Introduction to DelftBlue

Dennis Palagin
20 September 2024

Practical use of DelftBlue

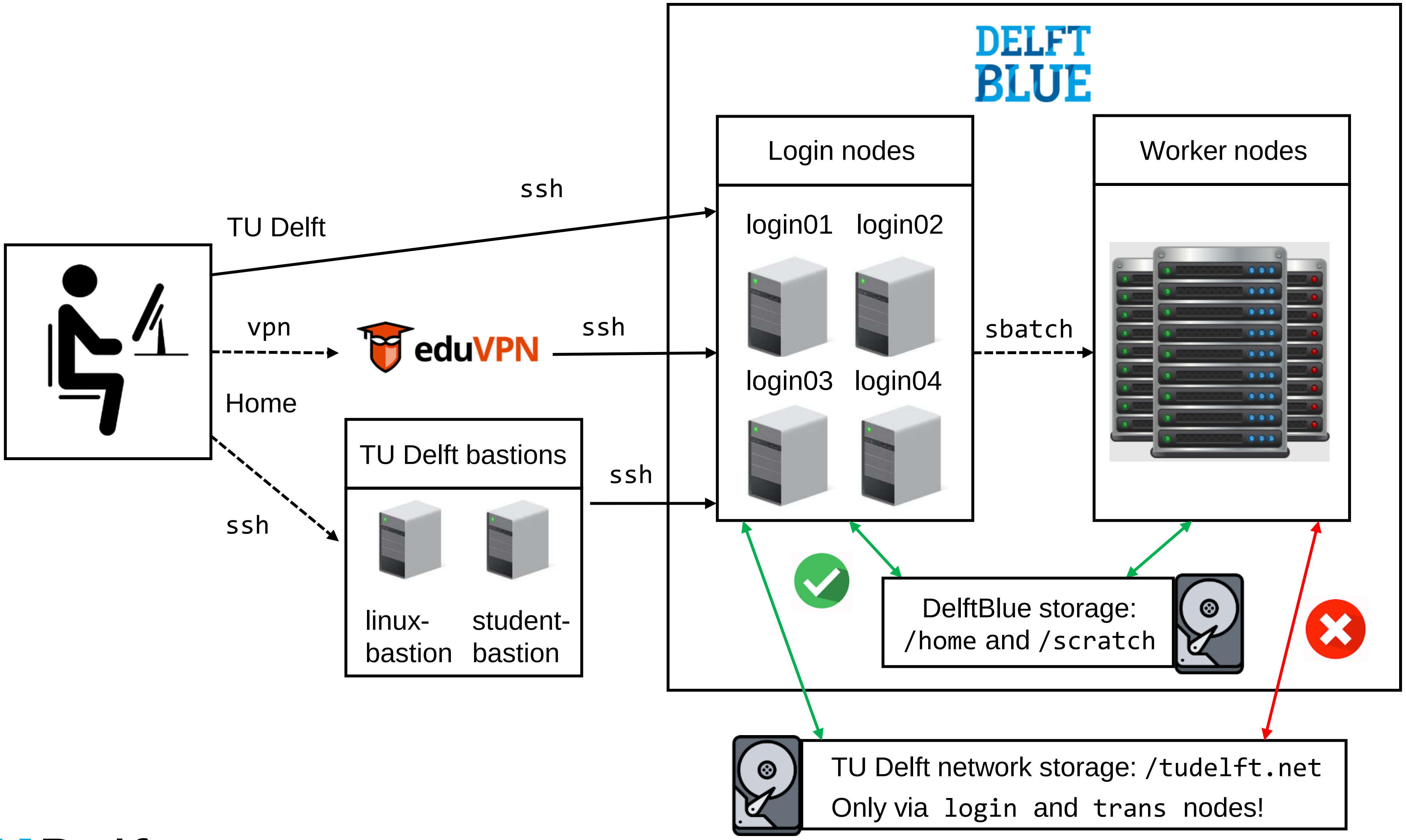
- Recap: what is a cluster computer
- DHPC and DelftBlue: organization
- Accessing the system
- File systems and data transfer
- Queuing, accounting
- Module system (lmod)

TU Delft

Practical use of DelftBlue

- Recap: what is a cluster computer
- DHPC and DelftBlue: organization
- Accessing the system
- File systems and data transfer
- Queuing, accounting
- Module system (lmod)

TUDelft


DELFT BLUE
Login nodes
Worker nodes
TU Delft
ssh
vpn
eduVPN
ssh
Home
ssh
TU Delft bastions
ssh
login01
login02
login03
login04
sbatch
linux-bastion
student-bastion
DelftBlue storage: /home and /scratch
TU Delft network storage: /tudelft.net
Only via login and trans nodes!

