

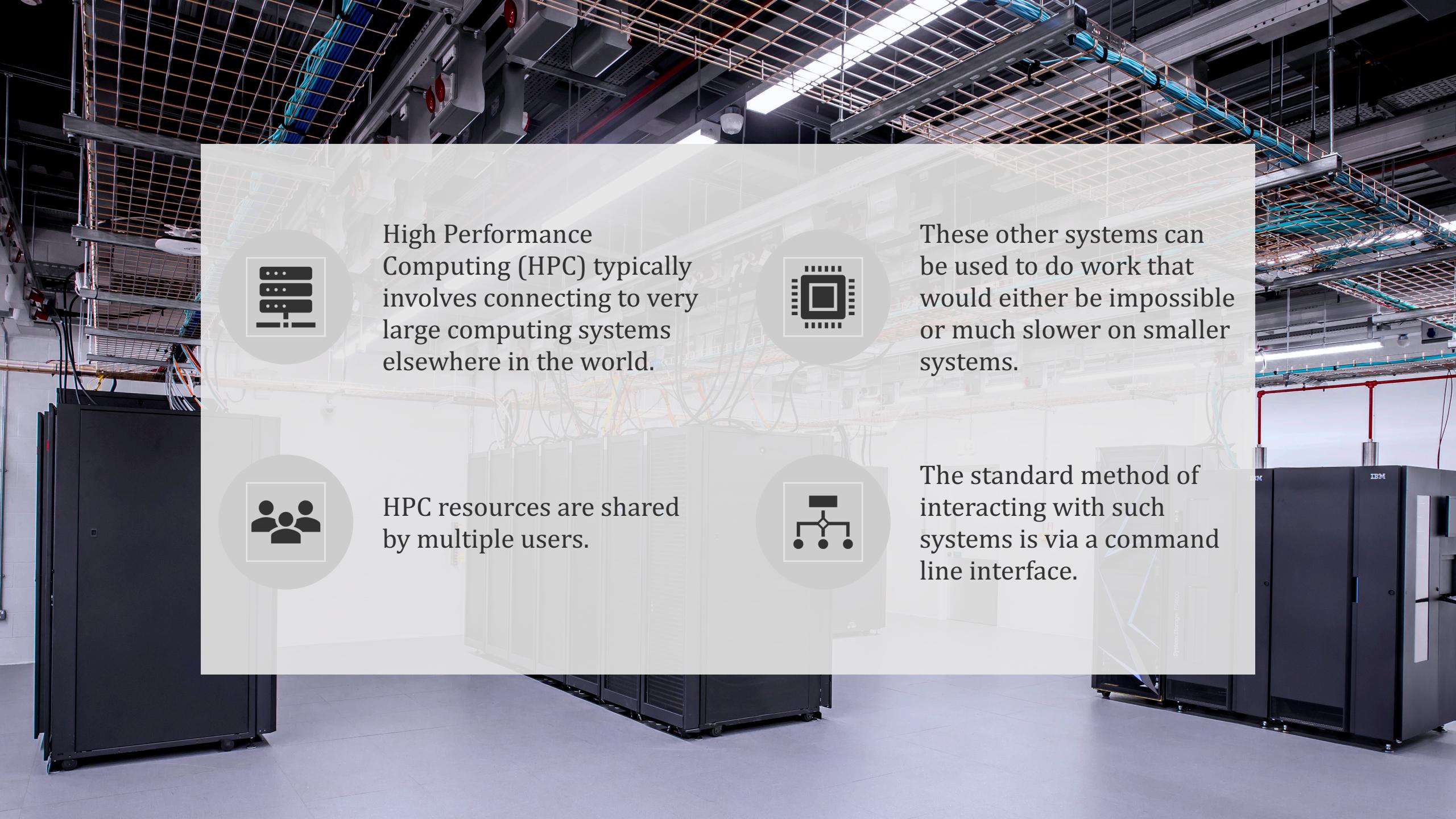
Intro to Baskerville

EPSRC Tier 2 HPC System
University of Birmingham

Overview

1. How can High Performance Computing benefit you?
2. Connecting to Baskerville
3. System Architecture
4. Running jobs using SLURM
5. Accessing software applications
6. Transferring files to/from Baskerville
7. Final Task: Running a job with GPUs

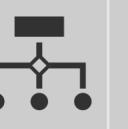
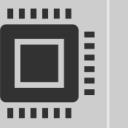
1. How can High Performance Computing benefit you?



