SLURM @UPPMAX

Diana Iuşan

Application expert and

training coordinator UPPMAX

UPPSALA
UNIVERSITET

The **Slurm** Workload Manager

- provides a framework for starting, executing, and monitoring jobs on the compute nodes

UPPSALA
UNIVERSITET

The **Slurm** Workload Manager

- provides a framework for starting, executing, and monitoring jobs on the compute nodes
- schedules the jobs on the clusters
- allocates the required resources (compute cores or nodes, memory)

UPPSALA
UNIVERSITET

The **Slurm** Workload Manager

- Free, popular, lightweight
- Open source:

  https://slurm.schedmd.com
- available at al SNIC centra
- UPPMAX Slurm userguide:

  https://docs.uppmax.uu.se/cluster_guides/slurm/

UPPSALA UNIVERSITET

How to submit a job?

- Recap:

`sbatch`	`-A uppmax2025-2-262`	`-t 10:00`	`-p pelle`	`-n 10`	`my_job.sh`
Slurm batch	Project name	Maximum runtime	Partition (“job type”)	#cores	job script

Job time limits

- `-t minutes`
- `-t minutes:seconds`
- `-t hours:minutes:seconds`
- `-t days-hours`
- `-t days-hours:minutes`
- **`-t days-hours:minutes:seconds`**

UPPSALA
UNIVERSITET

What should the wall time of a job be?

- Your first job? Testing new software or input?

UPPSALA
UNIVERSITET

What should the wall time of a job be?

- Your first job? Testing new software or input?
  - Use a short time limit, 10 min - 1h.

UPPSALA
UNIVERSITET

What should the wall time of a job be?

- Your first job? Testing new software or input?
  - Use a short time limit, 10 min - 1h.
- Q: When you have no idea how long a program will take to run, what should you book?

- Q: When you have an idea of how long a program would take to run, what should you book?

UPPSALA
UNIVERSITET

What should the wall time of a job be?

- Your first job? Testing new software or input?
  - Use a short time limit, 10 min - 1h.
- Q: When you have no idea how long a program will take to run, what should you book?
  - A: very long time, e.g. 10-00:00:00
- Q: When you have an idea of how long a program would take to run, what should you book?
  - A: overbook by 50%

UPPSALA
UNIVERSITET

Resources: Pelle

`sinfo` lists information on the available resources to Slurm

```
[user@pelle ~]$ sinfo -o "%10P %20l %10z %10c %20f %20G %20N %20m"
```

```
PARTITION  TIMELIMIT           S:C:T      CPUS       AVAIL_FEATURES       GRES                 NODELIST             MEMORY
pelle*     10-00:00:00         1:48:2     96         (null)               (null)               p[1-115]             772000
fat        10-00:00:00         1:48:2     96         2TB                  (null)               p251                 2320000
fat        10-00:00:00         1:48:2     96         3TB                  (null)               p252                 3090000
gpu        2-00:00:00          2:16:2     64         (null)               gpu:h100:2(S:1)      p[205-206]           386000
gpu        2-00:00:00          2:16:2     64         (null)               gpu:l40s:10(S:0-1)   p[201-204]           386000
```

