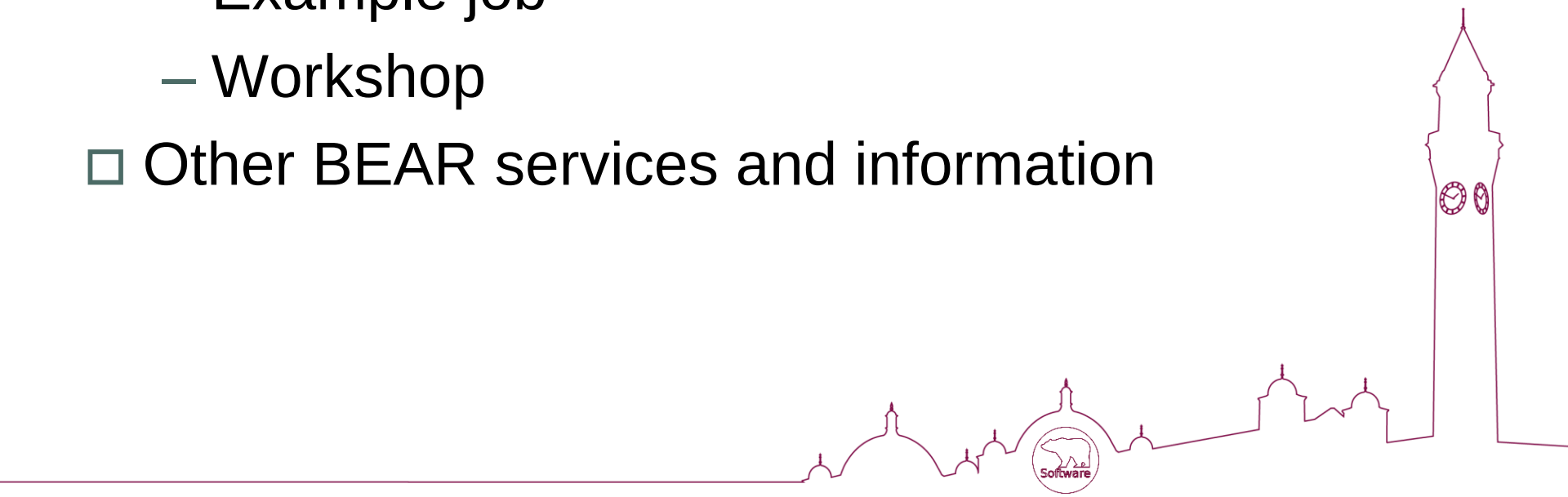
UNIVERSITY OF BIRMINGHAM | RESEARCH SOFTWARE GROUP

Introduction to BlueBEAR

March 2021

Overview

- Intro to BlueBEAR and batch computing
- Accessing and using BlueBEAR
- Running jobs on BlueBEAR
  - Example job
  - Workshop
- Other BEAR services and information

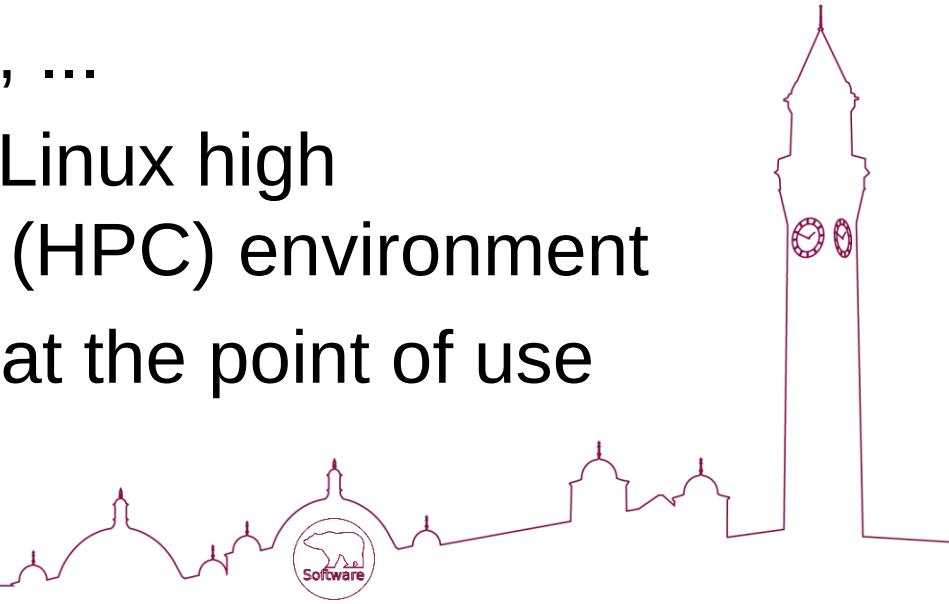


UNIVERSITY OF BIRMINGHAM | RESEARCH SOFTWARE GROUP

Intro to BlueBEAR and batch computing

What is BEAR?

- https://intranet.birmingham.ac.uk/bear
- Birmingham Environment for Academic Research
- BEAR is a collection of services: HPC, storage, fast networking, ...
- BlueBEAR refers to the Linux high performance computing (HPC) environment
- BEAR services are free at the point of use

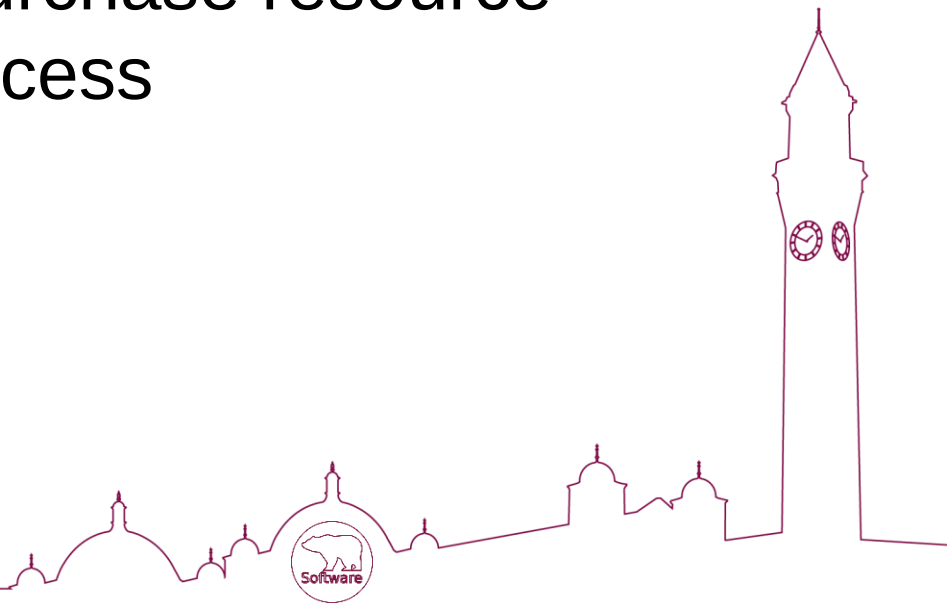


RDS and BEAR GitLab

- Use these two services!
  - GitLab
    - https://intranet.birmingham.ac.uk/bear-gitlab
  - Research Data Store
    - https://intranet.birmingham.ac.uk/research-data-store

BlueBEAR

- BlueBEAR is the Linux HPC system (cluster)
- Currently in its third generation
- Funded by the University
- Research groups can purchase resource providing preferential access

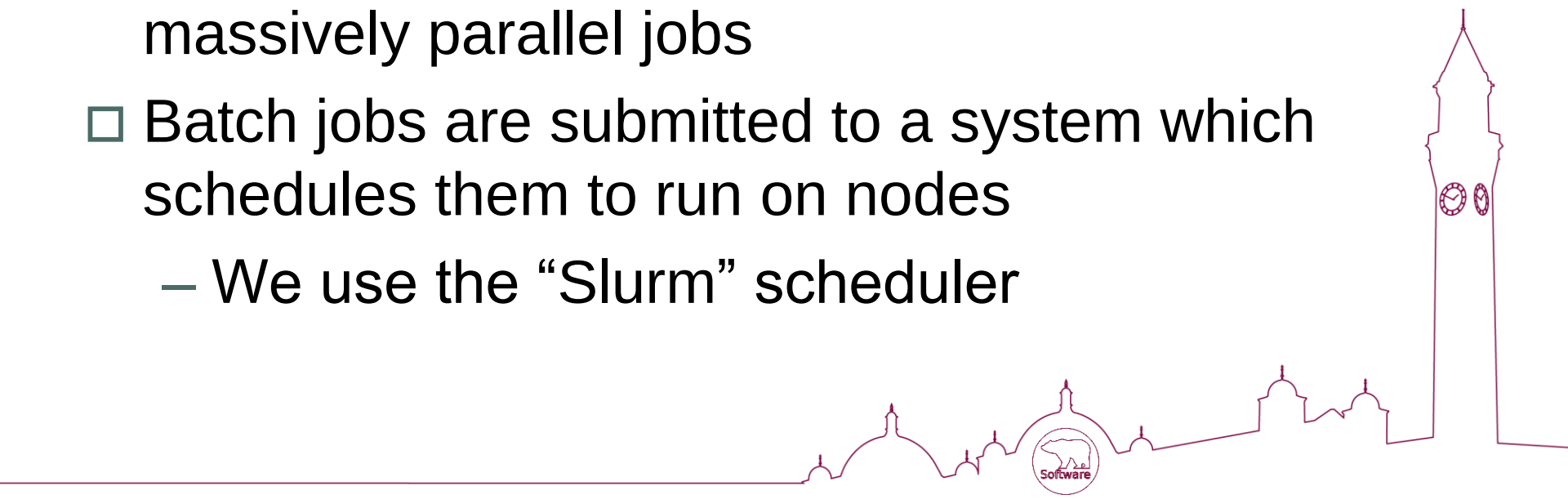


BlueBEAR

- Users need to register to use the service
- Users are attached to (multiple) projects
- Projects are created by staff
- Projects are used to account for time on the cluster
- Registrations are via: https://intranet.birmingham.ac.uk/bluebear