DeltfBlue

- Fast and flexible
- 16.000 CPU cores
- 20 GPU nodes
- High-Speed Interconnect based on Mellanox InfiniBand
- And a 700TB high-speed parallel storage subsystem

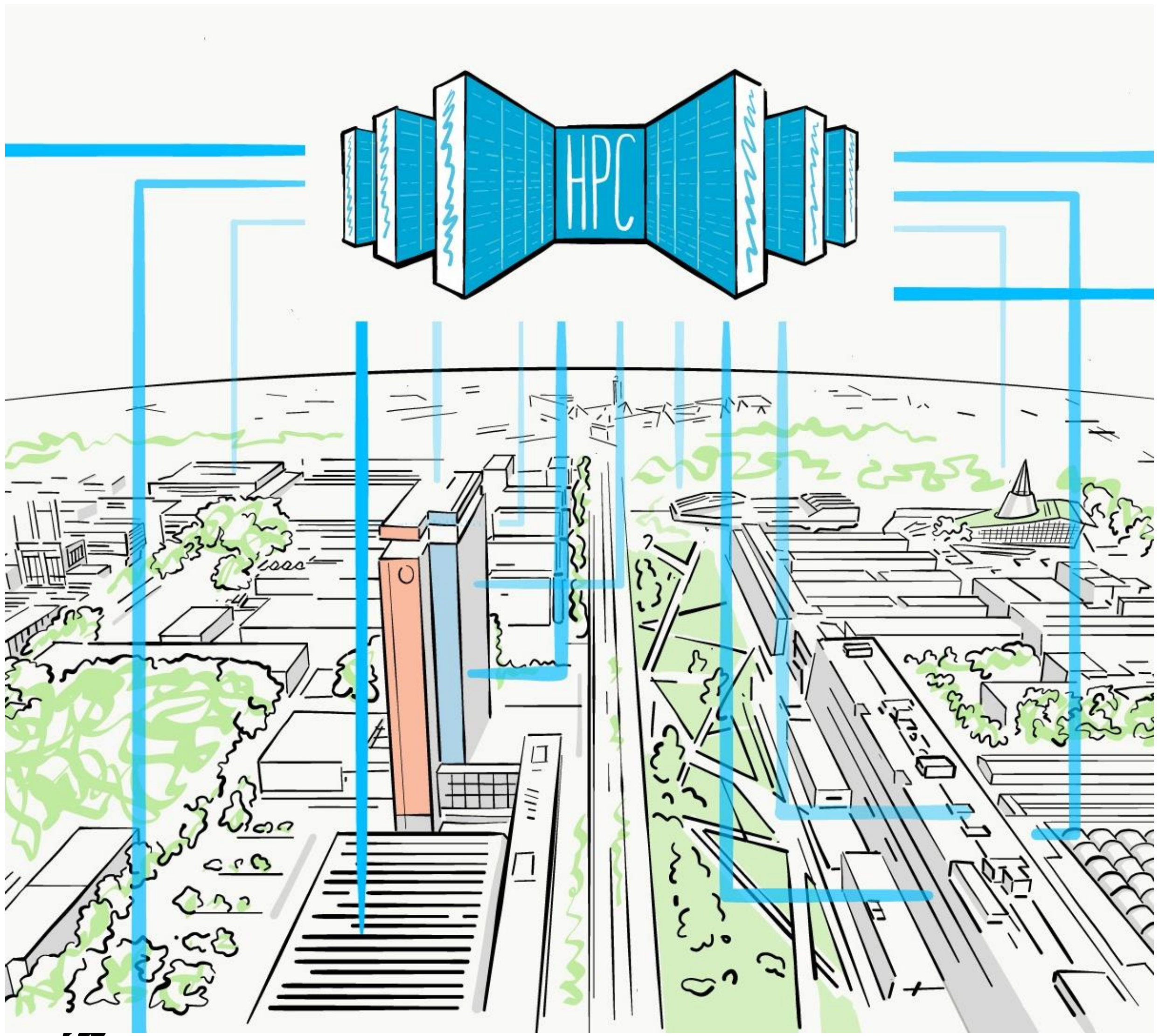
TUDelft

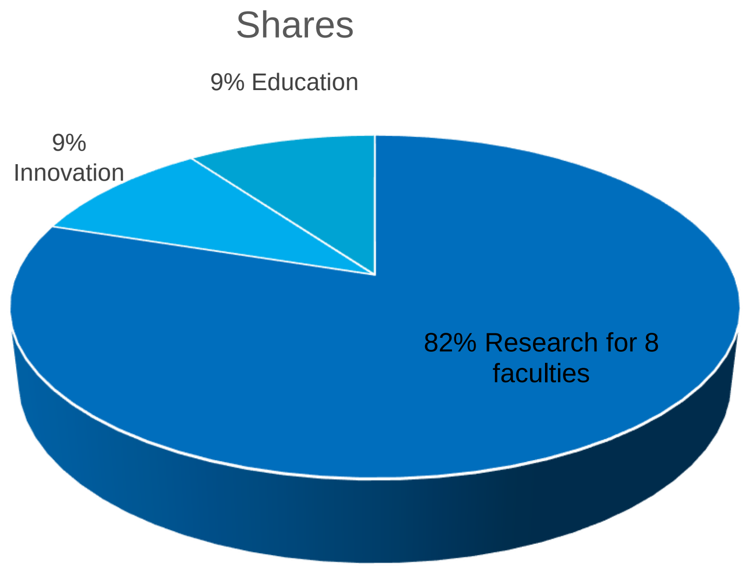
Practical use of DelftBlue

- Recap: what is a cluster computer
- DHPC and DelftBlue: organization
- Accessing the system
- File systems and data transfer
- Queuing, accounting
- Module system (lmod)