UPPSALA
UNIVERSITET

Resources: Bianca

`sinfo` lists information on the available resources to Slurm

```
[user@bianca ~]$ sinfo -o "%10P %20l %10z %10c %20f %20G %20N %20m"
```

```
PARTITION  TIMELIMIT           S:C:T      CPUS       AVAIL_FEATURES       GRES                 NODELIST             MEMORY
all        10-00:00:00         2:8:1      16         cpu,thin,mem512GB,fa (null)               sens2017625-b[1-3,30 489000
all        10-00:00:00         16:1:1     16         gpu,usage_mail       gpu:a100:2           sens2017625-b[204-21 240000
all        10-00:00:00         2:8:1      16         cpu,thin,mem256GB,fa (null)               sens2017625-b[4-8,10 240000
all        10-00:00:00         2:8:1      16         cpu,thin,mem128GB,us (null)               sens2017625-b[9-200] 117000
all        10-00:00:00         16:1:1     16         gpu,usage_mail       gpu:a100:2(S:0-15)   sens2017625-b[201-20 240000
node       10-00:00:00         2:8:1      16         cpu,thin,mem512GB,fa (null)               sens2017625-b[1-3,30 489000
node       10-00:00:00         16:1:1     16         gpu,usage_mail       gpu:a100:2           sens2017625-b[204-21 240000
node       10-00:00:00         2:8:1      16         cpu,thin,mem256GB,fa (null)               sens2017625-b[4-8,10 240000
node       10-00:00:00         2:8:1      16         cpu,thin,mem128GB,us (null)               sens2017625-b[9-200] 117000
node       10-00:00:00         16:1:1     16         gpu,usage_mail       gpu:a100:2(S:0-15)   sens2017625-b[201-20 240000
core*      10-00:00:00         2:8:1      16         cpu,thin,mem512GB,fa (null)               sens2017625-b[1-3,30 489000
core*      10-00:00:00         16:1:1     16         gpu,usage_mail       gpu:a100:2           sens2017625-b[204-21 240000
core*      10-00:00:00         2:8:1      16         cpu,thin,mem256GB,fa (null)               sens2017625-b[4-8,10 240000
core*      10-00:00:00         2:8:1      16         cpu,thin,mem128GB,us (null)               sens2017625-b[9-200] 117000
core*      10-00:00:00         16:1:1     16         gpu,usage_mail       gpu:a100:2(S:0-15)   sens2017625-b[201-20 240000
devel      1:00:00             2:8:1      16         cpu,thin,mem256GB,fa (null)               sens2017625-b[1073-1 240000
devel      1:00:00             2:8:1      16         cpu,thin,mem128GB,us (null)               sens2017625-b[9-200] 117000
devcore    1:00:00             2:8:1      16         cpu,thin,mem256GB,fa (null)               sens2017625-b[1073-1 240000
devcore    1:00:00             2:8:1      16         cpu,thin,mem128GB,us (null)               sens2017625-b[9-200] 117000
```

UPPSALA
UNIVERSITET

Debugging or complicated workflows

- Interactive jobs
  - handy for debugging a code or a script by executing it line by line or for using programs with a graphical user interface
  - `salloc -n 10 -t 03:00:00 -A uppmax2025-2-262`
  - **`interactive`** `-n 10 -t 03:00:00 -A uppmax2025-2-262`

UPPSALA
UNIVERSITET

Debugging or complicated workflows

- Interactive jobs
  - handy for debugging a code or a script by executing it line by line or for using programs with a graphical user interface
  - `salloc -n 10 -t 03:00:00 -A uppmax2025-2-262`
  - **`interactive`** `-n 10 -t 03:00:00 -A uppmax2025-2-262`
  - useful together with the `--begin=<time>` flag
  - `salloc -A uppmax2025-2-262 --begin=2025-10-20T08:00:00`

    asks for an interactive job that will start

    earliest on Monday at 08:00

UPPSALA
UNIVERSITET

Parameters in the job script or the command line?

- Command line parameters override script parameters
- A typical script may be:

```
#!/bin/bash
#SBATCH -A uppmax2025-2-262
#SBATCH -p pelle
#SBATCH -n 1
#SBATCH -t 24:00:00
```

- Just a quick test:

```
sbatch -t 00:15:00 jobscript.sh
```

UPPSALA UNIVERSITET

Hands-on #1:
`sbatch/jobinfo`

- login to Pelle
- find out which projects you're a member of using SUPR
- submit a short (10 min) test job; note the job ID
- submit a new job to use a different partition

UPPSALA
UNIVERSITET

More **sbatch** options

- `-J <jobname>`
- email:
  - `--mail-type=BEGIN,END,FAIL,TIME_LIMIT_80`
  - `--mail-user` - Don't use. The SUPR email is taken by default.
- out/err redirection:
  - `--output=slurm-%j.out` *and* `—error=slurm-%j.err`

    by default, where `%j` will be replaced by the job ID
  - `--output=my.output.file`
  - `--error=my.error.file`

UPPSALA UNIVERSITET

More **sbatch** options

- Memory
  - `--mem--mem=<size>[units]`
  - `--mem-per-cpu=<size>[units]`
  - on Pelle: `-C 2TB / -C 3TB`
  - on Bianca: `-C fat / -C mem256GB / -C 512GB`
- Dependencies: `–dependency`
- Job array: `--array`
- More at https://slurm.schedmd.com/sbatch.html
  - or just `man sbatch`
  - not all options work on all systems!

