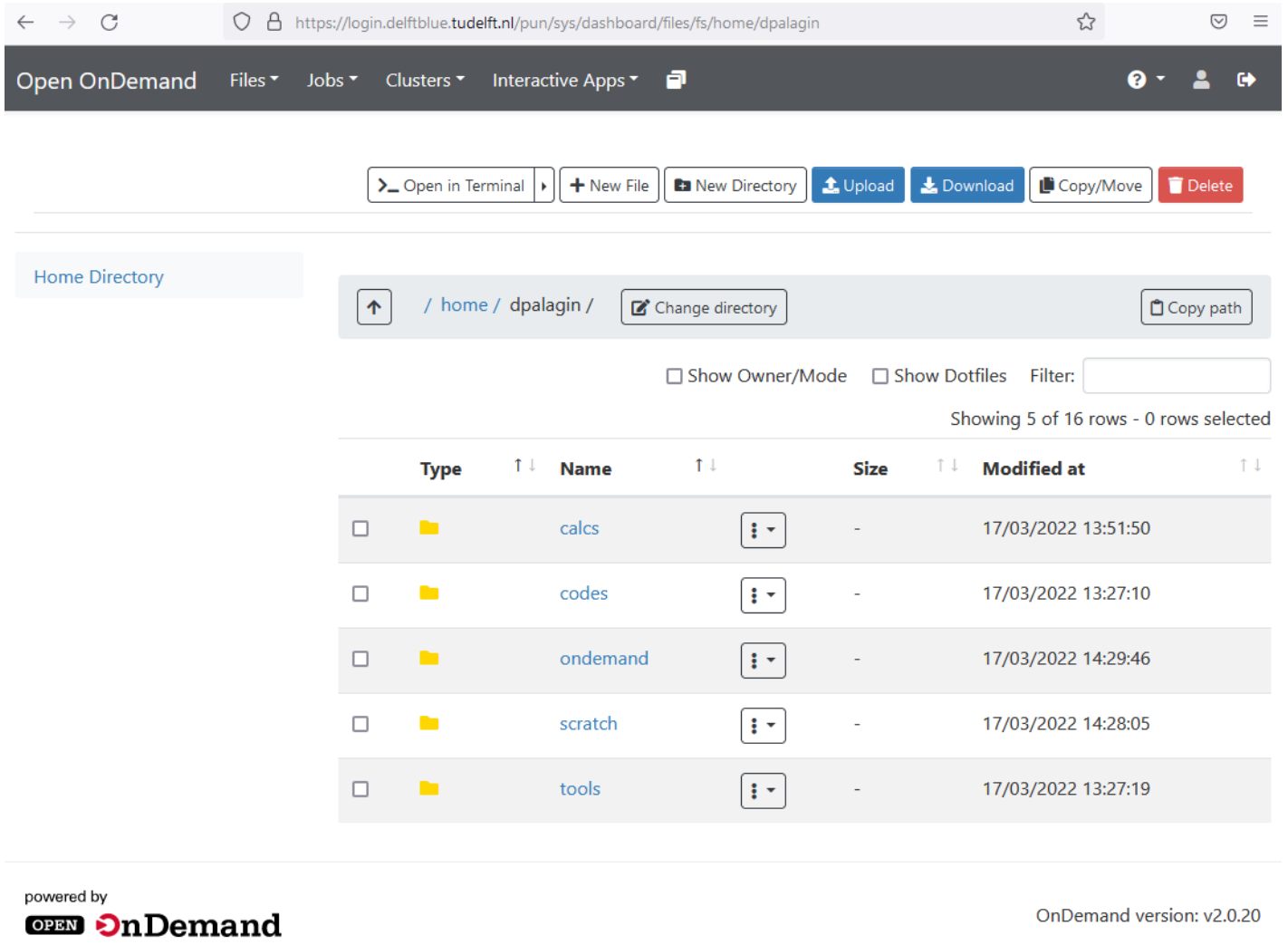
TU Delft

Graphical tools Linux: sftp via file manager



TU Delft

Web tools: OpenOnDemand



TU Delft

Network drives

```
[NetID@login02 tudelft.net]$ ls -l

total 104
drwxr-xr-x   12 root root  4096 Mar 16 15:16 staff-bulk
drwxr-xr-x   13 root root  4096 Mar 16 15:16 staff-groups
drwxr-xr-x   28 root root  4096 Mar 17 11:18 staff-homes
drwxr-xr-x   28 root root  4096 Mar 17 11:18 staff-homes-linux
drwxr-xr-x 1674 root root 65536 Mar 17 11:18 staff-umbrella
drwxr-xr-x   12 root root  4096 Mar 17 11:18 student-groups
drwxr-xr-x   28 root root  4096 Mar 17 11:18 student-homes
drwxr-xr-x   28 root root  4096 Mar 17 11:18 student-homes-linux
```

TU Delft

File transfer nodes

```
[NetID@login02 ~]$ sinfo

PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
compute*     up   infinite    226   idle cmp[001-218]
gpu          up   infinite     10   idle gpu[001-010]
memory       up   infinite     10   idle mem[001-010]
trans        up   infinite      2   idle file[01-02]
visual       up   infinite      2   idle visual[01-02]
```

```
#!/bin/sh
#
#SBATCH --job-name="js_data"
#SBATCH --partition=trans
#SBATCH --time=01:00:00
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=1

rsync -av --no-perms /tudelft.net/staff-umbrella/<my-project-name>/data_folder
/scratch/<netid>/
```

TU Delft

Exercise 101

ssh to DelftBlue

```
user@laptop:~ $ ssh NetID@login.delftblue.tudelft.nl
```

copy (rsync) /scratch/dpalagin/DelftBlueWorkshop/ to your own /scratch

```
NetID@login01:~ $ rsync -av /scratch/dpalagin/DelftBlueWorkshop /scratch/$USER/
```

TU Delft

Practical use of DelftBlue

- Recap: what is a cluster computer
- Accessing the system
- File systems and data transfer
- Queuing, accounting
- Module system (lmod)
- When (not to) use srun/mpirun/mpiexec
- Practical exercise