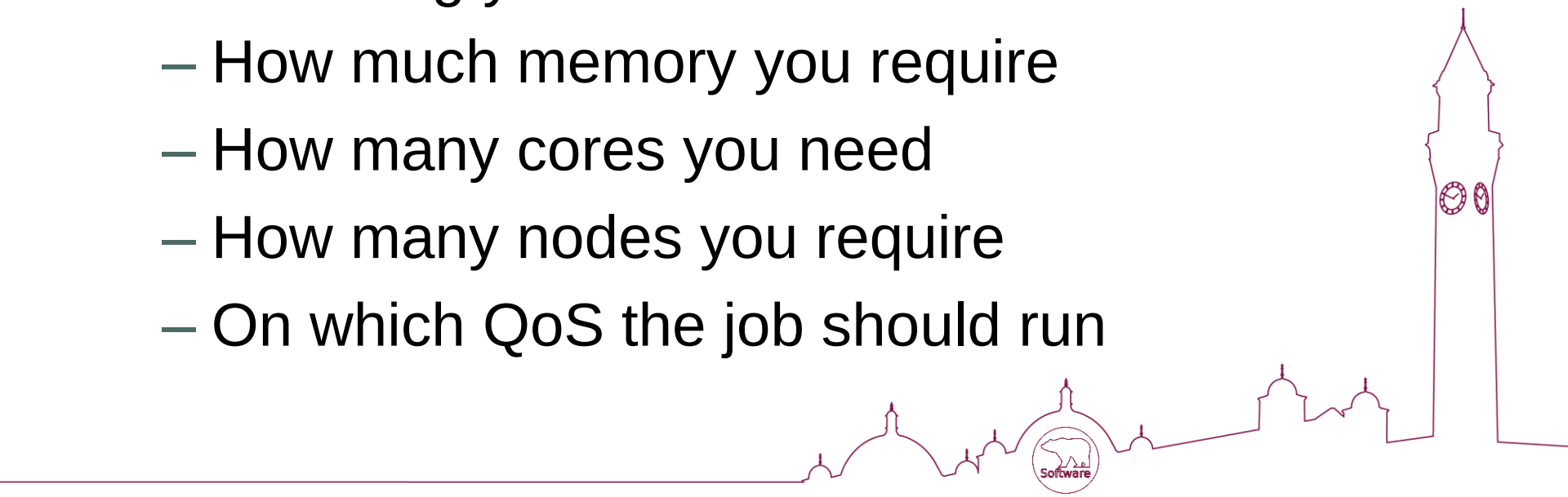
What is batch computing?

- Batch computing allows you to submit work for processing without you being present to control the work
- Batch computing may be single-core jobs, or massively parallel jobs
- Batch jobs are submitted to a system which schedules them to run on nodes
  - We use the “Slurm” scheduler



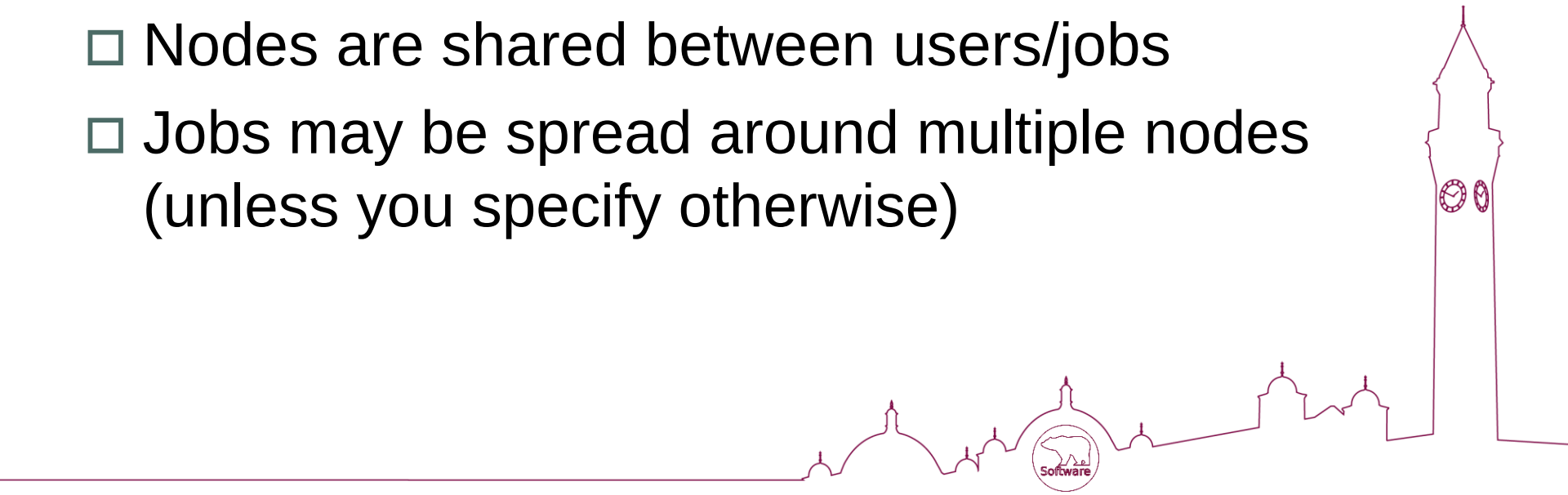
Batch computing

- When you submit a job, you tell Slurm some information about the job
  - (N.B. there are default values)
  - How long you think it will run for
  - How much memory you require
  - How many cores you need
  - How many nodes you require
  - On which QoS the job should run



Scheduling fairly

- Fair-share is applied
- Jobs must specify a project code to which the ‘work’ is attributed (unless you’ve only got one)
- Nodes are shared between users/jobs
- Jobs may be spread around multiple nodes (unless you specify otherwise)