High Performance Computing (HPC) typically involves connecting to very large computing systems elsewhere in the world.

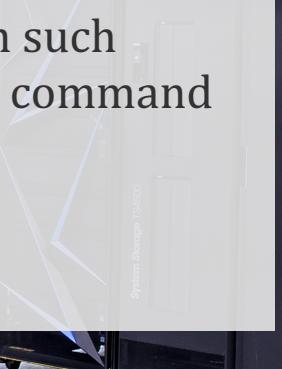


HPC resources are shared by multiple users.



These other systems can be used to do work that would either be impossible or much slower on smaller systems.

The standard method of interacting with such systems is via a command line interface.



2. Connecting to Baskerville

Connect using an SSH Client.

Windows: [PuTTY](#) | [MobaXterm](#)

MacOS or Linux: Terminal



```
$ ssh <USERNAME>@login.baskerville.ac.uk  
Password: <enter your password>  
OTP code: <enter your OTP>
```

Projects and QoS

Once you are connected to a login node, run



```
[username@bask-pg0310u16a ~]$ my_baskerville
```

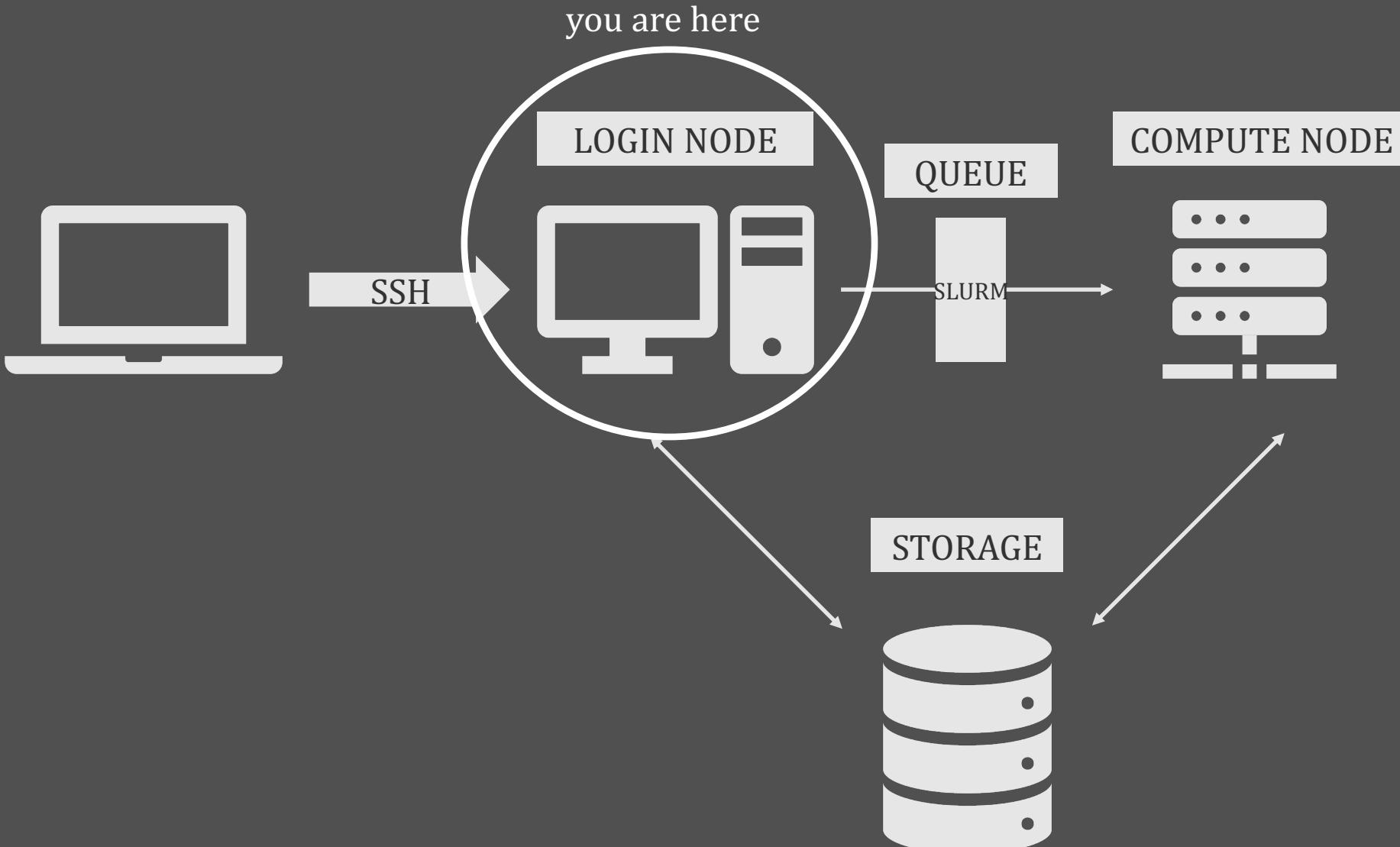
to see information about your Baskerville projects and available QoS.

