Getting Started on DiRAC systems

Ed Bennett

DiRAC Federation Workshop, 2023-06-12

Slides: https://github.com/edbennett/dirac-202306

Assumptions

- You are familiar with the Unix (bash) shell
- You have used an HPC system and batch scheduler before
- You have registered with SAFE and provided an SSH key

Connecting

- Tursa:
  ssh username@tursa.dirac.ed.ac.uk
  ed25519: SHA256:QFXBZzU5vChePHu/Y/FF42Xac7w2Shb/XT4G2+vTM48

- COSMA8:
  ssh username@login8.cosma.dur.ac.uk
  ed25519: SHA256:t50+QQcNZ6QAVCfD8n4fr4gfTT22TInc2xHke8g8ZhQ

- “Two-factor” authentication: SSH key *and* login password required

  - Passphrase-protect your keys!

Login environment

- Tursa: 1TB RAM, 128 cores
- COSMA8: 2TB RAM, 64 cores
- Only for compilation/installation, not for compute-intensive/production jobs

Software environment

- Software provided via Environment Modules
- To check available modules:
  `module avail`
- To load a module:
  `module load` *`module1_name module2_name …`*
- Tursa: barebones stack—recommended modules:
  `cuda/11.4.1 openmpi/4.1.1-cuda11.4.1 ucx/1.12.0-cuda11.4.1`
- COSMA8 stack more full-featured

Filesystems

Tursa

- Your home directory:
  /home/*project_core*/*project_code*/*username*

- To share within your project:
  /home/*project_code*/*project_code*/shared

- To share with other projects:
  /home/*project_code*/shared

Filesystems

COSMA8

- Your home directory: NFS mount, slow, small quota
  `/cosma/home/`*`project`*`/`*`username`*

- Fast Lustre storage, larger quota:
  `/cosma8/data/`*`project`*`/`*`username`*

- Super-fast scratch storage for checkpointing:
  `/snap8/scratch/`*`project`*`/`*`username`*

Filesystems

Striping

- Ensure that large files (≳1TB) are striped
    - Otherwise they can take out the filesystem
    - Unless you have a one-file-per-node write pattern
- Talk to your RSEs about how to do this