TUDelft

DelftBlue community: 3mE, ABE, AE, AS, CEG, EEMCS, IDE, TPM, QuTech





Research AE/LR	12%
Research EEMCS/EWI	12%
Research TPM/TBM	12%
Research CEG/CiTG	12%
Research 3mE	12%
Research AS/TNW	12%
Research ABE/BK	6%
Research IDE/IO	2%
Research Qutech	2%
Education	9%
Innovation	9%
	100%

TU Delft

Documentation:

- Documentation:

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

- Mattermost:

  https://mattermost.tudelft.nl/dhpc/

- Self Service Portal (TopDesk):

  https://tudelft.topdesk.net

- Training and courses:

  https://www.tudelft.nl/cse/education/courses

TU Delft

Practical use of DelftBlue

- Recap: what is a cluster computer
- DHPC and DelftBlue: organization
- Accessing the system
- File systems and data transfer
- Queuing, accounting
- Module system (lmod)

TUDelft

ssh

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

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

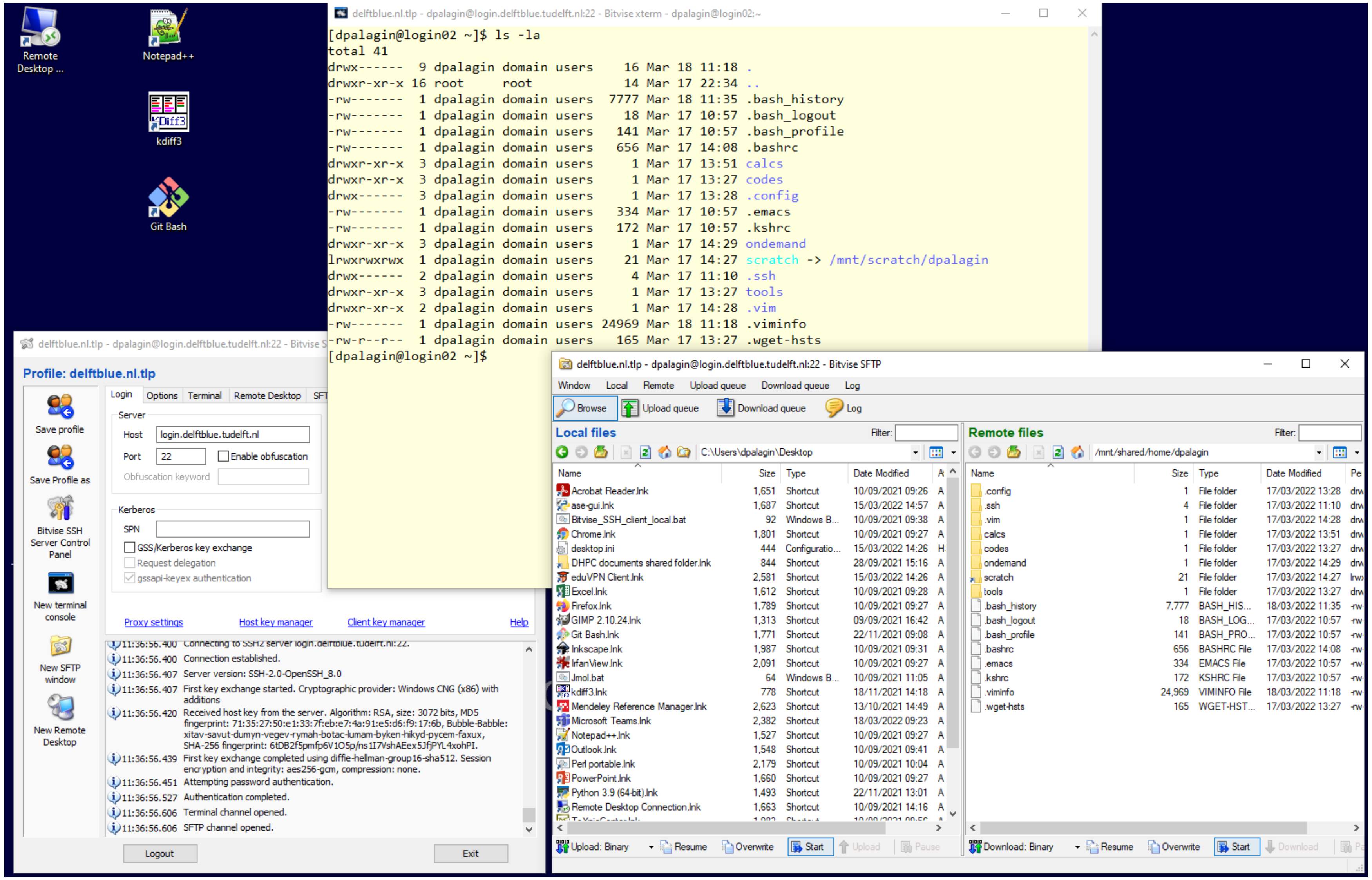
This will log you in into one of the four login nodes (login01, login02, login03, or login04). Your home directory is directly accessible (via /home/<netid>/):

```
[<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
```