This information is also available at admin.baskerville.ac.uk/.

3. System Architecture

Local: Your computer

Remote: Baskerville



Your home folder | 20GB



```
[username@bask-pg0310u16a ~]$ pwd  
/bask/homes/u/username  
[username@bask-pg0310u16a ~]$ my_quota  
Your home directory on Baskerville (/bask/homes  
/u/username):  
You are using 97.30 percent ( 19.48 GB ) of your total  
quota 20.02 GB
```

Your project folder | 1TB



```
[username@bask-pg0310u16a ~]$ cd /bask/projects/m/myproject  
[username@bask-pg0310u16a myproject]$ df -h .  
Filesystem      Size  Used Avail Use% Mounted on  
bask            1.0T   64M  1.0T   1% /bask
```

4. Running Jobs using SLURM

Choose your editor

[vim](#)



[nano](#)

```
...  
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
```



```
[username@bask-pg0310u16a myproject]$ vim example-job.sh
```

Create your job script



```
#!/bin/bash
#SBATCH --qos QOS
#SBATCH --account ACCOUNT
#SBATCH --time 1:0:0

module purge
module load baskerville

echo -n "This script is running on "
hostname
```

<ESC> :wq <ENTER> to write and quit vim

Submit your job script



```
[username@bask-pg0310u16a myproject]$ sbatch example-job.sh
```

Check your job status



```
[username@bask-pg0310u16a myproject]$ squeue -u $USER
      JOBID PARTITION      NAME      USER ST      TIME   NODES NODELIST(REASON)
      243982 baskervil example-    username R      0:02       1 bask-pg0309u03a
```

View your job outputs



```
[username@bask-pg0310u16a myproject]$ ls
example-job.sh  slurm-243982.out  slurm-243982.stats
[username@bask-pg0310u16a myproject]$ cat slurm-243982.out
This script is running on bask-pg0309u03a.cluster.baskerville.ac.uk
[username@bask-pg0310u16a myproject]$ cat slurm-243982.stats
| Assigned to nodes bask-pg0309u03a
| Command /bask/projects/m/myproject/example-job.sh
| WorkDir /bask/projects/m/myproject
+-----+
+-----+
| Finished at Fri Aug 26 08:45:11 2022 for username(834097) on the Baskerville Cluster
| Required (00:00.834 cputime, 3968K memory used) - 00:00:04 walltime
| JobState COMPLETING - Reason None
| Exitcode 0:0
+-----+
```

Your turn

- Adjust your job script with **sleep 5h** and submit your job. How do you cancel the job? *Hint:* use the command **scancel <jobid>**
- What happens if you **sleep 120** and request a wall-time job of 1 minute?