UPPSALA
UNIVERSITET

Technical summary of the UPPMAX clusters

	Pelle	**Bianca**
Purpose	General-purpose	Sensitive
GPU Nodes	4 nodes with 10 Nvidia L40s each 2 nodes with Nvidia H100 each	10 nodes with 2 Nvidia A100 each
Memory per node	768 GB 2 and 3 TB on the fat nodes 384 GB on the GPU nodes	128 GB 256 or 512 on the fat nodes
Local disk (`/scratch`)	1.7 TB 6.9 TB on the fat and GPU nodes	4 TB

UPPSALA UNIVERSITET

Technical summary of the UPPMAX clusters

	Pelle	**Bianca**
Purpose	General-purpose	Sensitive
GPU Nodes	4 nodes with 10 Nvidia L40s each 2 nodes with Nvidia H100 each	10 nodes with 2 Nvidia A100 each
Memory per node	768 GB 2 and 3 TB on the fat nodes 384 GB on the GPU nodes	128 GB 256 or 512 on the fat nodes
Local disk (`/scratch`)	1.7 TB 6.9 TB on the fat and GPU nodes	4 TB

- find info on jobs and partitions:

```
sinfo -o "%P %l %z %c %f %G %N %m"
sinfo -o "%10P %20l %10z %10c %20f %20G %20N %20m”
```

- print the requested memory of a job:

```
sacct --format=ReqMem -j <jobid>
```

UPPSALA UNIVERSITET

More **sbatch** options

- Dependencies: `--dependency`
- Job array: `--array`
- More at https://slurm.schedmd.com/sbatch.html
  - or just `man sbatch`
  - not all options work on all systems!

UPPSALA
UNIVERSITET

From job submission to job monitoring

- `sbatch`

- `sinfo`
- `squeue`

UPPSALA
UNIVERSITET

Monitoring jobs

- **`squeue`**
  - lists running and pending jobs
  - `squeue -u` *`username`*
  - `squeue --me`
  - `squeue -A uppmax2025-2-262`
  - `squeue -u` *`username`* `--state=running`
  - `squeue -u` *`username`* `--state=pending`
- One may also use the **`jobinfo`** `wrapper` on Bianca.

UPPSALA UNIVERSITET

Monitoring and modifying jobs

- **`scontrol`**
  - `scontrol show job` *`jobid`*

UPPSALA
UNIVERSITET

Monitoring and modifying jobs

- **`scontrol`**
  - `scontrol show job` *`jobid`*
- possible to modify the job details after the job has been submitted; some options, like maximum runtime, may be modified (=shortened) even after the job started
  - `scontrol update JobID=`*`jobid`* `QOS=short`
  - `scontrol update JobID=`*`jobid`* `TimeLimit=1-00:00:00`
  - `scontrol update JobID=`*`jobid`* `NumNodes=2`
  - `scontrol update JobID=`*`jobid`* `Features=2TB`

UPPSALA
UNIVERSITET

When a job goes wrong

- **`scancel`**
  - *`jobid`*
  - `-u` *`username`* - to cancel all your jobs
  - `-t` *`state`* - cancel pending or running jobs
  - `-n` *`name`* - cancel jobs with a given name
  - `-i` - ask for confirmation

Priority

- Roughly:
- The first job of the day has elevated priority
- Other normal jobs run in the order of submission (subject to scheduling)
- Projects exceeding their allocation get successively into the lower priority category
- Bonus jobs run after the jobs in the higher priority categories

UPPSALA
UNIVERSITET

Priority

- In practice:
  - submit early = run early
  - bonus jobs always run eventually, but may need to wait until the night or weekend

UPPSALA
UNIVERSITET

Hands-on #2:
`sbatch/squeue/scancel/scontrol/jobinfo`

- submit a new job; note the job ID
- check all your running jobs
- what is the priority or your recently-submitted job?
- submit a new job to run for 24h; note the job ID
- modify the name of the job to “wrongjob” and the maximum runtime to 7days, for example
- cancel your job with name “wrongjob”

