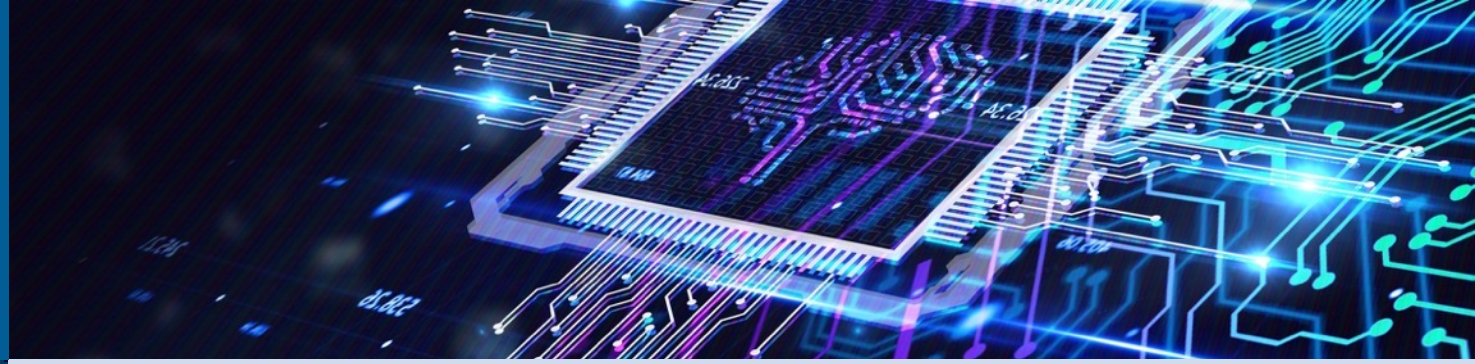
CSC

ICT Solutions for Brilliant Minds

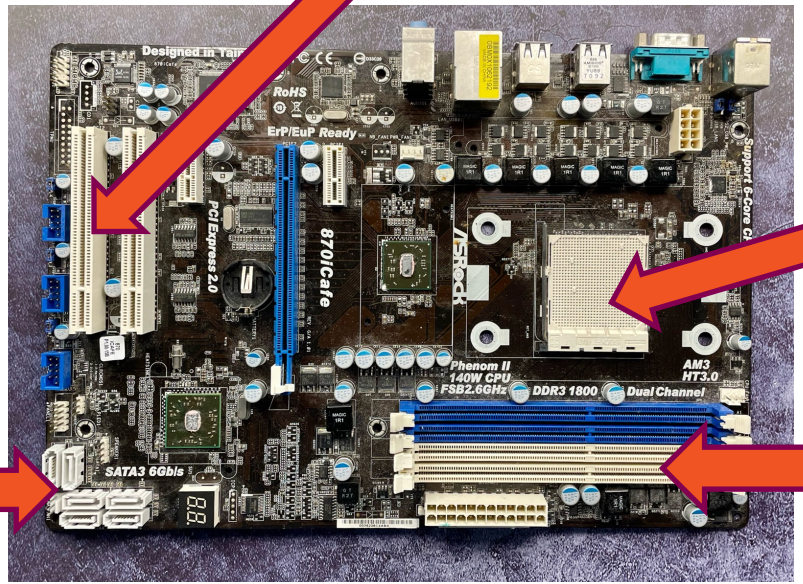


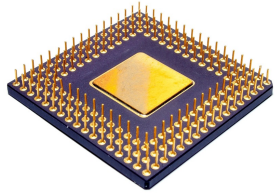
Writing batch job scripts: Introduction and allocating CPU resources

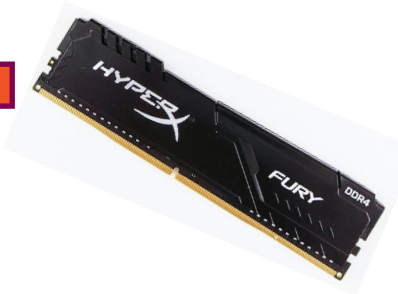
Maria Dimitrova, CSC

The anatomy of a computer

- CPU
- RAM memory
- Disk space
- GPU







Photos by: Maria Dimitrova / CSC; Walter Bichler from Pixabay; Andrey Matveev on Unsplash; Evan-Amos on Wikimedia; Andrey Matveev on Unsplash

Login nodes vs compute nodes

- Calculations on supercomputers are done on dedicated compute nodes. They are not directly accessible from outside.
- Users connect to the supercomputer using ssh to a login node
- Login nodes do not run any calculations
- There are only two login nodes, so make sure you don’t occupy them with anything heavy!