TU Delft

What is a scheduler?

```
34101180 kchoudhu    pr89 5vb2_ctd_unfol  PD Priority  Tomorr 20:15              0:00 1-00:00:00    16    192
34101393 kchoudhu    pr89 5vb2_ctd_unfol  PD Priority  Tomorr 20:15              0:00 1-00:00:00    16    192
34101545  ykarami   pr118         IgG1C1  PD Priority  Tomorr 20:15              0:00 1-00:00:00     6     72
34101588  ykarami   pr118       IgG3-1C1  PD Priority  Tomorr 20:15              0:00 1-00:00:00     6     72
34101854 mperisdi   pr107          Debug  PD Priority  Tomorr 20:15              0:00 1-00:00:00     3      3
34102044  calleva    pr89           4q4g  PD Priority  Tomorr 20:15              0:00 1-00:00:00     8     96
34102045  calleva    pr89           4g4q  PD Priority  Tomorr 20:15              0:00 1-00:00:00     8     96
34102780 asridhar    pr89         Kv3_WT  PD Priority  Tomorr 20:15              0:00 1-00:00:00    12    144
34102782 asridhar    pr89       Kv3_WT_2  PD Priority  Tomorr 20:15              0:00 1-00:00:00    12    144
34102783 asridhar    pr89      D120_V253  PD Priority  Tomorr 20:15              0:00 1-00:00:00     8     96
34102784 asridhar    pr89    D120_V253_2  PD Priority  Tomorr 20:15              0:00 1-00:00:00     8     96
34102785 asridhar    pr89          F256A  PD Priority  Tomorr 20:15              0:00 1-00:00:00     8     96
34102786 asridhar    pr89        F256A_2  PD Priority  Tomorr 20:15              0:00 1-00:00:00     8     96
34102787 asridhar    pr89          gaba5  PD Priority  Tomorr 20:15              0:00 1-00:00:00    12    144
34102788 asridhar    pr89          gaba4  PD Priority  Tomorr 20:15              0:00 1-00:00:00    12    144
34102792 asridhar    pr89          gaba6  PD Priority  Tomorr 20:15              0:00 1-00:00:00    12    144
34102827 mbiliche    pr66     Gkclp5_825  PD Priority  Tomorr 20:15              0:00 1-00:00:00    25    300
34102868 mbiliche    pr66     Gkclp4_866  PD Priority  Tomorr 20:15              0:00 1-00:00:00    25    300
34102902 mbiliche    pr66     Gkclp3_854  PD Priority  Tomorr 20:15              0:00 1-00:00:00    25    300
34103693 kchoudhu    pr89 5vb2_ctd_unfol  PD Priority  Tomorr 20:15              0:00 1-00:00:00    16    192
34105695 akumawat   pr117 ICA-SOFT-AB2.1  PD Priority  Tomorr 20:15              0:00 1-00:00:00     8     96
34105696 akumawat   pr117 ICA-SOFT-AB2.2  PD Priority  Tomorr 20:15              0:00 1-00:00:00     8     96
34105700 akumawat   pr117 ICA-SOFT-AB2.3  PD Priority  Tomorr 20:15              0:00 1-00:00:00     8     96
34105736 akumawat   pr117  MUT-NSc-AB2.9  PD Priority  Tomorr 20:15              0:00 1-00:00:00     8     96
34105739 akumawat   pr117 MUT-NSc-AB2.10  PD Priority  Tomorr 20:15              0:00 1-00:00:00     8     96
34105795  ykarami   pr118       IgG3-2C1  PD Priority  Tomorr 20:15              0:00 1-00:00:00     6     72
34105821 akumawat   pr117    AMB19-AB2.1  PD Priority  Tomorr 20:15              0:00 1-00:00:00     8     96
34108696  ykarami   pr118           MTDP  PD Priority  Tomorr 20:15              0:00 1-00:00:00   100   1200
34108877  ykarami   pr118       IgG3-1C2  PD Priority  Tomorr 20:15              0:00 1-00:00:00     6     72
34111511  sfurini   pr107            e4f  PD Priority  Tomorr 20:15              0:00 1-00:00:00     8    192
34087143  ykarami   pr118         IgG1B3  CG None      Ystday 12:28          20:54:35    3:05:25     1    144
dpalagin@daint105:~>
```

TU Delft

How do I work with cluster?

1. **Prepare input files for your code** on a personal computer
2. **Upload input files** and required data to the cluster's storage
3. **Determine** required **resources**
4. **Create job script**
5. **Submit** job(s) to scheduler
6. **Monitor** progress (via output files) and resource use (via statistics)
7. **Download** results to personal computer for further processing
8. **Cleanup** files

Typical commands?

- Create a job script in a file

```
#!/bin/sh

#SBATCH --job-name=job_name
#SBATCH --partition=compute
#SBATCH --account=research-eemcs-diam
#SBATCH --time=01:00:00
#SBATCH --ntasks=24
#SBATCH --cpus-per-task=1
#SBATCH --mem-per-cpu=1GB

module load 2022r2
module load openmpi
srun ./executable > output.log
```

```
➜ select name
➜ select partition
➜ specify account
➜ request run time
➜ number of tasks (parallel)
➜ CPUs (threads) per task
➜ request memory

➜ Use DelftBlue software collection
➜ load openmpi module
➜ start tasks with srun
```

TU Delft

Typical commands?