Your turn

- Adjust your job script with **sleep 5h** and submit your job. How do you cancel the job? *Hint:* use the command **scancel <jobid>**



```
[username@bask-pg0310u16a myproject]$ sbatch example-job.sh  
Submitted batch job 243985
```

```
[username@bask-pg0310u16a myproject]$ scancel 243985
```

```
[username@bask-pg0310u16a myproject]$ squeue -u $USER
```

JOBID	PARTITION	NAME	USER	ST	TIME	NODES	NODELIST(REASON)
243985	baskervil	example-	username	CG	0:06	1	bask-pg0309u03a

Your turn

- What happens if you **sleep 120** and request a wall-time job of 1 minute?



```
[username@bask-pg0310u16a myproject]$ cat slurm-243986.out
This script is running on slurmstepd: error: *** JOB 243986
ON bask-pg0309u03a CANCELLED AT 2022-08-26T09:33:12 DUE TO
TIME LIMIT ***
```

5. Accessing Software Applications

Let's look at your job script again

```
#!/bin/bash
#SBATCH --qos QOS
#SBATCH --account ACCOUNT
#SBATCH --time 1:0:0

module purge
module load baskerville

echo -n "This script is running on "
hostname
```

module purge unloads and loaded modules and resets the environment
module load baskerville loads the default Baskerville environment

List available modules

```
[username@bask-pg0310u16a myproject]$ module avail  
----- /bask/apps/live/EL8-cas/modules/all -----  
ASE/3.22.0-foss-2021a  
ATK/2.36.0-GCCcore-10.2.0  
AlphaFold/2.2.2-foss-2021a-CUDA-11.3.1  
Arm-Forge/22.0.3-foss-2021a  
Autoconf/2.69-GCCcore-10.2.0  
Autoconf/2.71-GCCcore-10.3.0 (D)  
Automake/1.16.2-GCCcore-10.2.0  
Automake/1.16.3-GCCcore-10.3.0 (D)  
Autotools/20200321-GCCcore-10.2.0  
Autotools/20210128-GCCcore-10.3.0 (D)  
BLIS/0.8.1-GCC-10.3.0  
Bazel/3.7.2-GCCcore-10.2.0  
Bazel/3.7.2-GCCcore-10.3.0 (D)  
Biopython/1.79-foss-2021a
```

Load Python module

```
[username@bask-pg0309u03a myproject]$ module purge
[username@bask-pg0309u03a myproject]$ module load baskerville
Detected OS: RedHatEnterprise 8.6
[username@bask-pg0309u03a myproject]$ module load Python/3.8.6-GCCcore-10.2.0
GCCcore/10.2.0
zlib/1.2.11-GCCcore-10.2.0
binutils/2.35-GCCcore-10.2.0
bzip2/1.0.8-GCCcore-10.2.0
zlib/1.2.11-GCCcore-10.2.0
ncurses/6.2-GCCcore-10.2.0
libreadline/8.0-GCCcore-10.2.0
ncurses/6.2-GCCcore-10.2.0
Tcl/8.6.10-GCCcore-10.2.0
SQLite/3.33.0-GCCcore-10.2.0
XZ/5.2.5-GCCcore-10.2.0
GMP/6.2.0-GCCcore-10.2.0
libffi/3.3-GCCcore-10.2.0
Python/3.8.6-GCCcore-10.2.0
[username@bask-pg0309u03a myproject]$ which python3
/bask/apps/live/EL8-ice/software/Python/3.8.6-GCCcore-10.2.0/bin/python3
```

View loaded modules



```
[username@bask-pg0309u03a myproject]$ module list
```

Currently Loaded Modules:

- | | |
|---------------------------------|------------------------------------|
| 1) bask-variables/live | 10) zlib/1.2.11-GCCcore-10.2.0 |
| 2) bask-singularity-conf/live | 11) libreadline/8.0-GCCcore-10.2.0 |
| 3) openmpi-env/live | 12) ncurses/6.2-GCCcore-10.2.0 |
| 4) bask-licences/live | 13) Tcl/8.6.10-GCCcore-10.2.0 |
| 5) bask-apps/live/live | 14) SQLite/3.33.0-GCCcore-10.2.0 |
| 6) baskerville/live | 15) XZ/5.2.5-GCCcore-10.2.0 |
| 7) GCCcore/10.2.0 | 16) GMP/6.2.0-GCCcore-10.2.0 |
| 8) binutils/2.35-GCCcore-10.2.0 | 17) libffi/3.3-GCCcore-10.2.0 |
| 9) bzip2/1.0.8-GCCcore-10.2.0 | 18) Python/3.8.6-GCCcore-10.2.0 |

Unload Python module



```
[username@bask-pg0309u03a myproject]$ module unload Python/3.8.6-GCCcore-10.2.0
```

View module details



```
[username@bask-pg0309u03a myproject]$ module spider Python/3.8.6-GCCcore-10.2.0
```

Python: Python/3.8.6-GCCcore-10.2.0

Description:

Python is a programming language that lets you work more quickly and integrate your systems more effectively.