UPPSALA
UNIVERSITET

Different flavours of Slurm

Job script examples and workflows

UPPSALA
UNIVERSITET

Simple workflow

```
#!/bin/bash
#SBATCH -J jobname
#SBATCH -A uppmax2025-2-262
#SBATCH -p pelle
#SBATCH -n 10
#SBATCH -t 10:00:00

module load software/version
module load python/3.9.5

./my-script.sh
./another-script.sh
./myprogram.exe
```

UPPSALA
UNIVERSITET

Why do we need to load modules in the job script?
or in other words:
What does module load do?

Is it needed to specify the version of the software when loading the module? Why?

UPPSALA
UNIVERSITET

Job dependencies

- `sbatch jobscript.sh` submitted job with *`jobid1`*
- `sbatch anotherjobscript.sh` submitted job with *`jobid2`*
- `--dependency=afterok:`*`jobid1`*`:`*`jobid2`* job will only start running after the successful end of jobs *`jobid1`*`:`*`jobid2`*
- very handy for clearly defined workflows
- One may also use `--dependency=afternotok:`*`jobid`* in case you'd like to resubmit a failed job, OOM for example, to a node with a higher memory: `-C 2TB` or `-C 3TB`

UPPSALA UNIVERSITET

I/O intensive jobs

- you may use the node local `/scratch` directory for jobs that are I/O intensive
- always test!

UPPSALA
UNIVERSITET

I/O intensive jobs

- you may use the node local `/scratch` directory for jobs that are I/O intensive
- always test!

	Pelle	**Bianca**
Purpose	General-purpose	Sensitive
GPU Nodes	4 nodes with 10 Nvidia L40s each 2 nodes with Nvidia H100 each	10 nodes with 2 Nvidia A100 each
Memory per node	768 GB 2 and 3 TB on the fat nodes 384 GB on the GPU nodes	128 GB 256 or 512 on the fat nodes
Local disk (`/scratch`)	1.7 TB 6.9 TB on the fat and GPU nodes	4 TB

- *Remember:*

```
sinfo -o "%P %l %z %c %f %G %N %m"
sinfo -o "%10P %20l %10z %10c %20f %20G %20N %20m"
```

UPPSALA UNIVERSITET

I/O intensive jobs: $TMPDIR or $SNIC_TMP (/scratch)

```
#!/bin/bash
#SBATCH -J jobname
#SBATCH -A uppmax2025-2-262
#SBATCH -p pelle
#SBATCH -n 1
#SBATCH -t 10:00:00

module load bioinfotools
module load bwa/0.7.17 samtools/1.14

export SRCDIR=$HOME/path-to-input

cp $SRCDIR/foo.pl $SRCDIR/bar.txt $SNIC_TMP/.
cd $SNIC_TMP

./foo.pl bar.txt

cp *.out $SRCDIR/path-to-output/.
```

UPPSALA UNIVERSITET

OpenMP or multi-threaded job

```
#!/bin/bash
#SBATCH -A uppmax2025-2-262
#SBATCH -p pelle
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=20
#SBATCH -t 01:00:00

module load uppasd
export OMP_NUM_THREADS=20

sd > out.log
```

UPPSALA
UNIVERSITET

GPU nodes on Pelle

```
#!/bin/bash
#SBATCH -A uppmax2025-2-262
#SBATCH -p gpu
#SBATCH --gpus=1
#SBATCH --gpus-per-node=1
#SBATCH --gres=l40s:1
#SBATCH -t 01:00:00

echo $CUDA_VISIBLE_DEVICES
nvidia-smi
```

UPPSALA
UNIVERSITET

GPU nodes on Pelle

```
#!/bin/bash
#SBATCH -A uppmax2025-2-262
#SBATCH -p gpu
#SBATCH --gpus=1
#SBATCH --gpus-per-node=1
#SBATCH --gres=l40s:1
#SBATCH -t 01:00:00

echo $CUDA_VISIBLE_DEVICES
nvidia-smi
```