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



```
[dpalagin@login02 ~]$ ls -la
total 41
drwx------  9 dpalagin domain users    16 Mar 18 11:18 .
drwxr-xr-x 16 root     root            14 Mar 17 22:34 ..
-rw-------  1 dpalagin domain users  7777 Mar 18 11:35 .bash_history
-rw-------  1 dpalagin domain users    18 Mar 17 10:57 .bash_logout
-rw-------  1 dpalagin domain users   141 Mar 17 10:57 .bash_profile
-rw-------  1 dpalagin domain users   656 Mar 17 14:08 .bashrc
drwxr-xr-x  3 dpalagin domain users     1 Mar 17 13:51 calcs
drwxr-xr-x  3 dpalagin domain users     1 Mar 17 13:27 codes
drwx------  3 dpalagin domain users     1 Mar 17 13:28 .config
-rw-------  1 dpalagin domain users   334 Mar 17 10:57 .emacs
-rw-------  1 dpalagin domain users   172 Mar 17 10:57 .kshrc
drwxr-xr-x  3 dpalagin domain users     1 Mar 17 14:29 ondemand
lrwxrwxrwx  1 dpalagin domain users    21 Mar 17 14:27 scratch -> /mnt/scratch/dpalagin
drwx------  2 dpalagin domain users     4 Mar 17 11:10 .ssh
drwxr-xr-x  3 dpalagin domain users     1 Mar 17 13:27 tools
drwxr-xr-x  2 dpalagin domain users     1 Mar 17 14:28 .vim
-rw-------  1 dpalagin domain users 24969 Mar 18 11:18 .viminfo
-rw-r--r--  1 dpalagin domain users   165 Mar 17 13:27 .wget-hsts
[dpalagin@login02 ~]$
```

delftblue.nl.tlp - dpalagin@login.delftblue.tudelft.nl:22 - Bitvise S

Profile: delftblue.nl.tlp

Server

Host	login.delftblue.tudelft.nl
Port	22
Enable obfuscation	☐
Obfuscation keyword	

Kerberos

SPN	
GSS/Kerberos key exchange	☐
Request delegation	☐
gssapi-keyex authentication	✓

```
11:36:56.400 Connecting to SSH2 server login.delftblue.tudelft.nl:22.
11:36:56.400 Connection established.
11:36:56.407 Server version: SSH-2.0-OpenSSH_8.0
11:36:56.407 First key exchange started. Cryptographic provider: Windows CNG (x86) with
             additions
11:36:56.420 Received host key from the server. Algorithm: RSA, size: 3072 bits, MD5
             fingerprint: 71:35:27:50:e1:33:7f:eb:e7:4a:91:e5:d6:f9:17:6b, Bubble-Babble:
             xitav-savut-dumyn-vegev-rymah-botac-lumam-byken-hikyd-pycem-faxux,
             SHA-256 fingerprint: 6tDB2f5pmfp6V1O5p/ns1I7VshAEex5JfjPYL4xohPI.
11:36:56.439 First key exchange completed using diffie-hellman-group16-sha512. Session
             encryption and integrity: aes256-gcm, compression: none.
11:36:56.451 Attempting password authentication.
11:36:56.527 Authentication completed.
11:36:56.606 Terminal channel opened.
11:36:56.606 SFTP channel opened.
```