- Log in to one of the login nodes
- Submit job using the job script

```
$ sbatch jobscript.sbatch
Submitted batch job 1
```

- See queue status

```
$ squeue
JOBID PARTITION     NAME          USER ST   TIME

1      general    jobscrip   somebody  R   0:01
```

- See job output

```
$ cat slurm-1.out
Hello world!
```

- Cancel job

```
$ scancel 1
$ squeue
JOBID PARTITION     NAME          USER ST   TIME
NODES NODELIST(REASON)
```

TU Delft

What do I need to know about accounting?

```
#!/bin/sh

#SBATCH --account=innovation


#SBATCH --account=research-eemcs-diam


#SBATCH --account=education-eemcs-msc
```

TU Delft

Example 1.1: Hello, World! on 8 CPUs

Our first submission script `helloworld.sh`:

```
#!/bin/bash
#SBATCH --job-name="01_hello"
#SBATCH --time=00:10:00
#SBATCH --ntasks=8
#SBATCH --cpus-per-task=1
#SBATCH --partition=compute
#SBATCH --mem=1GB
#SBATCH --account=innovation
#SBATCH –-reservation=db-workshop

cd $SLURM_SUBMIT_DIR

echo "Hello, World!" >> helloworld.txt
echo "The following nodes are reporting for duty:" >> helloworld.txt
srun hostname >> helloworld.txt
echo "Have a great day!" >> helloworld.txt
```

TU Delft

Example 1.1: Hello, World! on 8 CPUs

```
NetID@login01:~ $ sbatch helloworld.sh
```

```
Hello, World!
The following nodes are reporting for duty:
cmpXXX
cmpXXX
cmpXXX
cmpXXX
cmpXXX
cmpXXX
cmpXXX
cmpXXX
cmpXXX
Have a great day!
```

TU Delft

Example 1.2: Hello, World!

Our submission script `helloworld2nodes.sh`:

```
#!/bin/bash
#SBATCH --job-name="01_hello"
#SBATCH --time=00:10:00
#SBATCH --nodes=2
#SBATCH --ntasks-per-node=4
#SBATCH --cpus-per-task=1
#SBATCH --partition=compute
#SBATCH --mem=1GB
#SBATCH --account=innovation
#SBATCH --reservation=db-workshop

cd $SLURM_SUBMIT_DIR

echo "Hello, World!" >> helloworld.txt
echo "The following nodes are reporting for duty:" >> helloworld.txt
srun hostname >> helloworld.txt
echo "Have a great day!" >> helloworld.txt
```

TU Delft

Example 1.2: Hello, World!

```
NetID@login01:~ $ sbatch helloworld2nodes.sh
```

```
Hello, World!
The following nodes are reporting for duty:
cmpXXX
cmpXXX
cmpXXX
cmpXXX
cmpYYY
cmpYYY
cmpYYY
cmpYYY
Have a great day!
```

TU Delft

Practical use of DelftBlue

- Recap: what is a cluster computer
- Accessing the system
- File systems and data transfer
- Queuing, accounting
- Module system (lmod)
- When (not to) use srun/mpirun/mpiexec
- Practical exercise

Module system demo

```
[NetID@login02 ~]$ module avail                  ➔ list available modules
[NetID@login02 ~]$ module load                   ➔ load module
[NetID@login02 ~]$ module unload                 ➔ unload module
[NetID@login02 ~]$ module list                   ➔ list loaded modules
[NetID@login02 ~]$ module spider {module}        ➔ find module {module}
```

Trilinos example: https://gitlab.tudelft.nl/dhpc/docs/-/wikis/DHPC-modules

TU Delft

Example 2: Data processing with R

Let's prepare the R script `generate.R` to generate normal data sets:

```
options(bitmapType='cairo')

arg = commandArgs(TRUE)

samples = rep(NA, 100000)

for ( i in 1:100000 )  { samples[i] = mean(rexp(40, 0.2)) }

pdf(paste('plots/', arg, '.pdf', sep=""))
hist(samples, main="", prob=T, color="darkred")
lines(density(samples), col="darkblue", lwd=3)
dev.off()
```