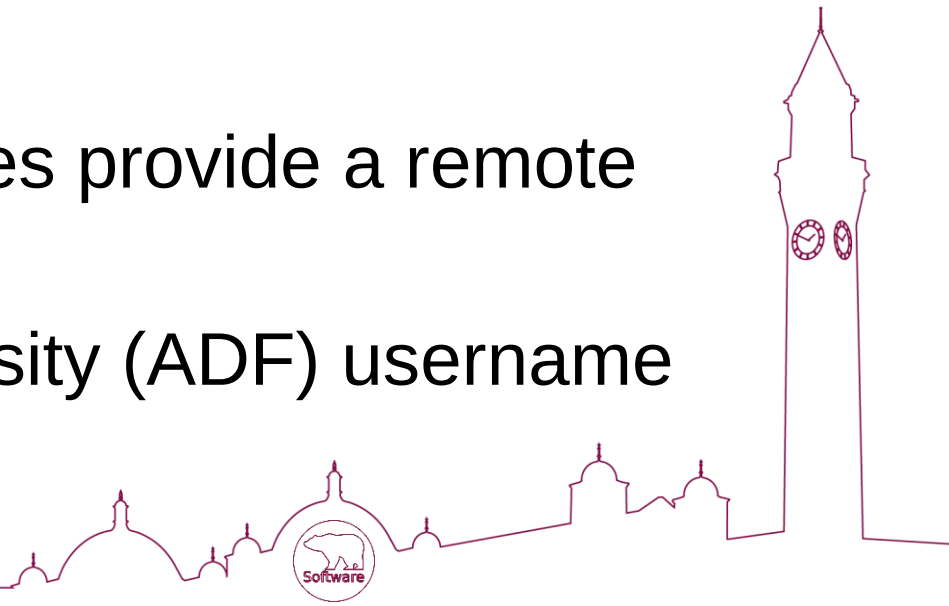


UNIVERSITY OF BIRMINGHAM | RESEARCH SOFTWARE GROUP

Accessing and using BlueBEAR

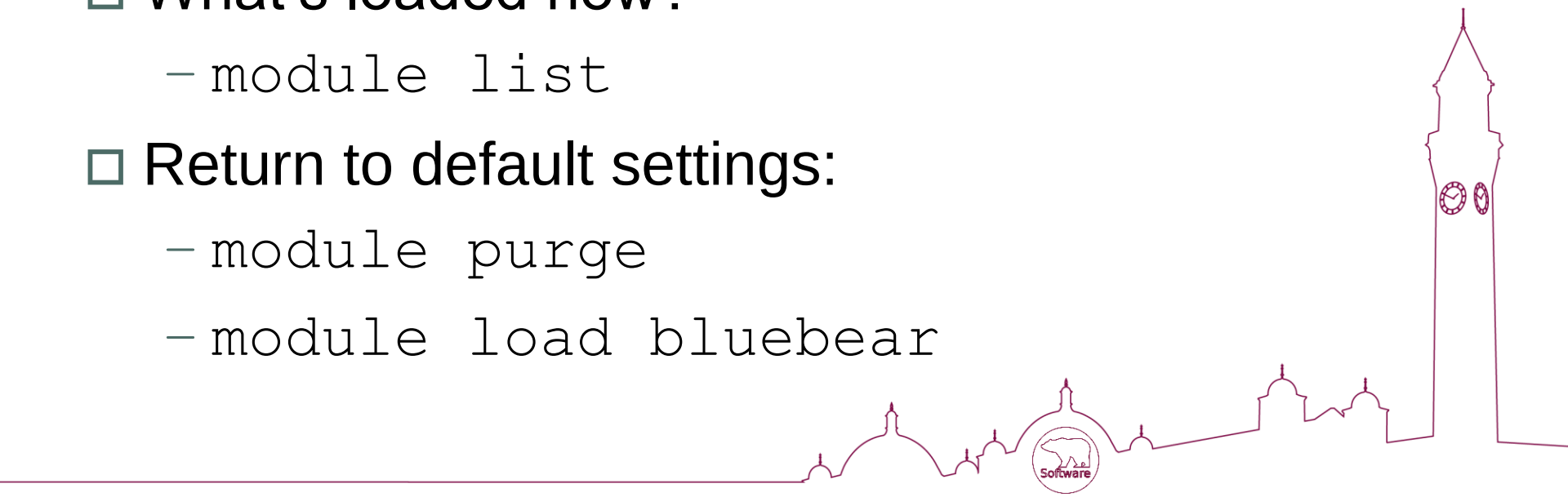
Accessing BlueBEAR

- You must register for access to BlueBEAR
- You will need an SSH client (e.g. *PuTTY* on Windows, *Terminal* on macOS etc.)
- You can only connect to the cluster from the University network
  - But the University does provide a remote access service (VPN)
- Use your normal University (ADF) username and password



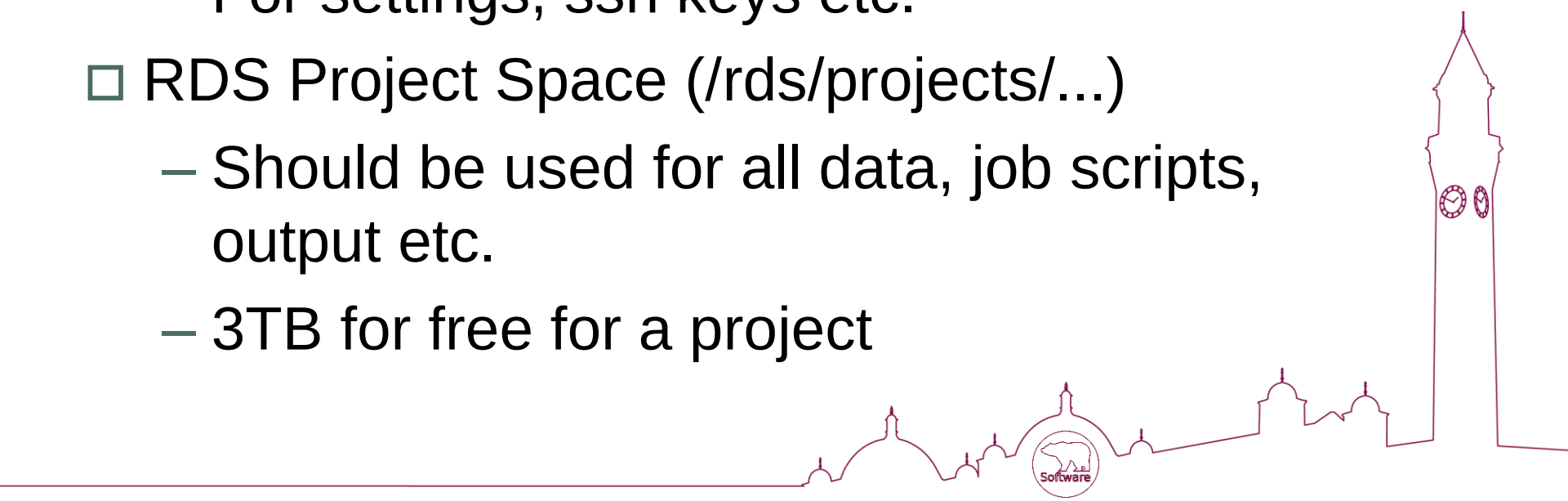
Applications

- We use “module” to manage applications
- Load Matlab:
  - `module load MATLAB/2020a`
- What’s loaded now?
  - `module list`
- Return to default settings:
  - `module purge`
  - `module load bluebear`



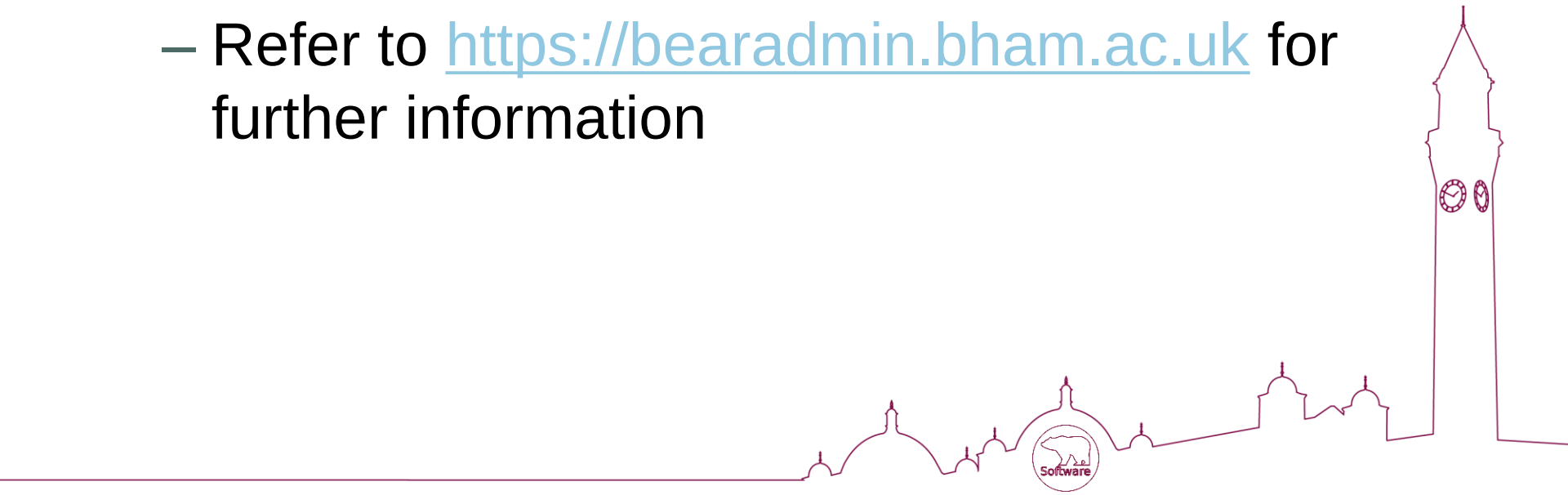
Storage

- Available on all nodes:
- Your home directory (/rds/homes/…)
  - 20GB quota
  - For settings, ssh keys etc.
- RDS Project Space (/rds/projects/...)
  - Should be used for all data, job scripts, output etc.
  - 3TB for free for a project



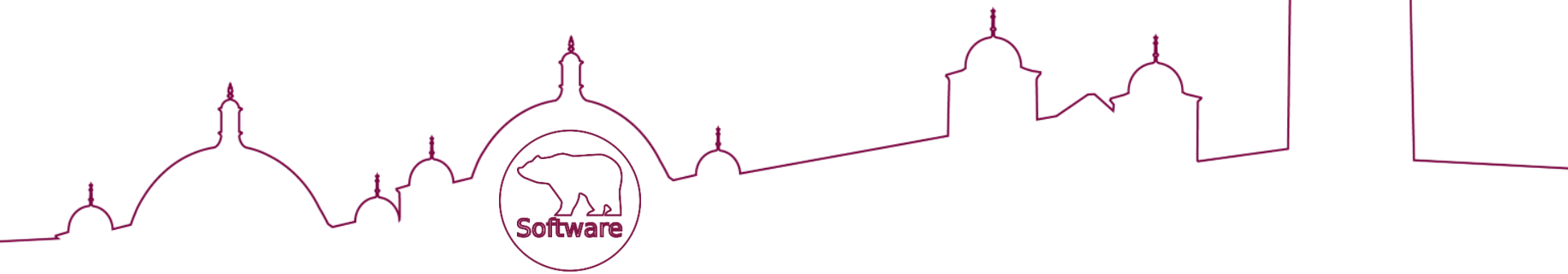
Accessing and transferring files

- Mount your RDS project on your desktop machine and copy files in and out
  - https://intranet.birmingham.ac.uk/HowToRDS
  - Refer to https://bearadmin.bham.ac.uk for further information



UNIVERSITY OF BIRMINGHAM | RESEARCH SOFTWARE GROUP

Running jobs on BlueBEAR



Running jobs on BlueBEAR

- How to write a job script
- Limits
- Choosing the right QoS
- Multi-core jobs
- Example job scripts

```
Example snippets look like this:
#SBATCH --ntasks 16
```

Running jobs: Job scripts

- A job script contains:
  - A header, telling the scheduler what resources you need.