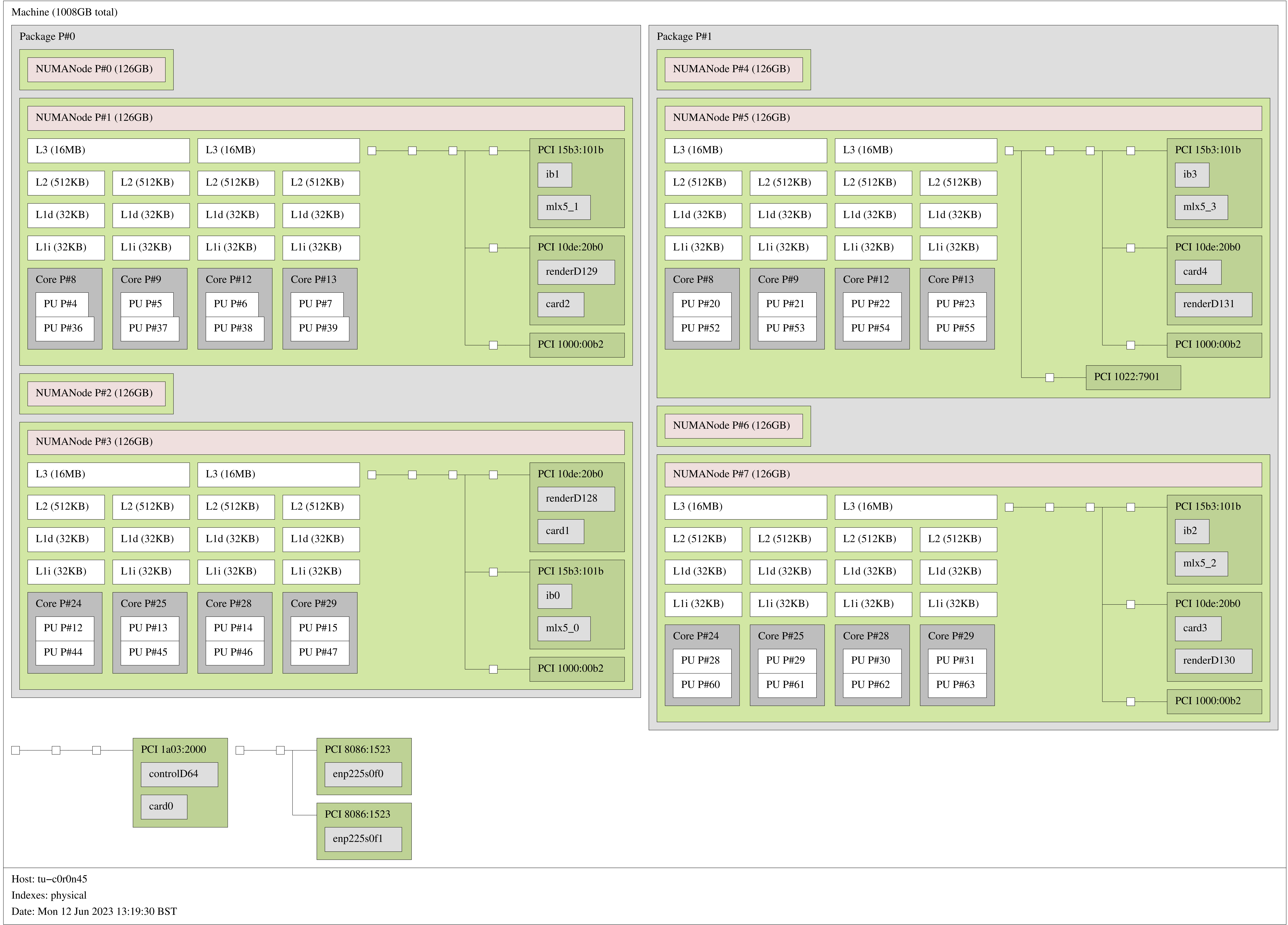
Compute nodes

Tursa

- 2 × AMD EPYC 7302
  - Each (16 cores/4 NUMA nodes/8 chiplets) × 2 hardware threads
- 4 × NVIDIA A100 [40/80GB] GPUs
- 4 × NVIDIA Networking ConnectX-6 Infiniband HDR adapters
  - Both on NUMA nodes 1, 3, 5, 7
- 1TiB RAM (128GiB per NUMA node)

Compute nodes

Tursa



Host: tu−c0r0n45

Indexes: physical

Date: Mon 12 Jun 2023 13:19:30 BST

Compute nodes

COSMA8

- 2 × AMD EPYC 7H12
  - Each (64 cores/4 NUMA nodes/8 chiplets) × 2 threads
- 1 TiB RAM (128 GiB per NUMA node)
- 1 × NVIDIA Networking ConnectX-6 Infiniband HDR adapters