Example 2: Data processing with R

Now let's prepare the submission script `R_submit.sh`:

```
#!/bin/bash
#SBATCH --job-name="02_R_hist"
#SBATCH --time=00:10:00
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=1
#SBATCH --partition=compute
#SBATCH --mem=1GB
#SBATCH --account=innovation
#SBATCH –-reservation=db-workshop

cd $SLURM_SUBMIT_DIR

mkdir plots

srun R --vanilla -f generate.R --args "plot$SLURM_ARRAY_TASK_ID"
```

Example 2: Data processing with R

Finally, let's run the job:

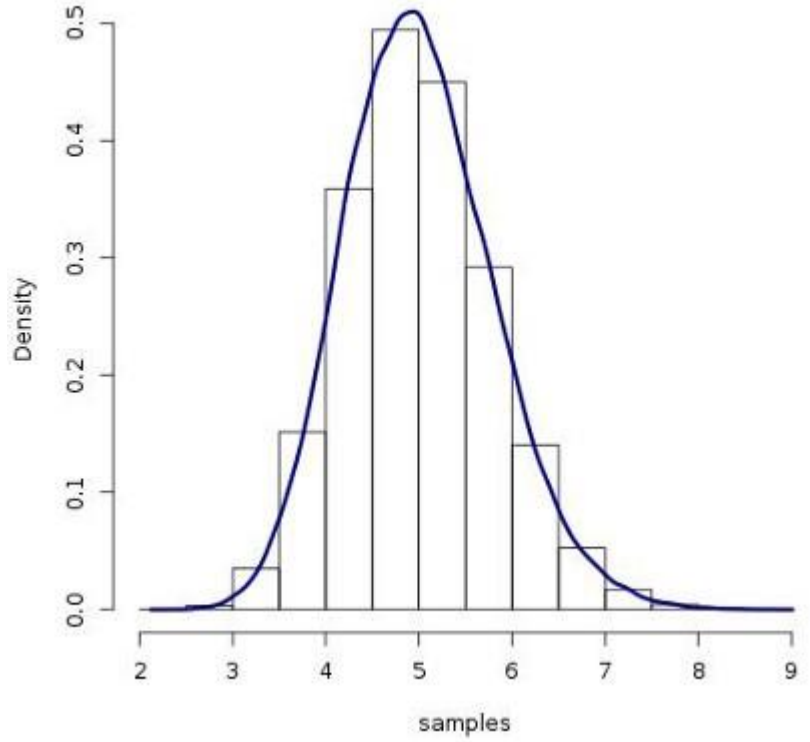
```
NetID@login01 :~ $ sbatch --array=[1-8] R_submit.sh
```

And monitor the queue:

```
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
       354_[11-50] mycluster R_submit       pi PD       0:00      1 (Resources)
             354_9 mycluster R_submit       pi  R       0:02      1 cnat
            354_10 mycluster R_submit       pi  R       0:02      1 cnat
             354_1 mycluster R_submit       pi  R       0:18      1 p1
             354_2 mycluster R_submit       pi  R       0:18      1 p2
             354_3 mycluster R_submit       pi  R       0:18      1 p3
             354_4 mycluster R_submit       pi  R       0:18      1 p4
```

TU Delft

Example 2: Data processing with R



TU Delft

Practical use of DelftBlue

- Recap: what is a cluster computer
- Accessing the system
- File systems and data transfer
- Queuing, accounting
- Module system (lmod)
- When (not to) use srun/mpirun/mpiexec
- Practical exercise

TU Delft

When (not to) use srun/mpirun/mpiexec

Always use **srun** with parallel executables relying on mprun/mpiexec

Practical use of DelftBlue

- Recap: what is a cluster computer
- Accessing the system
- File systems and data transfer
- Queuing, accounting
- Module system (lmod)
- When (not to) use srun/mpirun/mpiexec
- Practical exercise