Local files

C:\Users\dpalagin\Desktop

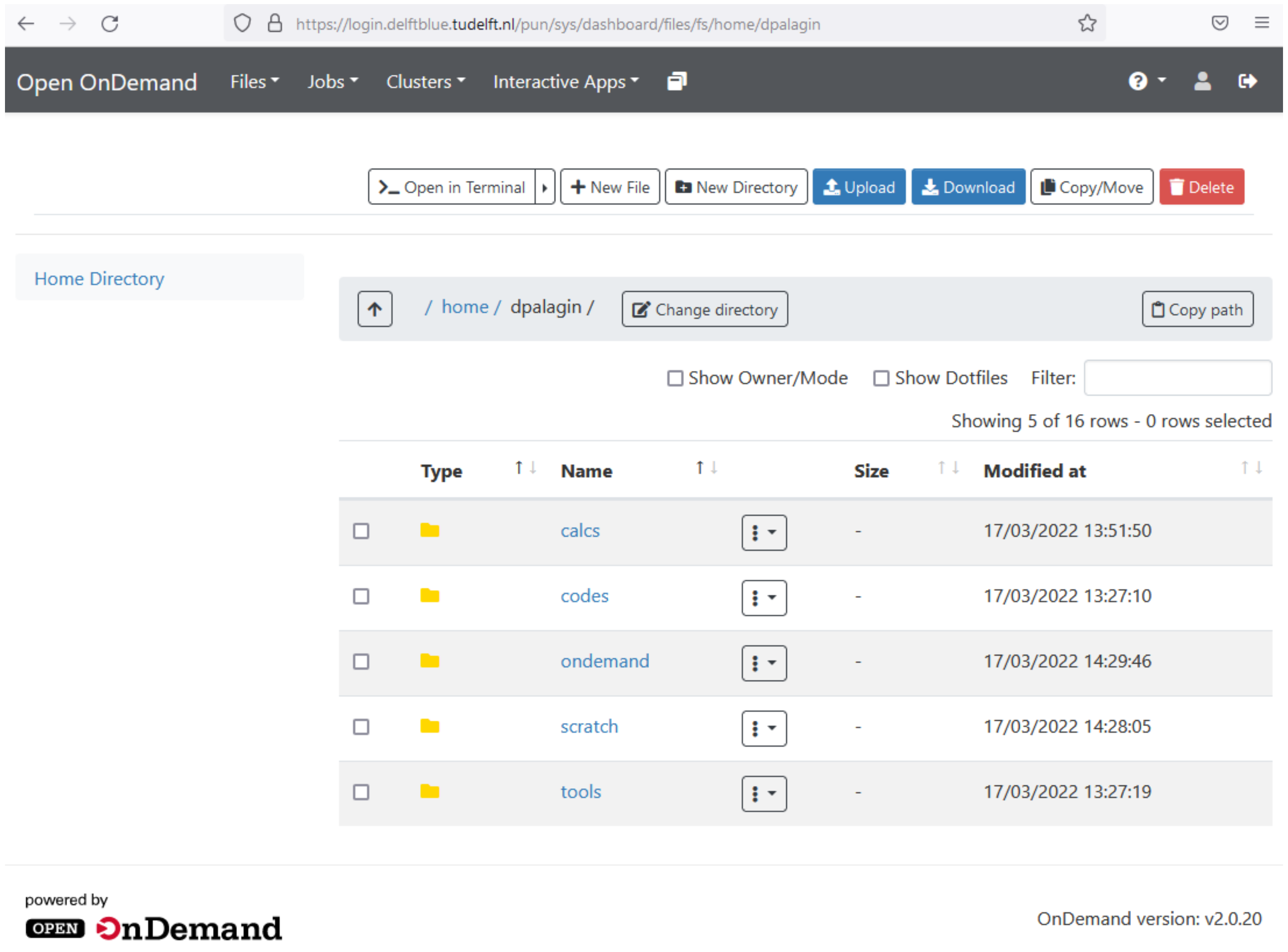
Name	Size	Type	Date Modified	A
Acrobat Reader.lnk	1,651	Shortcut	10/09/2021 09:26	A
ase-gui.lnk	1,687	Shortcut	15/03/2022 14:57	A
Bitvise_SSH_client_local.bat	92	Windows B...	10/09/2021 09:38	A
Chrome.lnk	1,801	Shortcut	10/09/2021 09:27	A
desktop.ini	444	Configuratio...	15/03/2022 14:26	H
DHPC documents shared folder.lnk	844	Shortcut	28/09/2021 15:16	A
eduVPN Client.lnk	2,581	Shortcut	15/03/2022 14:26	A
Excel.lnk	1,612	Shortcut	10/09/2021 09:28	A
Firefox.lnk	1,789	Shortcut	10/09/2021 09:27	A
GIMP 2.10.24.lnk	1,313	Shortcut	09/09/2021 16:42	A
Git Bash.lnk	1,771	Shortcut	22/11/2021 09:08	A
Inkscape.lnk	1,987	Shortcut	10/09/2021 09:31	A
IrfanView.lnk	2,091	Shortcut	10/09/2021 09:27	A
Jmol.bat	64	Windows B...	10/09/2021 11:05	A
kdiff3.lnk	778	Shortcut	18/11/2021 14:18	A
Mendeley Reference Manager.lnk	2,623	Shortcut	13/10/2021 14:49	A
Microsoft Teams.lnk	2,382	Shortcut	18/03/2022 09:23	A
Notepad++.lnk	1,527	Shortcut	10/09/2021 09:27	A
Outlook.lnk	1,548	Shortcut	10/09/2021 09:41	A
Perl portable.lnk	2,179	Shortcut	10/09/2021 10:04	A
PowerPoint.lnk	1,660	Shortcut	10/09/2021 09:27	A
Python 3.9 (64-bit).lnk	1,493	Shortcut	22/11/2021 13:01	A
Remote Desktop Connection.lnk	1,663	Shortcut	10/09/2021 14:16	A

Remote files

/mnt/shared/home/dpalagin

Name	Size	Type	Date Modified	Pe
.config	1	File folder	17/03/2022 13:28	drw
.ssh	4	File folder	17/03/2022 11:10	drw
.vim	1	File folder	17/03/2022 14:28	drw
calcs	1	File folder	17/03/2022 13:51	drw
codes	1	File folder	17/03/2022 13:27	drw
ondemand	1	File folder	17/03/2022 14:29	drw
scratch	21	File folder	17/03/2022 14:27	lrwx
tools	1	File folder	17/03/2022 13:27	drw
.bash_history	7,777	BASH_HIS...	18/03/2022 11:35	-rw-
.bash_logout	18	BASH_LOG...	17/03/2022 10:57	-rw-
.bash_profile	141	BASH_PRO...	17/03/2022 10:57	-rw-
.bashrc	656	BASHRC File	17/03/2022 14:08	-rw-
.emacs	334	EMACS File	17/03/2022 10:57	-rw-
.kshrc	172	KSHRC File	17/03/2022 10:57	-rw-
.viminfo	24,969	VIMINFO File	18/03/2022 11:18	-rw-
.wget-hsts	165	WGET-HST...	17/03/2022 13:27	-rw-