Compute nodes

COSMA8

Machine (1007GB total)
Package P#0
NUMANode P#0 (125GB)
L3 (16MB) L3 (16MB) L3 (16MB) L3 (16MB)
PCI 102b:0536
card0
controlD64
L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB)
L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB)
L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB)
Core P#0 PU P#0 PU P#128
Core P#1 PU P#1 PU P#129
Core P#2 PU P#2 PU P#130
Core P#3 PU P#3 PU P#131
Core P#4 PU P#4 PU P#132
Core P#5 PU P#5 PU P#133
Core P#6 PU P#6 PU P#134
Core P#7 PU P#7 PU P#135
Core P#8 PU P#8 PU P#136
Core P#9 PU P#9 PU P#137
Core P#10 PU P#10 PU P#138
Core P#11 PU P#11 PU P#139
Core P#12 PU P#12 PU P#140
Core P#13 PU P#13 PU P#141
Core P#14 PU P#14 PU P#142
Core P#15 PU P#15 PU P#143
NUMANode P#1 (126GB)
L3 (16MB) L3 (16MB) L3 (16MB) L3 (16MB)
PCI 1022:7
sda
L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB)
L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB)
L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB)
Core P#16 PU P#16 PU P#144
Core P#17 PU P#17 PU P#145
Core P#18 PU P#18 PU P#146
Core P#19 PU P#19 PU P#147
Core P#20 PU P#20 PU P#148
Core P#21 PU P#21 PU P#149
Core P#22 PU P#22 PU P#150
Core P#23 PU P#23 PU P#151
Core P#24 PU P#24 PU P#152
Core P#25 PU P#25 PU P#153
Core P#26 PU P#26 PU P#154
Core P#27 PU P#27 PU P#155
Core P#28 PU P#28 PU P#156
Core P#29 PU P#29 PU P#157
Core P#30 PU P#30 PU P#158
Core P#31 PU P#31 PU P#159
NUMANode P#2 (126GB)
L3 (16MB) L3 (16MB) L3 (16MB) L3 (16MB)
PCI 15b3:101b
ib0
mlx5_0
L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB)
L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB)
L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB)
Core P#32 PU P#32 PU P#160
Core P#33 PU P#33 PU P#161
Core P#34 PU P#34 PU P#162
Core P#35 PU P#35 PU P#163
Core P#36 PU P#36 PU P#164
Core P#37 PU P#37 PU P#165
Core P#38 PU P#38 PU P#166
Core P#39 PU P#39 PU P#167
Core P#40 PU P#40 PU P#168
Core P#41 PU P#41 PU P#169
Core P#42 PU P#42 PU P#170
Core P#43 PU P#43 PU P#171
Core P#44 PU P#44 PU P#172
Core P#45 PU P#45 PU P#173
Core P#46 PU P#46 PU P#174
Core P#47 PU P#47 PU P#175
NUMANode P#3 (126GB)
L3 (16MB) L3 (16MB) L3 (16MB) L3 (16MB)
L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB)
L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB)
L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB)
Core P#48 PU P#48 PU P#176
Core P#49 PU P#49 PU P#177
Core P#50 PU P#50 PU P#178
Core P#51 PU P#51 PU P#179
Core P#52 PU P#52 PU P#180
Core P#53 PU P#53 PU P#181
Core P#54 PU P#54 PU P#182
Core P#55 PU P#55 PU P#183
Core P#56 PU P#56 PU P#184
Core P#57 PU P#57 PU P#185
Core P#58 PU P#58 PU P#186
Core P#59 PU P#59 PU P#187
Core P#60 PU P#60 PU P#188
Core P#61 PU P#61 PU P#189
Core P#62 PU P#62 PU P#190
Core P#63 PU P#63 PU P#191
Package P#1
NUMANode P#4 (126GB)
L3 (16MB) L3 (16MB) L3 (16MB) L3 (16MB)
PCI 14e4:165f
em1
L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB)
L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB)
L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB)
Core P#0 PU P#64 PU P#192
Core P#1 PU P#65 PU P#193
Core P#2 PU P#66 PU P#194
Core P#3 PU P#67 PU P#195
Core P#4 PU P#68 PU P#196
Core P#5 PU P#69 PU P#197
Core P#6 PU P#70 PU P#198
Core P#7 PU P#71 PU P#199
Core P#8 PU P#72 PU P#200
Core P#9 PU P#73 PU P#201
Core P#10 PU P#74 PU P#202
Core P#11 PU P#75 PU P#203
Core P#12 PU P#76 PU P#204
Core P#13 PU P#77 PU P#205
Core P#14 PU P#78 PU P#206
Core P#15 PU P#79 PU P#207
NUMANode P#5 (126GB)
L3 (16MB) L3 (16MB) L3 (16MB) L3 (16MB)
L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB)
L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB)
L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB)
Core P#16 PU P#80 PU P#208
Core P#17 PU P#81 PU P#209
Core P#18 PU P#82 PU P#210
Core P#19 PU P#83 PU P#211
Core P#20 PU P#84 PU P#212
Core P#21 PU P#85 PU P#213
Core P#22 PU P#86 PU P#214
Core P#23 PU P#87 PU P#215
Core P#24 PU P#88 PU P#216
Core P#25 PU P#89 PU P#217
Core P#26 PU P#90 PU P#218
Core P#27 PU P#91 PU P#219
Core P#28 PU P#92 PU P#220
Core P#29 PU P#93 PU P#221
Core P#30 PU P#94 PU P#222
Core P#31 PU P#95 PU P#223
NUMANode P#6 (126GB)
L3 (16MB) L3 (16MB) L3 (16MB) L3 (16MB)
L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB)
L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB)
L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB)
Core P#32 PU P#96 PU P#224
Core P#33 PU P#97 PU P#225
Core P#34 PU P#98 PU P#226
Core P#35 PU P#99 PU P#227
Core P#36 PU P#100 PU P#228
Core P#37 PU P#101 PU P#229
Core P#38 PU P#102 PU P#230
Core P#39 PU P#103 PU P#231
Core P#40 PU P#104 PU P#232
Core P#41 PU P#105 PU P#233
Core P#42 PU P#106 PU P#234
Core P#43 PU P#107 PU P#235
Core P#44 PU P#108 PU P#236
Core P#45 PU P#109 PU P#237
Core P#46 PU P#110 PU P#238
Core P#47 PU P#111 PU P#239
NUMANode P#7 (126GB)
L3 (16MB) L3 (16MB) L3 (16MB) L3 (16MB)
L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB) L2 (512KB)
L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB) L1d (32KB)
L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB) L1i (32KB)
Core P#48 PU P#112 PU P#240
Core P#49 PU P#113 PU P#241
Core P#50 PU P#114 PU P#242
Core P#51 PU P#115 PU P#243
Core P#52 PU P#116 PU P#244
Core P#53 PU P#117 PU P#245
Core P#54 PU P#118 PU P#246
Core P#55 PU P#119 PU P#247
Core P#56 PU P#120 PU P#248
Core P#57 PU P#121 PU P#249
Core P#58 PU P#122 PU P#250
Core P#59 PU P#123 PU P#251
Core P#60 PU P#124 PU P#252
Core P#61 PU P#125 PU P#253
Core P#62 PU P#126 PU P#254
Core P#63 PU P#127 PU P#255