Remember:

```
$ sinfo -o "%10P %20l %10z %10c %20G %20N" -p gpu
PARTITION  TIMELIMIT           S:C:T     CPUS      GRES                 NODELIST
gpu        2-00:00:00          2:16:2    64        gpu:h100:2(S:1)      p[205-206]
gpu        2-00:00:00          2:16:2    64        gpu:l40s:10(S:0-1)   p[201-204]
```

UPPSALA UNIVERSITET

Running on several nodes: MPI jobs

```
#!/bin/bash
#SBATCH -J rsptjob
#SBATCH —mail-type=FAIL
#SBATCH -A uppmax2025-2-262
#SBATCH -t 00-07:00:00
#SBATCH -n 80

module load RSPt/2024-10-04
export RSPT_SCRATCH=$SNIC_TMP

srun -n 80 rspt

rm -f apts dmft_lock_file e_entropy efgArray.dat.0 efgData.out.0
energy_matrices eparm_last interstitialenergy jacob1 jacob2
locust.* out_last pot_last rspt_fft_wisdom.* runs.a symcof_new
```

Job arrays

- Submit many jobs at once with the same or similar parameters
- Use `$SLURM_ARRAY_TASK_ID` in the script in order to find the correct path

```
#!/bin/bash
#SBATCH -A uppmax2025-2-262
#SBATCH -n 40
#SBATCH -t 01:00:00
#SBATCH -J jobarray
#SBATCH --array=0-19
#SBATCH --mail-type=ALL,ARRAY_TASKS

# SLURM_ARRAY_TASK_ID tells the script which iteration to run
echo $SLURM_ARRAY_TASK_ID

cd /pathtomydirectory/dir_$SLURM_ARRAY_TASK_ID/

srun -n 40 my-program
env
```

- You may use `scontrol` to modify some of the job arrays.

UPPSALA UNIVERSITET

Snakemake and Nextflow

- Conceptually similar, but with different flavours
- First define steps, each with an input, an output, and a command that transforms the input into output
- Then just ask for the desired output and the system will handle the rest

UPPSALA
UNIVERSITET

Hands-on #3: make it your own

- use 2 or 3 of the sample job scripts as a starting point for your own job script
- tweak them so that you run something closer to your research; or just feel free to experiment
- paste at least one of the examples in the HackMD
- great if you could add a comment what the job script is about

UPPSALA
UNIVERSITET

Determining job efficiency

- **`jostats`** - custom-made UPPMAX tool
- works currently on Bianca, Rackham, and Snowy, not yet on Pelle

UPPSALA
UNIVERSITET

Job efficiency

- **`jobstats`** - a tool in the fight for productivity
  - it works only for jobs longer than 5-15 minutes
  - `-r jobid` - check running jobs
  - `-A project` - check all recent jobs of a given project
  - **`-p jobid`** - produce a CPU and memory usage plot
  - `-M cluster` - check jobs on other cluster
  - https://docs.uppmax.uu.se/software/jobstats/

UPPSALA UNIVERSITET

Hands-on #4: `jobstats`

- Generate jobstats plots for your jobs
  - Firstly, find some job IDs from this month
  - `finishedjobinfo -m` *`username`*
  - Write down the IDs from some interesting jobs.
  - Generate the images:

    ```
    $ jobstats -p ID1 ID2 ID3
    ```
- Look at the images. You may find some interesting ones in