Web tools: OpenOnDemand



Showing 5 of 16 rows - 0 rows selected

Type	Name	Size	Modified at
📁	calcs	-	17/03/2022 13:51:50
📁	codes	-	17/03/2022 13:27:10
📁	ondemand	-	17/03/2022 14:29:46
📁	scratch	-	17/03/2022 14:28:05
📁	tools	-	17/03/2022 13:27:19

OnDemand version: v2.0.20

Practical use of DelftBlue

- Recap: what is a cluster computer
- DHPC and DelftBlue: organization
- Accessing the system
- File systems and data transfer
- Queuing, accounting
- Module system (lmod)

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 ./
```

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
```

Exercise 101

- ssh to DelftBlue

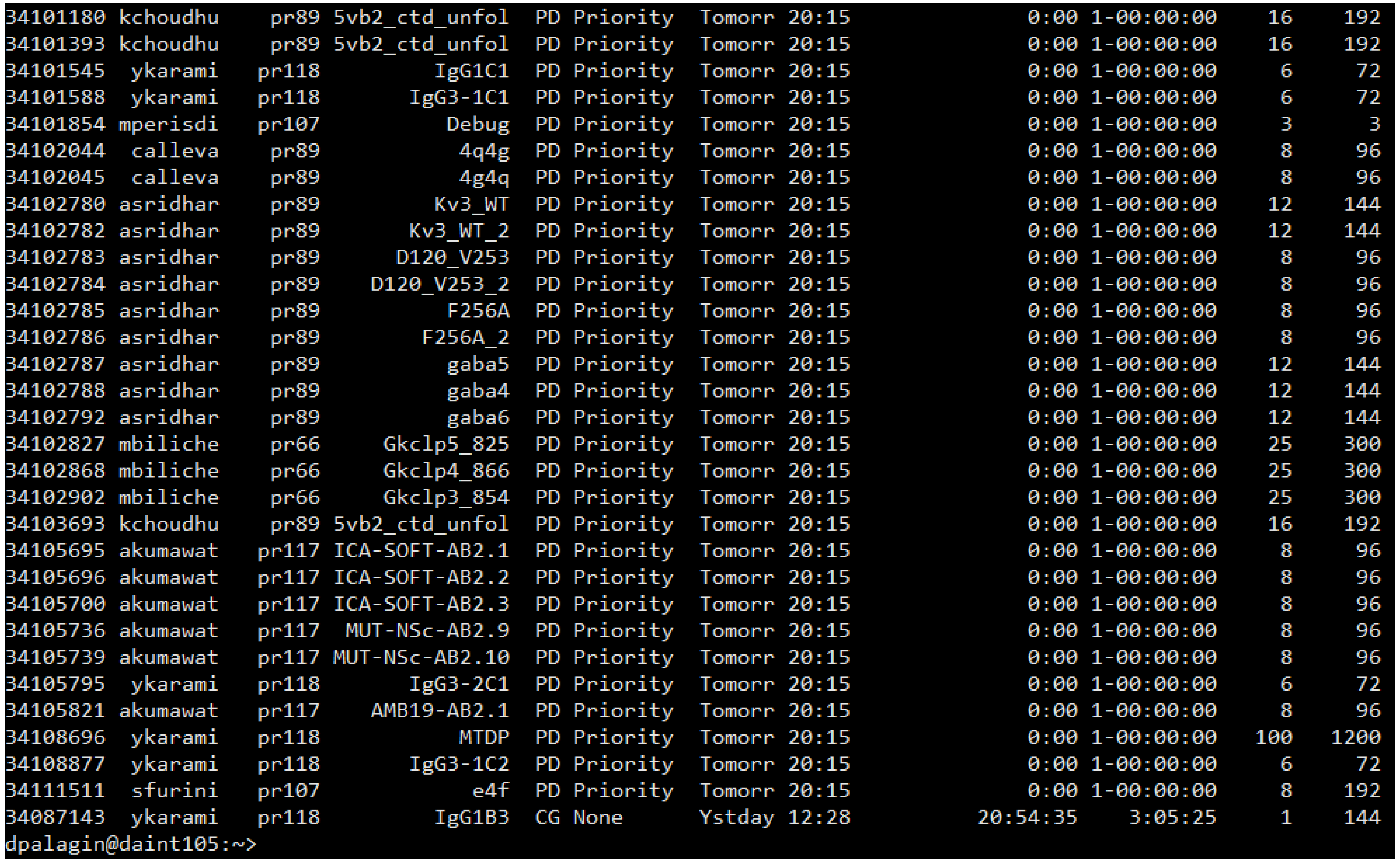
```
user@laptop:~ $ ssh NetID@login.delftblue.tudelft.nl
```

- Download exercises from https://doc.dhpc.tudelft.nl/delftblue/
- Copy to DelftBlue and unzip.