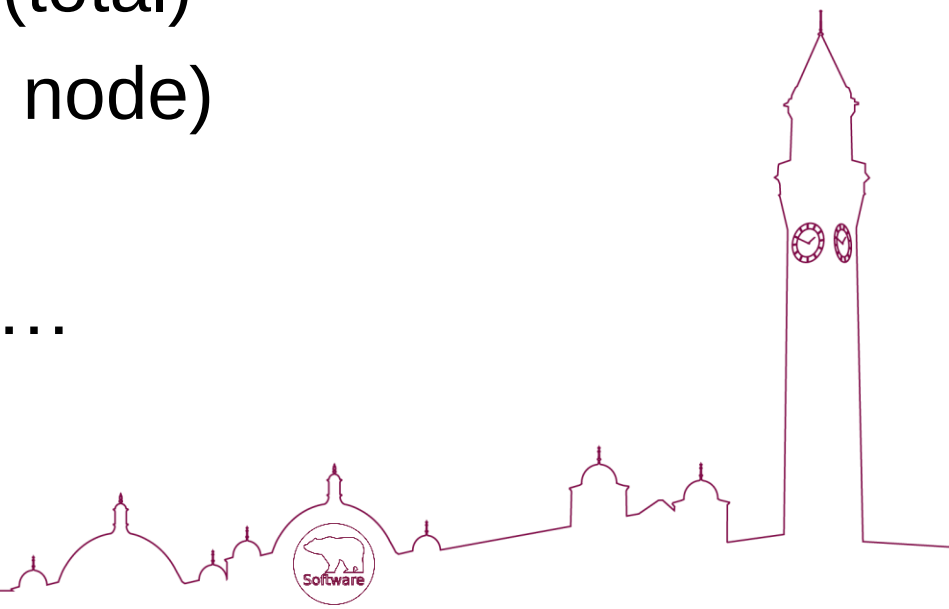
```
#!/bin/bash
#SBATCH --time 5
```

  - A body, containing the commands you want to run

```
module load Python/3.8.2-GCCcore-9.3.0
python -c "print('hello world')"
```

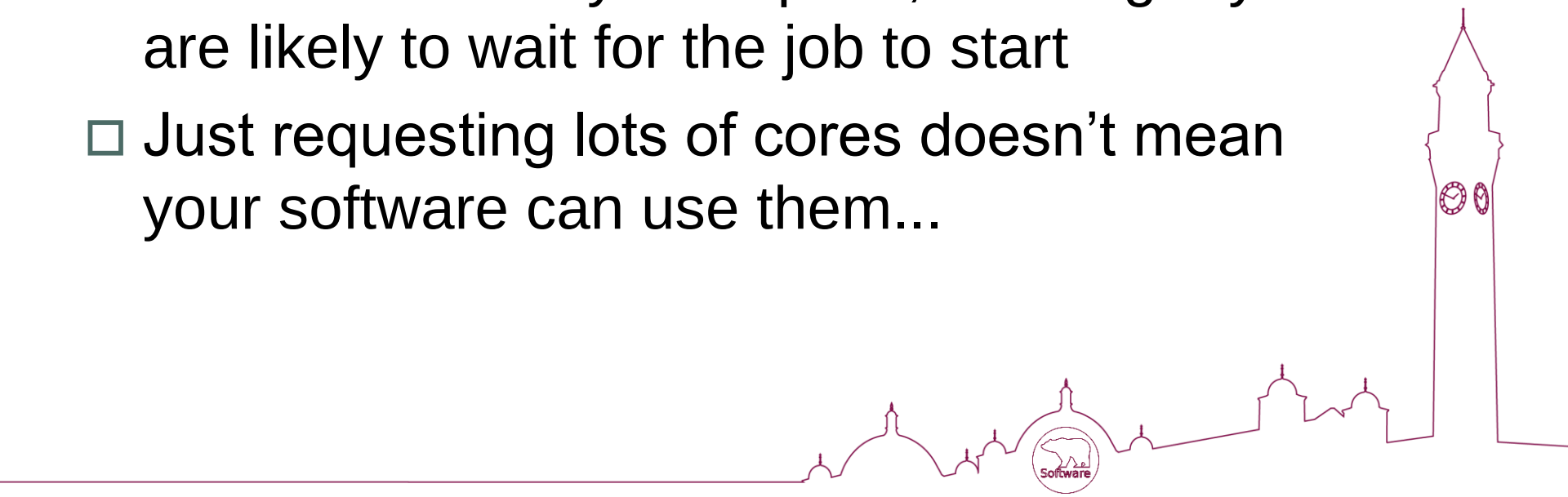
Running: Limits

- Total per user per shared QoS:
  - 320 cores, 3TB RAM
- Biggest job (in a shared QoS):
  - 320 cores, 3TB RAM (total)
  - 498 GB RAM (on one node)
- Time: 10 days
- More details in a minute…



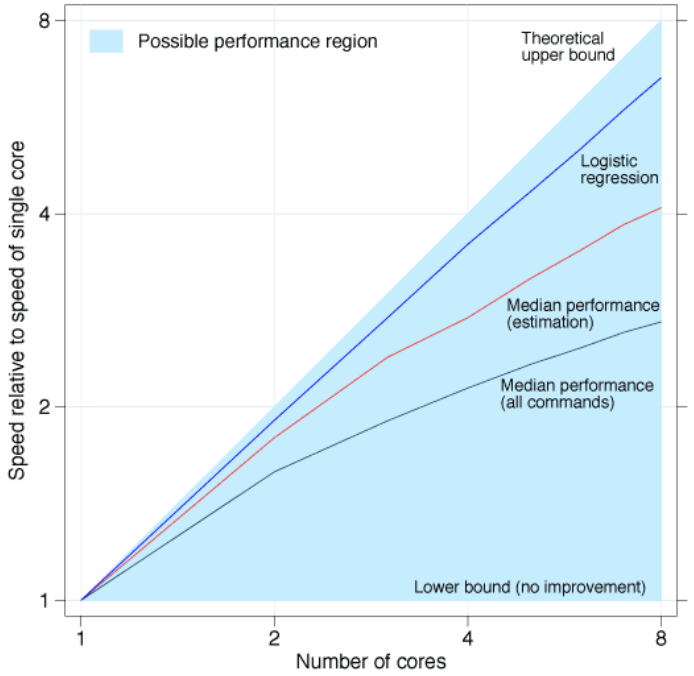
Multi-core jobs

- Not all jobs scale well over multiple cores
- Take the time to look at some short runs of your jobs to see how they perform
- The more cores you request, the longer you are likely to wait for the job to start
- Just requesting lots of cores doesn’t mean your software can use them...



Multi-core jobs

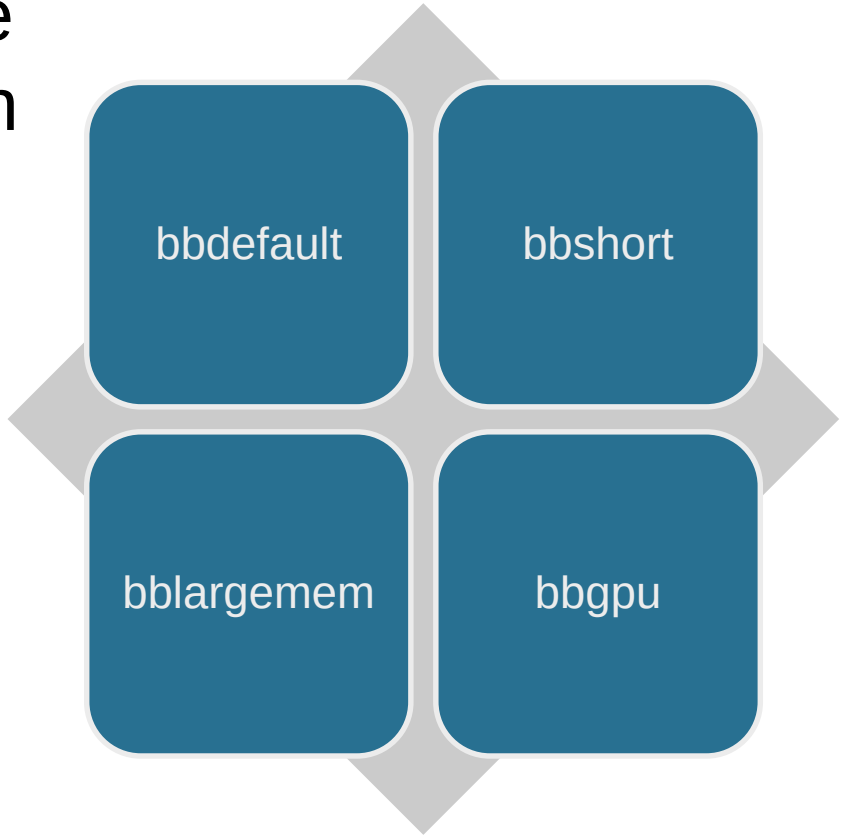
- For example, for STATA
- In a perfect world, software would run twice as fast on two cores, four times as fast on four cores, eight times as fast on eight cores, and so on.
- Across all commands, Stata/MP runs 1.6 times faster on two cores, 2.1 times faster on four cores, and 2.7 times faster on eight cores.
- These values are median speed improvements. Half the commands run even faster.