Your turn

- Create a Python script called **hello.py** that contains

```
● ● ●

# hello.py
import socket
import time
node = socket.gethostname()
ts1 = time.ctime()
print(f'Hello from {node} at {ts1}')
time.sleep(3)
ts2 = time.ctime()
print(f'Bye-bye from {node} at {ts2}')
```

- Write a job script that loads a Python module and executes the command **python hello.py**
- Submit and run the job

Your turn - Result

```
[username@bask-pg0310u16a myproject]$ cat slurm-244002.out
GCCcore/10.2.0
zlib/1.2.11-GCCcore-10.2.0
binutils/2.35-GCCcore-10.2.0
bzip2/1.0.8-GCCcore-10.2.0
zlib/1.2.11-GCCcore-10.2.0
ncurses/6.2-GCCcore-10.2.0
libreadline/8.0-GCCcore-10.2.0
ncurses/6.2-GCCcore-10.2.0
Tcl/8.6.10-GCCcore-10.2.0
SQLite/3.33.0-GCCcore-10.2.0
XZ/5.2.5-GCCcore-10.2.0
GMP/6.2.0-GCCcore-10.2.0
libffi/3.3-GCCcore-10.2.0
Python/3.8.6-GCCcore-10.2.0
Hello from bask-pg0309u03a.cluster.baskerville.ac.uk at Fri
Aug 26 11:04:15 2022
Bye-bye from bask-pg0309u03a.cluster.baskerville.ac.uk at
Fri Aug 26 11:04:18 2022
```

6. Transferring files to/from Baskerville

Baskerville → Local Machine



```
→ ~ rsync -zavh username@login.baskerville.ac.uk:/bask  
/projects/m/myproject/slurm-244002.out ./
```

Local Machine → Baskerville



```
→ ~ rsync -zavh ./local_file.py  
username@login.baskerville.ac.uk:/bask/projects/m/myproject
```

7. Final Task: Running a Job with GPUs

Available GPUs



```
username@bask-pg0310u16a myproject]$ baskstatus
Current Baskerville GPU availability:
* 1 node with 3 x A100-40 available
* 10 nodes with 1 x A100-40 available
* 2 nodes with 4 x A100-80 available
* 1 node with 2 x A100-80 available
```

Query GPUs with PyTorch

1. Write a job script requesting a single GPU and **module load PyTorch**
2. Create a file called **pytorch-gpu.py** and execute with your job script
3. Submit your job script



```
#SBATCH --nodes 1  
#SBATCH --gpus 1  
#SBATCH --cpus-per-gpu 36
```

Snippet for requesting GPUs



```
import torch  
  
use_cuda = torch.cuda.is_available()  
  
if use_cuda:  
    print('__CUDNN VERSION:', torch.backends.cudnn.version())  
    print('__Number CUDA Devices:',  
          torch.cuda.device_count())  
    print('__CUDA Device  
Name:', torch.cuda.get_device_name(0))  
    print('__CUDA Device Total Memory  
[GB]:', torch.cuda.get_device_properties(0).total_memory/1e9)
```

pytorch-gpu.py

Query GPUs with PyTorch - Result

```
+ python pytorch-gpu.py
__CUDNN VERSION: 8201
__Number CUDA Devices: 1
__CUDA Device Name: NVIDIA A100-SXM4-40GB
__CUDA Device Total Memory [GB]: 42.505273344
```

Snippet of **slurm-<xxxxxx>.out**

Congratulations!

Useful links:

- <https://docs.baskerville.ac.uk/>
- <https://admin.baskerville.ac.uk/>
- <https://apps.baskerville.ac.uk/>
- <https://portal.baskerville.ac.uk/>

Email:

- baskerville-tier2-support@contacts.bham.ac.uk