Practical use of DelftBlue

- Recap: what is a cluster computer
- DHPC and DelftBlue: organization
- Accessing the system
- File systems and data transfer
- Queuing, accounting
- Module system (lmod)

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:~>
```

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

TU Delft

Typical commands?

- Create a job script in a file

```
#!/bin/sh                                          ➔ system info: type of script

#SBATCH --job-name=job_name                        ➔ select job name
#SBATCH --partition=compute                        ➔ select partition to run your job on
#SBATCH --account=research-eemcs-diam              ➔ specify account
#SBATCH --time=01:00:00                            ➔ request run time
#SBATCH --ntasks=4                                 ➔ number of tasks (parallel)
#SBATCH --cpus-per-task=1                          ➔ CPUs (threads) per task
#SBATCH --mem-per-cpu=1G                           ➔ RAM per CPU

module load 2023r1                                 ➔ Use DelftBlue software collection
module load openmpi                                ➔ load openmpi module
srun ./executable > output.log                     ➔ start tasks with srun
```

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

Example 1: Hello, World! on 4 CPUs

Our first submission script `helloworld.sh`:

```
#!/bin/bash
#SBATCH --job-name="01_hello"
#SBATCH --time=00:10:00
#SBATCH --ntasks=4
#SBATCH --cpus-per-task=1
#SBATCH --partition=compute
#SBATCH --mem=1GB
#SBATCH --account=Education-EEMCS-Courses-WI4049TU

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
```

Example 1: Hello, World! on 4 CPUs

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

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

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=2
#SBATCH --cpus-per-task=1
#SBATCH --partition=compute
#SBATCH --mem=1GB
#SBATCH --account=Education-EEMCS-Courses-WI4049TU

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
```

TUDelft

Example 1.2: Hello, World!

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

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

Practical use of DelftBlue

- Recap: what is a cluster computer
- DHPC and DelftBlue: organization
- Accessing the system
- File systems and data transfer
- Queuing, accounting
- Module system (lmod)

Module system demo

```
[NetID@login02 ~]$ module avail
[NetID@login02 ~]$ module load
[NetID@login02 ~]$ module unload
[NetID@login02 ~]$ module list
[NetID@login02 ~]$ module spider {module}
```

➔ `list available modules`

➔ `load module`

➔ `unload module`

➔ `list loaded modules`

➔ `find module {module}`

Trilinos example: https://doc.dhpc.tudelft.nl/delftblue/DHPC-modules/

Example 2: Julia

Let’s prepare a little program to draw a Mandelbrot set:

```
function mandelbrot(a)
    z = 0
    for i=1:50
        z = z^2 + a
    end
    return z
end

for y=1.0:-0.05:-1.0
    for x=-2.0:0.0315:0.5
        abs(mandelbrot(complex(x, y))) < 2 ? print("*")
: print(" ")
    end
    println()
end
```

Example 2: Julia

Submission script `run_julia_mandelbrot.sh`:

```
#!/bin/bash
#
#SBATCH --job-name="julia"
#SBATCH --time=00:10:00
#SBATCH --partition=compute
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=1
#SBATCH --mem-per-cpu=1G
#SBATCH --account=Education-EEMCS-Courses-WI4049TU

module load 2023r1
module load julia

srun julia mandelbrot.jl
```

Example 2: Julia

Let’s submit the job and check out the result:

```
                                                          **
                                                        ******
                                                      ********
                                                        ******
                                                     ******** **   *
                                             ***   ****************
                                             ************************  ***
                                             ****************************
                                          ******************************
                                           ******************************
                                        ************************************
                               *         **********************************
                          ** ***** *     **********************************
                          ***********   ************************************
                        ************** ************************************
                        ***************************************************
                    *****************************************************
***********************************************************************
                    *****************************************************
                        ***************************************************
                        ************** ************************************
                          ***********   ************************************
                          ** ***** *     **********************************
                               *         **********************************
                                        ************************************
                                           ******************************
                                          ******************************
                                             ****************************
                                             ************************  ***
                                             ***   ****************
                                                     ******** **   *
                                                        ******
                                                      ********
                                                        ******
                                                         **
```

Example 3: Python with MPI: mpi4py

Let’s prepare a little program to calculate number pi in Python called `calculate_pi.py`:

```
from mpi4py import MPI
from math   import pi as PI
from numpy  import array

def comp_pi(n, myrank=0, nprocs=1):
    h = 1.0 / n
    s = 0.0
    for i in range(myrank + 1, n + 1, nprocs):
        x = h * (i - 0.5)
        s += 4.0 / (1.0 + x**2)
    return s * h

def prn_pi(pi, PI):
    message = "pi is approximately %.16f, error is 
%.16f"
    print  (message % (pi, abs(pi - PI)))
```

Example 3: Python with MPI: mpi4py

```
comm = MPI.COMM_WORLD
nprocs = comm.Get_size()
myrank = comm.Get_rank()

n    = array(0, dtype=int)
pi   = array(0, dtype=float)
mypi = array(0, dtype=float)

if myrank == 0:
    _n = 20 # Enter the number of intervals
    n.fill(_n)
comm.Bcast([n, MPI.INT], root=0)
_mypi = comp_pi(n, myrank, nprocs)
mypi.fill(_mypi)
comm.Reduce([mypi, MPI.DOUBLE], [pi, MPI.DOUBLE],
            op=MPI.SUM, root=0)
if myrank == 0:
    prn_pi(pi, PI)
```

Example 3: Python with MPI: mpi4py

Submission script `sub_calc_pi.sh`:

```
#!/bin/bash
#
#SBATCH --job-name="Py_pi"
#SBATCH --time=00:10:00
#SBATCH --ntasks=4
#SBATCH --cpus-per-task=1
#SBATCH --partition=compute
#SBATCH --mem-per-cpu=1G
#SBATCH --account=Education-EEMCS-Courses-WI4049TU

module load 2023r1
module load openmpi
module load python
module load py-numpy
module load py-mpi4py

srun python calculate_pi.py
```