Source: https://www.stata.com/statamp/

Running jobs: QoS

QoS: Choose one based on your needs



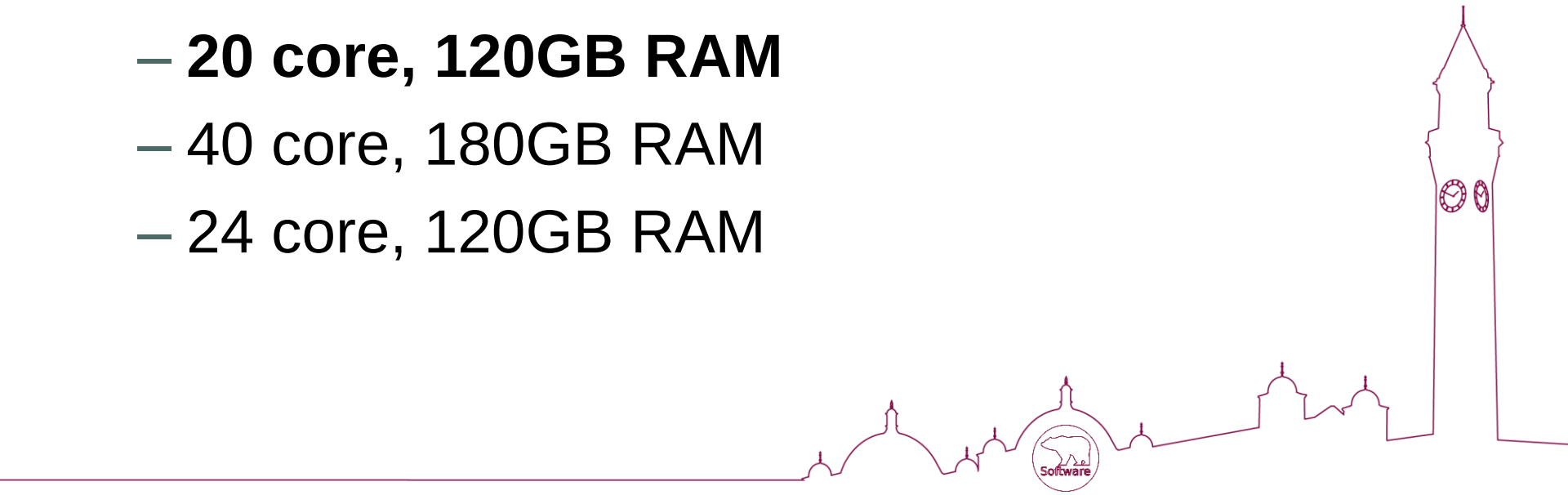

```
#SBATCH --qos bbshort
```

Running Jobs: bbdefault

- The *bbdefault* QoS is made up of different types of node:

```
#SBATCH --qos bbdefault
#SBATCH --ntasks 8
#SBATCH --time 1-2:0:0
```

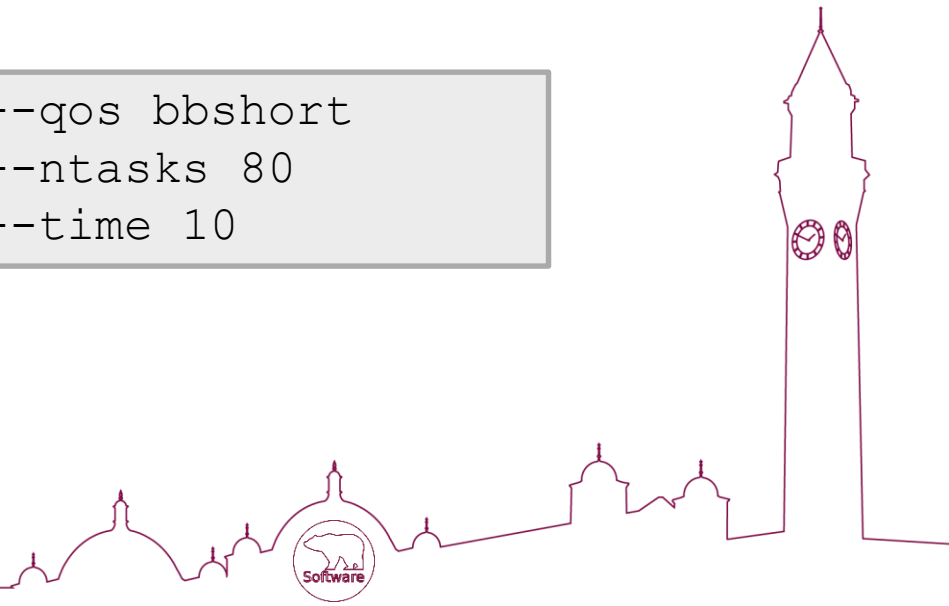
  - **20 core, 120GB RAM**
  - 40 core, 180GB RAM
  - 24 core, 120GB RAM



Running Jobs: bbshort

- The *bbshort* QoS:
  - Contains all nodes
  - Fastest way of getting your job run
  - 10 minute max time

```
#SBATCH --qos bbshort
#SBATCH --ntasks 80
#SBATCH --time 10
```



Running Jobs: bblargemem

- You have to request for this QoS to be added to your project
- The *bblargemem* QoS:
  - Contains a mix of large memory nodes, up to a maximum of 498G RAM
  - Specify what you need, for example…

```
#SBATCH --qos bblargemem
#SBATCH --ntasks 10
#SBATCH --mem 200G
```

Running Jobs: bbgpu

- You have to request for this QoS to be added to your project
- The *bbgpu* QoS:
  - Contains nodes with Nvidia p100 GPUs
  - Specify the following in your batch script…

```
#SBATCH --qos bbgpu
#SBATCH --gres gpu:p100:1
```

Running jobs: Multi-core

- Multi-core jobs:
  - Needs to be on one node:
    - Multiprocessing / Threading
    - OpenMP

```
#SBATCH --nodes 1-1
#SBATCH --ntasks 10
```

  - Can span multiple nodes:
    - OpenMPI

```
#SBATCH --ntasks 200
```

- NOTE: Array Jobs are better in some cases.

UNIVERSITY OF BIRMINGHAM | RESEARCH SOFTWARE GROUP

Example job

Python

Example job: Python

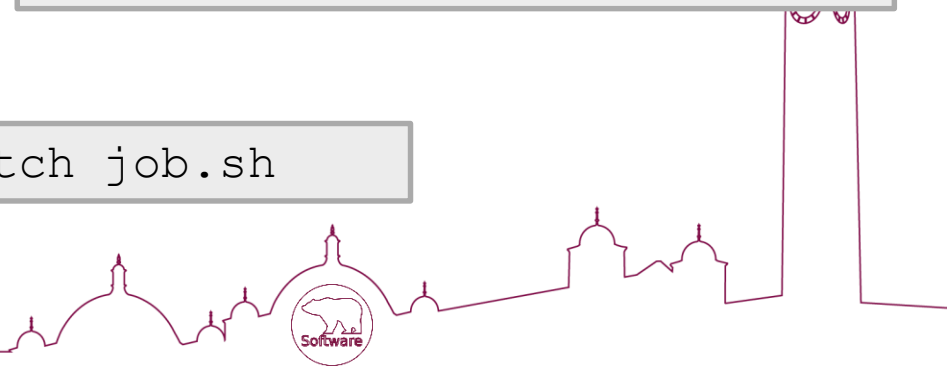
job.sh

```
#!/bin/bash
#SBATCH --ntasks 1
#SBATCH --time 5
#SBATCH --qos bbshort
#SBATCH --mail-type ALL

set -e

module purge; module load bluebear
module load Python/3.8.2-GCCcore-9.3.0

python hello.py
```