  `/proj/introtouppmax/labs/moreslurm/jobstatsplots/`

  ```
  $ eog *png &
  ```

UPPSALA
UNIVERSITET

Hands-on #4: `jobstats`

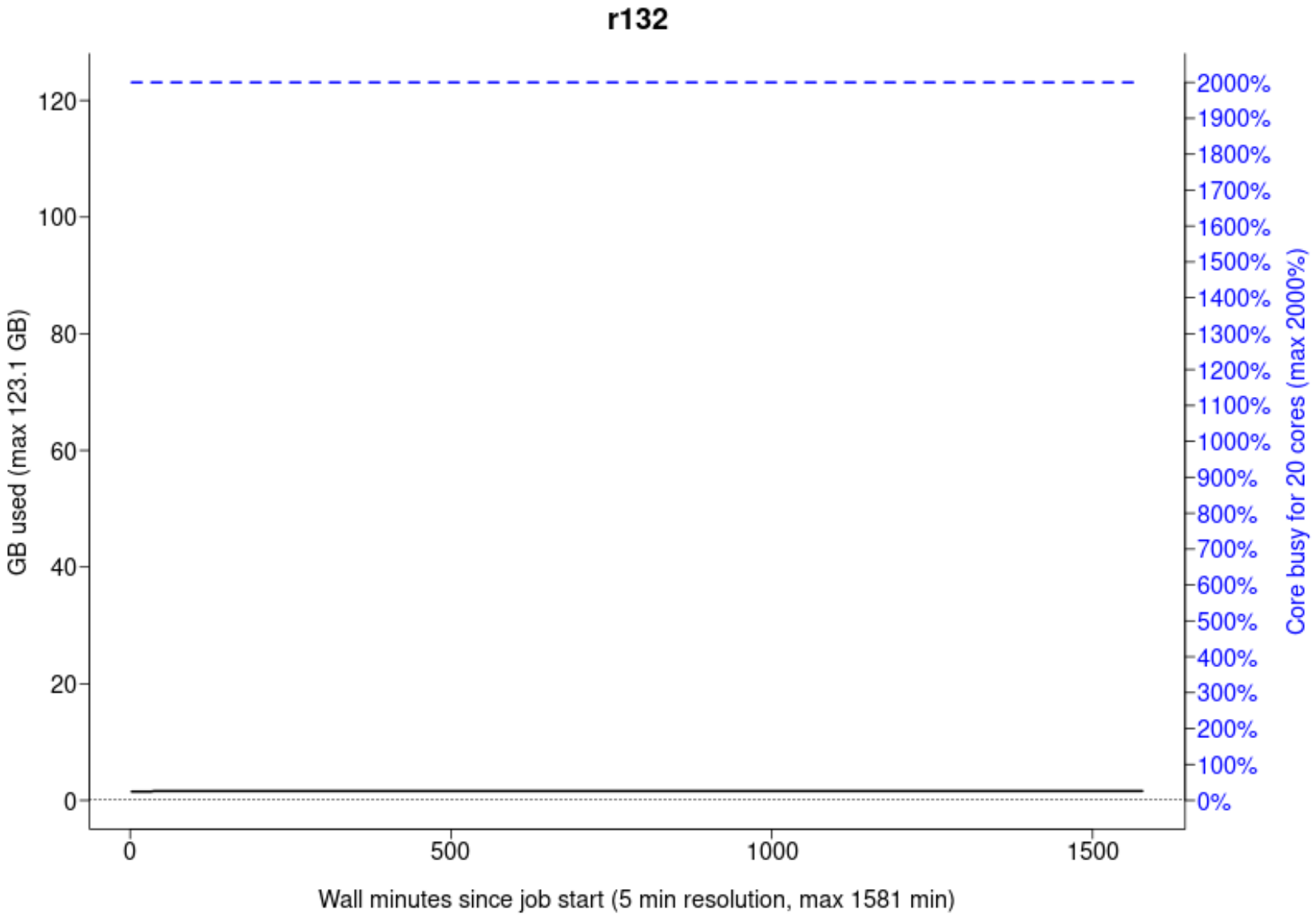
- Which of the plots in `/proj/introtouppmax/labs/moreslurm/jobstatsplots/`
  - Show good CPU or memory usage?
  - Indicate that the job requires a fat node?