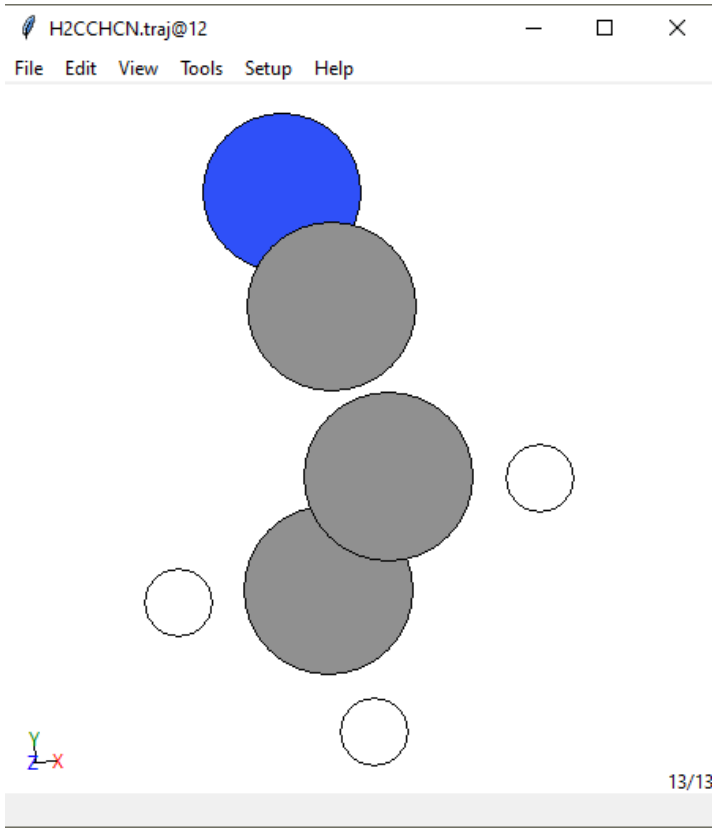
TU Delft

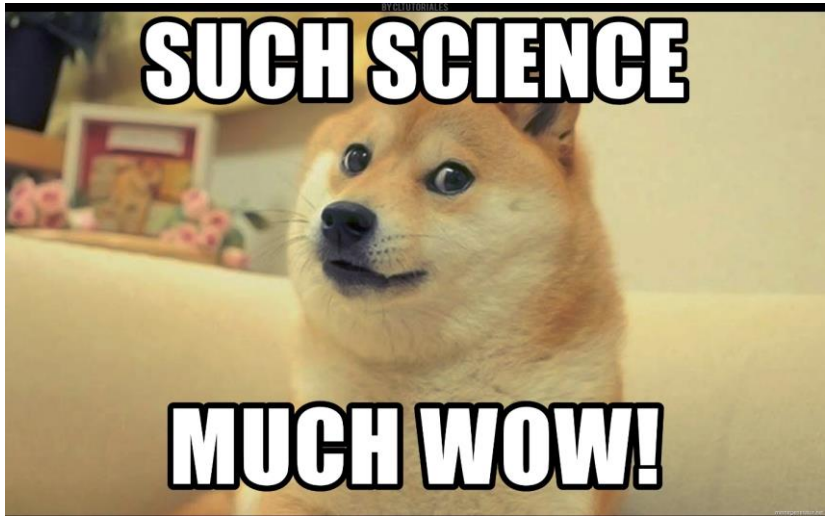
ASE molecules generator

```
[NetID@login02 ~]$ ./install_ase.sh

[NetID@login02 ~]$ sbatch sub_to_queue.sh

[NetID@login02 ~]$ cat slurm-XXX.out
```

Exercise:

1. Submit the job. Inspect the output file and generated folders.
2. rsync resulting folders to your staff-home.
3. Submit the job again. Can you see what is different in the output now?
4. rsync resulting folders to your staff-home once again.





TU Delft

Discussion and questions

Thanks for your attention

TU Delft