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}')
```

To schedule it: `sbatch job.sh`



Running jobs: status / cancel

- To find out what's going on run:
  - `showq` or `squeue`
  - `sacct`
  - `scontrol show job 12345`
- To cancel a job, run:
  - `scancel 12345`

Job Output

- Two files created:
  - `slurm-3381968.out`
    - The output from the job as it runs
    - what would normally be printed to screen
  - `slurm-3381968.stats`
    - System information on the job

UNIVERSITY OF BIRMINGHAM | RESEARCH SOFTWARE GROUP

Workshop

Python

Workshop

- Time: 20 minutes (including a break)
- Run the previous Python example yourself
- Use **`showq`** or **`squeue`** to see it in action
- Use **`tail -f`** to watch the output log files
- Use **`scontrol show job 12345`** to find out about the job – if you want!

- When you're finished, take a break!

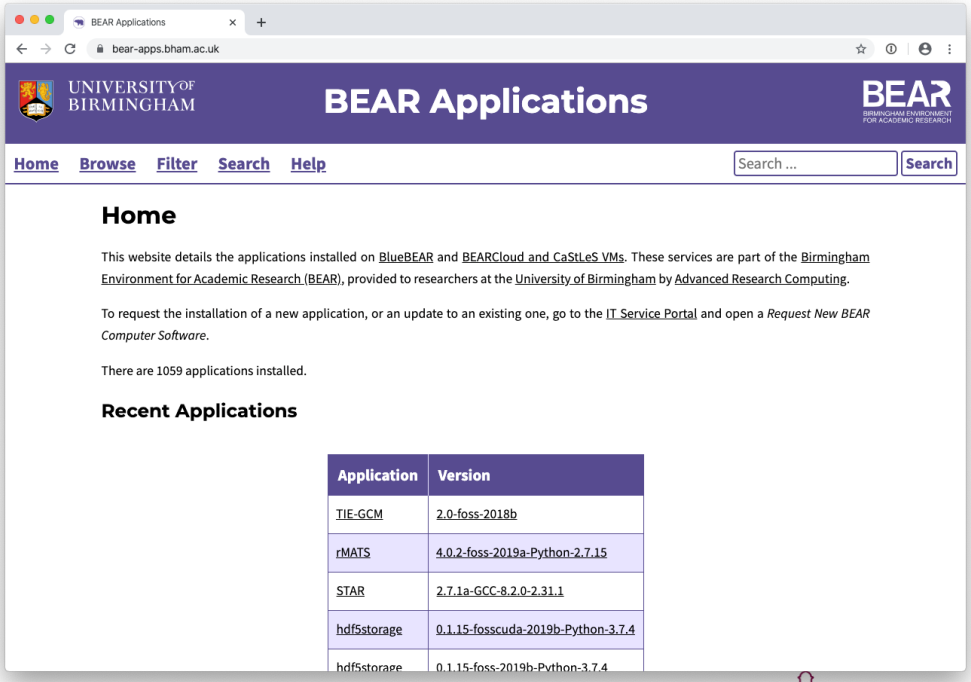
UNIVERSITY OF BIRMINGHAM | RESEARCH SOFTWARE GROUP

Making the most of BlueBEAR

Further info, options and advanced tools

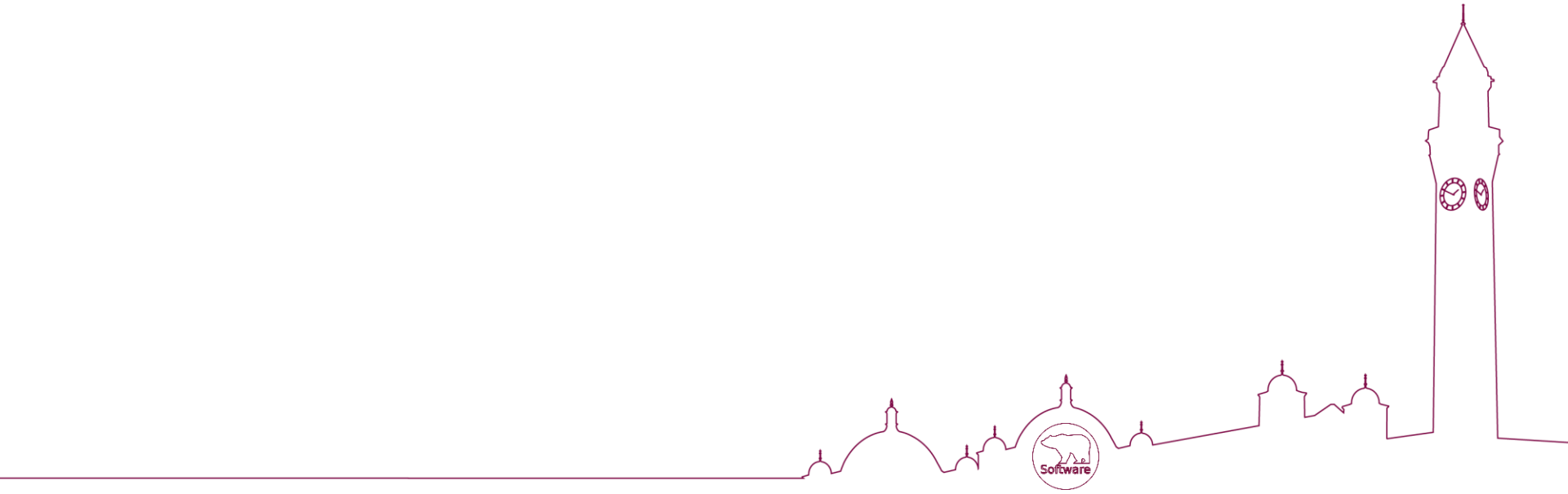
https://bear-apps.bham.ac.uk

- For current details on all of the applications available on BlueBEAR



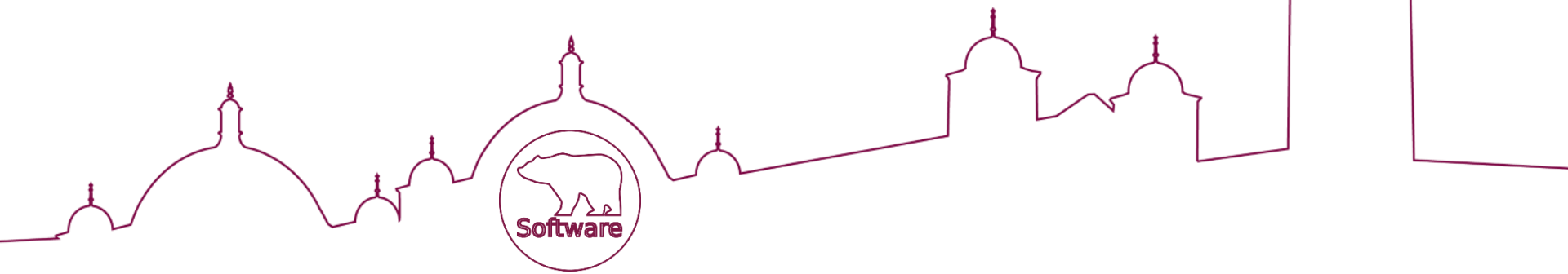
BEAR Portal

https://portal.bear.bham.ac.uk



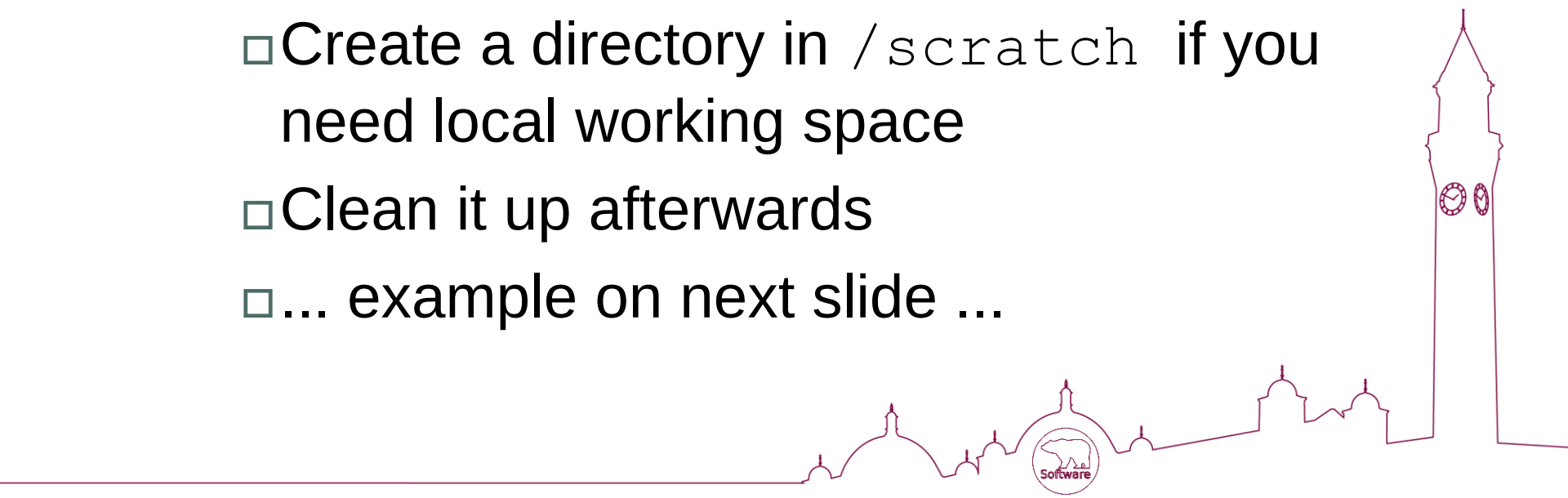
UNIVERSITY OF BIRMINGHAM | RESEARCH SOFTWARE GROUP

Temporary storage using /scratch



/scratch storage

- `/scratch` storage is available on each node, to be used by jobs for working data (e.g. `${TMPDIR}`)
  - In your submission script:
    - Create a directory in `/scratch` if you need local working space
    - Clean it up afterwards
    - ... example on next slide ...



/scratch storage

- See "*Use Local Disk Space*" in BlueBEAR Job Submission web page
- At the start of your job script:

```
BB_WORKDIR=$(mktemp -d /scratch/${USER}_${SLURM_JOBID}.XXXXXX)

export TMPDIR=${BB_WORKDIR}
```

- And clean up at the end of your job script:

```
test -d ${BB_WORKDIR} && /bin/rm -rf ${BB_WORKDIR}
```

UNIVERSITY OF BIRMINGHAM | RESEARCH SOFTWARE GROUP

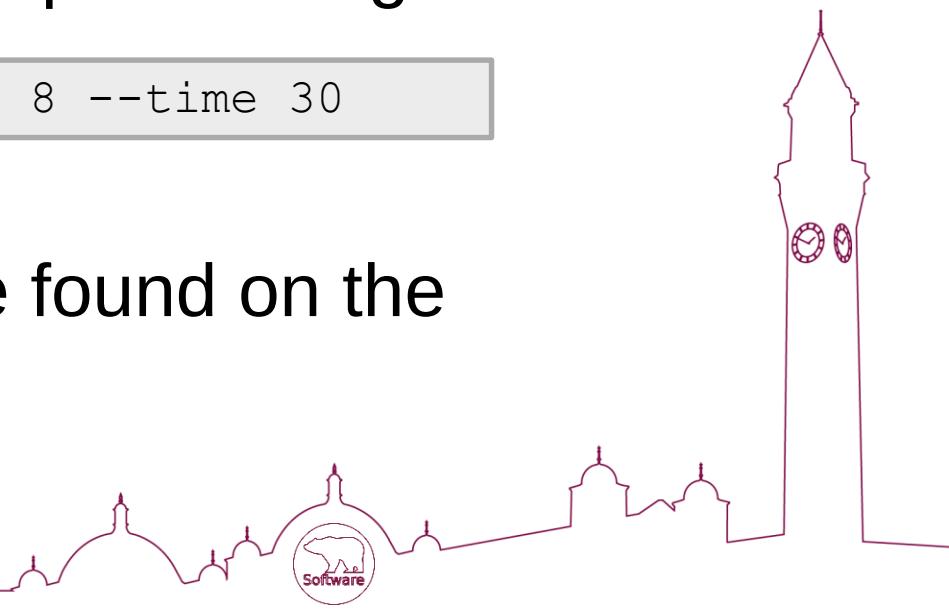
BlueBEAR interactive jobs

Interactive Jobs overview

- For scenarios where you need more interaction, e.g. testing commands & workflow
- module load slurm-interactive …
- fisbatch_screen <slurm-options> e.g:

```
fisbatch_screen --ntasks 8 --time 30
```

- More information can be found on the BlueBEAR webpages



Email notifications

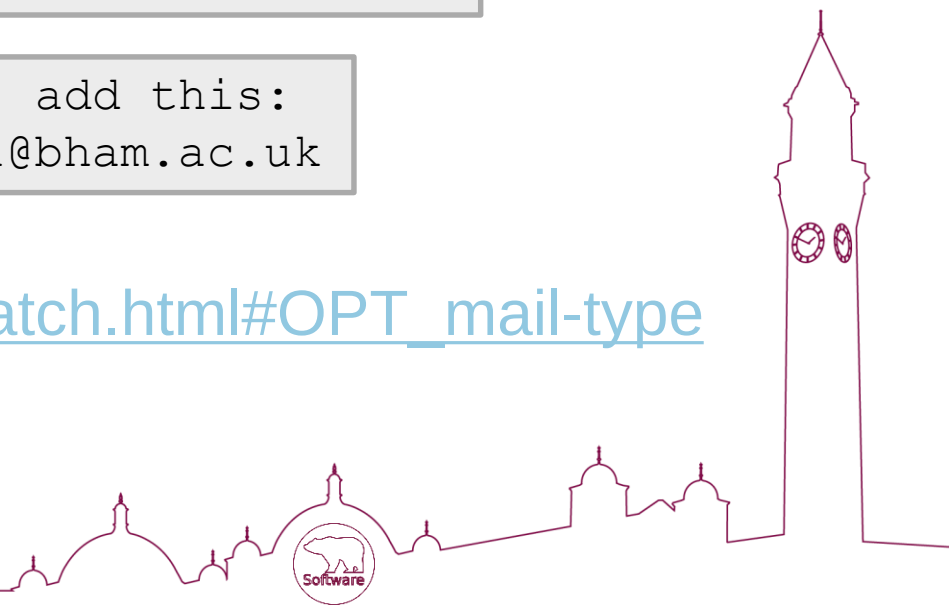
- Slurm can tell you when jobs start, complete, fail, ...

```
#SBATCH --mail-type ALL
```

```
#SBATCH --mail-type FAIL
```

```
If you're not getting emails, add this:
#SBATCH --mail-user m.y.email@bham.ac.uk
```

https://slurm.schedmd.com/sbatch.html#OPT_mail-type



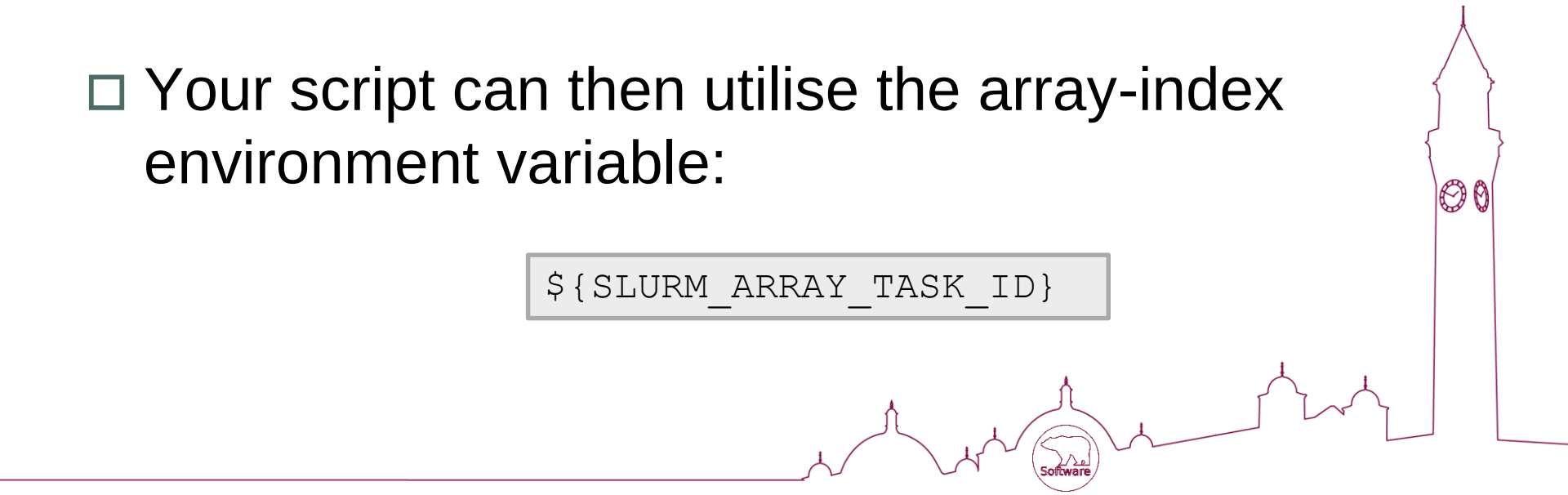
Slurm Job Arrays

- If you have multiple repetitive tasks (especially if your code can't be parallelised) then array jobs can be very helpful:

```
#SBATCH --array 0-31
```

- Your script can then utilise the array-index environment variable:

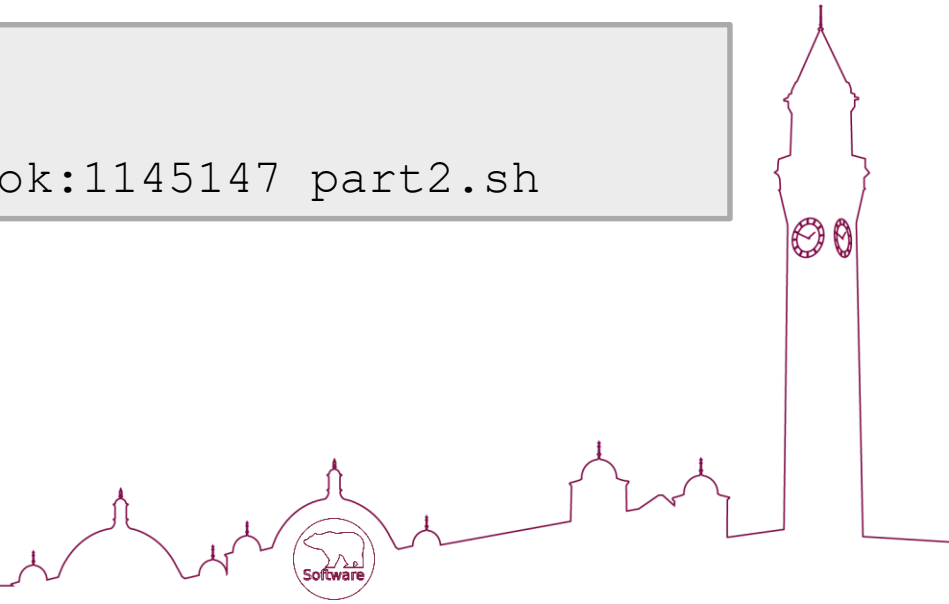
```
${SLURM_ARRAY_TASK_ID}
```



Job dependencies

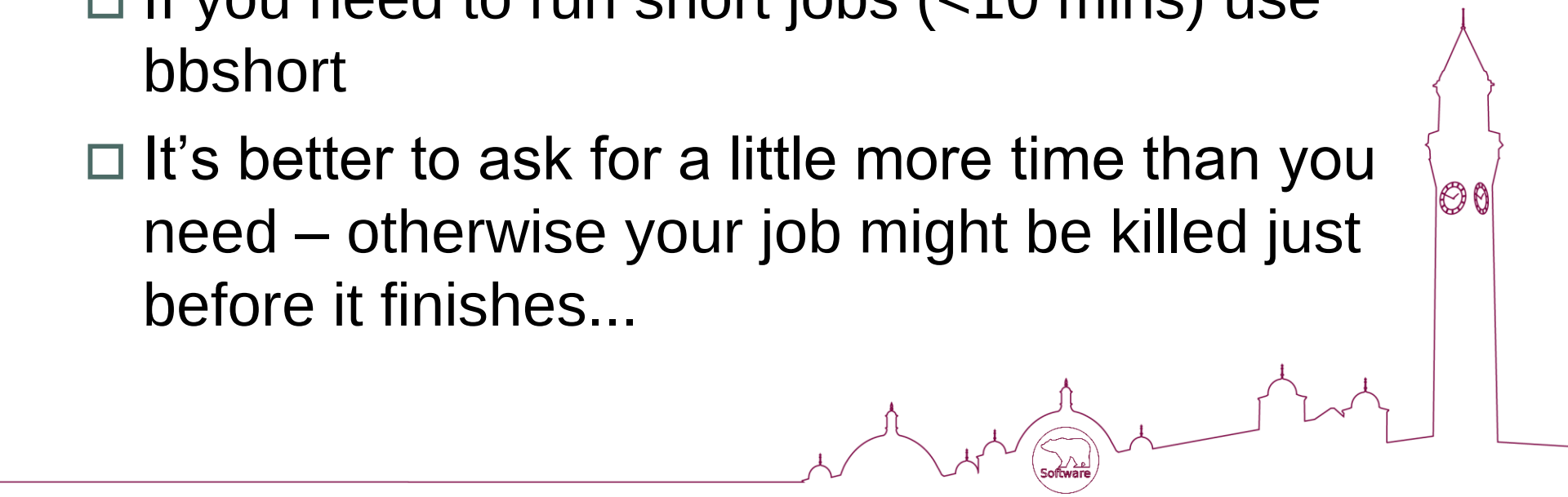
- If your work splits into several jobs then
  - you can submit these all at the same time and tell Slurm to run the later jobs after the earlier jobs have successfully completed

```
$ sbatch part1.sh
Submitted batch job 1145147
$ sbatch --dependency=afterok:1145147 part2.sh
```



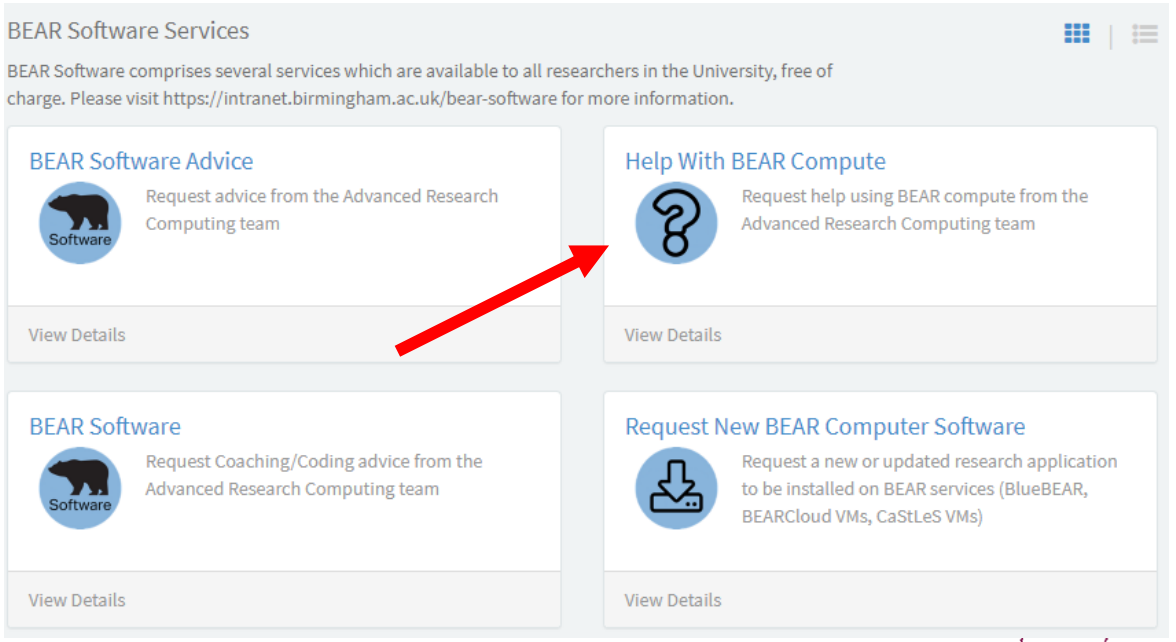
Resources

- Aim to be accurate with your resource requests – this will help utilise the resources better and reduce queue times
- Break your work into smaller chunks
- If you need to run short jobs (<10 mins) use bbshort
- It’s better to ask for a little more time than you need – otherwise your job might be killed just before it finishes...



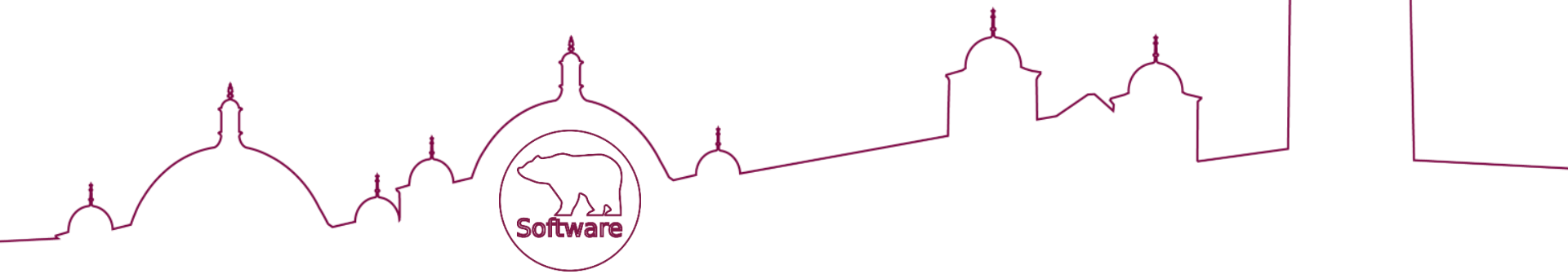
Help is available

- If you want help or advice on batch jobs – visit the IT Service Desk web portal:
  - ‘Help with BEAR compute’



UNIVERSITY OF BIRMINGHAM | RESEARCH SOFTWARE GROUP

Other BEAR services and information



Other BEAR Services

- Research Data Store (RDS): Free storage for research projects (up to 3TB per project)
- BEAR DataShare: File synchronisation and sharing service
- Research Data Network (RDN): dedicated network to connect research facilities that generate very high volumes of data
- BEAR Cloud: Local high-performance cloud computing integrated with campus services
- BEAR Software: Free advice/help from BEAR RSEs
- ... and more at https://intranet.birmingham.ac.uk/bear

Campus Groups

- Birmingham RSE Slack Channel:
  - https://bham-rse.slack.com/
- Special Interest Groups:
  - Bioinformatics; Academic Programmers; Computational Fluid Dynamics (CFD); Finite Element Method (FEM); Matlab; Stata