UPPSALA
UNIVERSITET

11217530 COMPLETED on rackham end: 2019-12-17T12:35:44 runtime: 1-02:16:12

User:
Proj: snic2019-1-12
Jobname: fesn_x

Flags: none

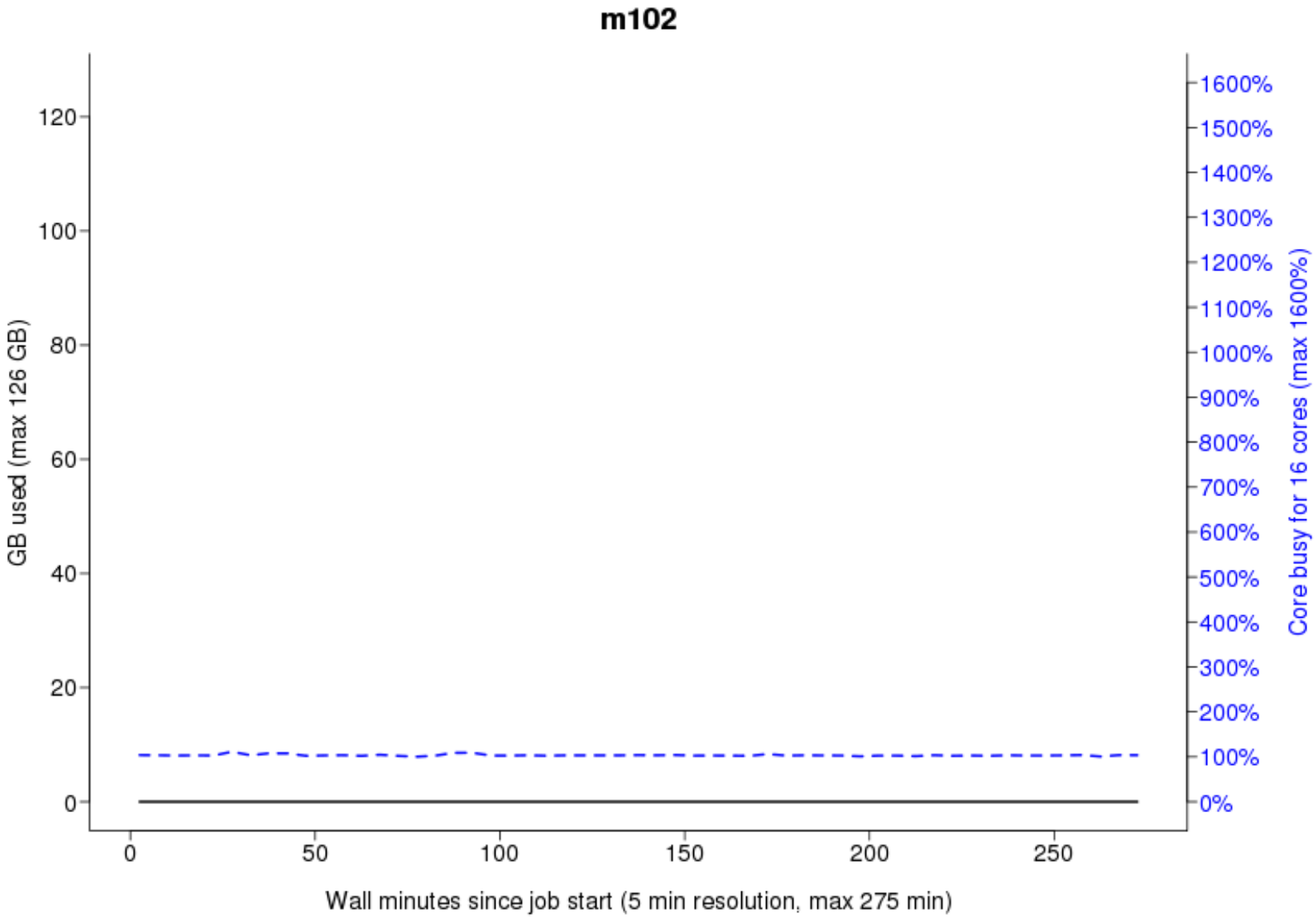


UPPSALA UNIVERSITET

8804061 COMPLETED on milou end: 2016-10-10T20:30:26 runtime: 04:36:33

User:
Proj: b2015110
Jobname: nf-MergeBam_(2)

overbooked:12%, !!half_overbooked, !!severely_overbooked,
cores_overbooked:16:2, mem_overbooked:126:0, core_mem_overbooked:15.8:0