Example 3: Python with MPI: mpi4py

Let’s submit the job:

```
user@login01:~ $ sbatch sub_calc_pi.sh
```

And check out the result:

```
pi is approximately 3.1418009868930934, error is
0.0002083333033003
```

Example 4: ASE molecules generator

Let’s install a new python module, called ASE:

```
#!/bin/bash

# Install ASE:
module load 2023r1
module load python
module load py-pip
module load py-numpy
module load py-scipy
module load py-matplotlib

python -m pip install --user ase
```

```
[NetID@login02 ~]$ chmod +x install_ase.sh

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

Example 4: ASE molecules generator

Now we are able to run our script:

```
# Import modules:
import os
import sys
import subprocess
import ase
from ase.io import read,write
from ase.build import molecule
from ase.optimize import BFGS
from ase.calculators.emt import EMT

def main():
    # Print info:
    info = """
    This script does the following:
    1. Creates a parent directory "molecules".
    2. Reads an array of molecules and does the following steps for each molecule:
        3. Creates a sub-directory for current molecule.
        4. Generates .xyz file containing Cartesian coordinates of current molecule.
        5. Decides if to write or to append the .log file.
        6. Optimizes the initial geometry of current molecule with EMT and ...
        6. ... writes optimization .log and ASE trajectory files for current molecule.
    """
    print(info)
```

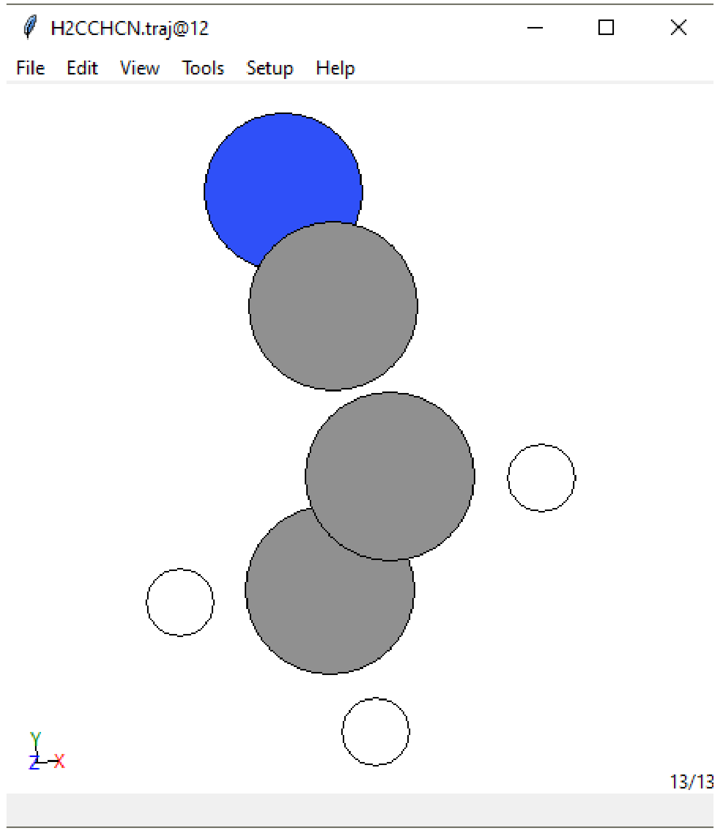
Example 4: ASE molecules generator

```
[NetID@login02 ~]$ sbatch sub_to_queue.sh

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

<u>Exercise:</u>

Install ASE. Submit the job. Inspect the output file and generated folders.



(Bonus) Example 5: PyTorch

Let’s prepare a little program to check how many GPUs are available to us:

```
import torch
cuda_avail = torch.cuda.is_available()
if cuda_avail:
        print("Torch CUDA is available")
        num_of_devices = torch.cuda.device_count()
        if num_of_devices:
            print("Number of CUDA devices: {}".format(num_of_devices))
            current_device = torch.cuda.current_device()
            current_device_id = torch.cuda.device(current_device)
            current_device_name = torch.cuda.get_device_name(current_device)
            print("Current device id: {}".format(current_device_id))
            print("Current device name: {}".format(current_device_name))
        else:
            print("No CUDA devices!")
else:
        print("Torch CUDA is not available!")
```

TU Delft

(Bonus) Example 5: PyTorch

Submission script `sub_pytorch_gpus.sh`:

```
#!/bin/bash
#
#SBATCH --job-name="PyTorch"
#SBATCH --time=00:10:00
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=1
#SBATCH –gpus-per-task=1
#SBATCH --partition=gpu
#SBATCH --mem=4G
#SBATCH --account=Education-EEMCS-Courses-WI4049TU

module load 2023r1
module load openmpi
module load py-torch

srun python test_pytorch_gpus.py
```

(Bonus) Example 5: PyTorch

Let’s submit the job and check out the result:

```
Torch CUDA is available
Number of CUDA devices: 1
Current device id: <torch.cuda.device object at 0x155555379dc0>
Current device name: Tesla V100S-PCIE-32GB
```

TUDelft

Discussion and questions

Thanks for your attention

TU Delft