Host: m8287.pri.cosma7.alces.network
Indexes: physical
Date: Mon 12 Jun 2023 13:35:14 BST

Slurm

`sinfo`	Get overall partition status
`squeue --me`	Get status of my jobs in queue
`sbatch` *job_script_filename*	Submit a job to the queue
`scancel` *job_id*	Cancel a running or pending job

Example job script

Tursa

Replace with `standard` in production

```
#!/bin/bash
#SBATCH --job-name NAME
#SBATCH --qos reservation
#SBATCH --time 10:00
#SBATCH --account PROJECT
#SBATCH --nodes NUM_NODES
#SBATCH --ntasks NUM_TASKS
#SBATCH --ntasks-per-node 4
#SBATCH --cpus-per-task 8
#SBATCH --partition gpu
#SBATCH --gres gpu:4
#SBATCH --output %x.%j.out
#SBATCH --error %x.%j.err
#SBATCH --reservation workshop
umask 0002
```

```
module purge
module load cuda/11.4.1 \
    openmpi/4.1.1-cuda11.4.1 \
    ucx/1.12.0-cuda11.4.1

export OMP_NUM_THREADS=4
export OMPI_MCA_btl=^uct,openib
export UCX_TLS=gdr_copy,rc,rc_x,sm,cuda_copy,cuda_ipc
export UCX_RNDV_SCHEME=put_zcopy
export UCX_RNDV_THRESH=16384
export UCX_IB_GPU_DIRECT_RDMA=yes
export UCX_MEMTYPE_CACHE=n

mpirun -np ${SLURM_NTASKS} \
       --bind-to none \
       ./wrapper.sh ./my_tool
```

Wrapper script

Tursa

- Bind processes to GPUs and fabric adapters with direct connection

```
#!/bin/bash

lrank=$OMPI_COMM_WORLD_LOCAL_RANK
numa1=$(( 2 * $lrank))
numa2=$(( 2 * $lrank + 1 ))
netdev=mlx5_${lrank}:1

export CUDA_VISIBLE_DEVICES=$OMPI_COMM_WORLD_LOCAL_RANK
export UCX_NET_DEVICES=mlx5_${lrank}:1
BINDING="--interleave=$numa1,$numa2"

echo "`hostname` - $lrank device=$CUDA_VISIBLE_DEVICES binding=$BINDING"

numactl ${BINDING}  $*
```

Example job script

COSMA8

```
#!/bin/bash
#SBATCH --job-name NAME
#SBATCH --account PROJECT
#SBATCH --partition cosma8
#SBATCH --nodes NUM_NODES
#SBATCH --ntasks NUM_TASKS
#SBATCH --cpus-per-task 8
#SBATCH --time 10:00
#SBATCH --exclusive
#SBATCH --reservation onboarding

module purge
module load ucx/1.13.0rc2 oneAPI/2022.3.0

mpirun ./my_tool
```

Project codes

dp287	CompBioMed
dp288	HECBioSim
dp289	MRCGlasgow
dp290	UKAEA
dp291	Materials Science/PAX
dp292	BritLLM

Other thoughts

- We are all sharing a reservation
  - 16 nodes on COSMA8, 32 nodes on Tursa
  - Think before running long large jobs
- Tools like Miniconda, NVIDIA HPC SDK can be installed in your user space
- You can SSH into nodes that you have active jobs on
  - e.g. To run `nvidia-smi`

More details

- Tursa user guide:
  https://epcced.github.io/dirac-docs/tursa-user-guide
- COSMA8 user guide:
  https://www.dur.ac.uk/icc/cosma/support/cosma8/