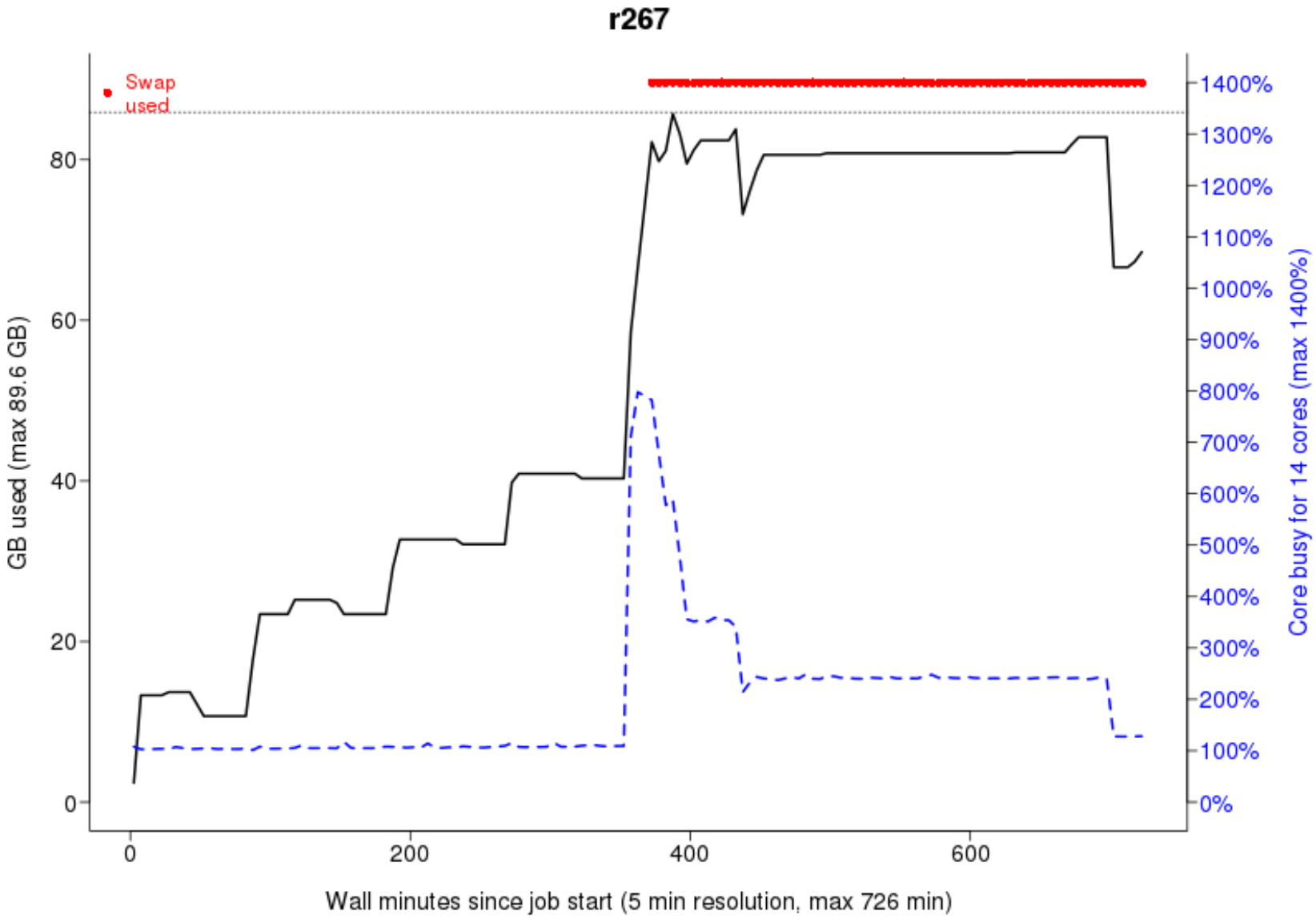


UPPSALA
UNIVERSITET

607031 OUT_OF_MEMORY on rackham end: 2018-08-25T07:29:39 runtime: 12:08:5

User:
Proj: snic2017-1-355
Jobname: pgd.apps

!!swap_used, cores_overbooked:14:11



UPPSALA UNIVERSITET

Where to go from here?

Code documentation

NAISS training newsletter or https://www.naiss.se/training/ - software-specific training events included

https://coderefinery.org/workshops/upcoming/ (Git, Jupyter, code testing, writing code documentation, …)

https://nbis.se/training/ (bio)

https://enccs.se/events/

https://supr.naiss.se/support/ or email support@uppmax.uu.se

UPPSALA UNIVERSITET