Serial jobs

- Some jobs can only be run on one CPU core – they are called **serial jobs**
- Serial jobs should be run on a laptop or another computer unless they are either very small, e.g., a script to process some calculation output or to make a plot
  - Exception: some serial jobs might need much more memory than a laptop can offer

Parallel jobs

- Many programs can split calculations among multiple CPU cores – they are called **parallel jobs**
- Parallel jobs may run in **shared-memory mode** where all cores see all the working memory
  - They can only work on one node
- **MPI jobs**
  - Each CPU core reserves memory for its own portion of the calculation
  - Cores can interact with each other if needed
  - They can be spread to as many nodes as needed
    - **Adding more nodes does not always make the job faster!**

The SLURM queuing system

- A workload manager that allocates computing resources in a "fair" and optimal manner
  - It keeps the compute nodes as busy as possible
  - It ensures that a single user does not take over all the resources
  - Job priority is determined by the length of the job, the number of CPU cores, the amount of memory, disk space and GPUs requested in the batch job script
- Provides accounting data for estimating job efficiency

The structure of SLURM scripts

- An ordinary bash script gives the commands to be executed on the compute nodes
  - E.g., what software to run
- Resources are allocated in lines starting with #SBATCH

General parameters in a script

- A specific name for the job
- The billing project has to be defined
  - Check my.csc.fi or type `csc-workspaces`
- The time to execute the job on the compute node is given in the format:
  days-hours:minutes:seconds
  - E.g., 1-06:00:00
  - Equivalently: 30:00:00

```
#!/bin/bash
#SBATCH --job-name=myTest
#SBATCH --account=<project>
#SBATCH --time=02:00:00
#SBATCH --mem-per-cpu=2G
#SBATCH --partition=small

module load myprog/1.2.3

srun myprog -i input -o output
```

https://docs.csc.fi/computing/running/creating-job-scripts-puhti/

Partition limitations

- There are multiple partitions with different limitations regarding the amount of resources that can be allocated
- Details can be seen using

```
$ sjstat –c
```

- https://docs.csc.fi/computing/running/batch-job-partitions/

Scheduling pool data:

Pool	Memory	Cpus	Total	Usable	Free	Other Traits
small*	382000Mb	40	92	92	0	type_l
small*	190000Mb	40	478	478	1	type_m
small*	190000Mb	40	44	44	0	type_m,type_io
small*	382000Mb	40	38	38	0	type_l,type_io
large	382000Mb	40	92	92	0	type_l
large	190000Mb	40	478	478	1	type_m
large	190000Mb	40	44	44	0	type_m,type_io
large	382000Mb	40	38	38	0	type_l,type_io
test	190000Mb	40	6	6	5	type_m
longrun	382000Mb	40	92	92	0	type_l
longrun	190000Mb	40	478	478	1	type_m
longrun	190000Mb	40	44	44	0	type_m,type_io
longrun	382000Mb	40	38	38	0	type_l,type_io
fmi	190000Mb	80	238	238	55	type_m_ht
fmitest	190000Mb	80	2	2	2	type_m_ht
hugemem	764000Mb	40	12	12	0	type_xl,type_io
hugemem	1532000Mb	40	6	6	0	type_bigmem
hugemem_l	764000Mb	40	12	12	0	type_xl,type_io
hugemem_l	1532000Mb	40	6	6	0	type_bigmem
gputest	382000Mb	40	2	1	1	type_gpu
gpu	382000Mb	40	78	78	6	type_gpu
interacti	190000Mb	40	4	4	0	type_m,type_io
interacti	382000Mb	40	2	2	0	type_l,type_io

Requesting CPUs for serial and shared-memory jobs

- You need to request one node:
  #SBATCH --nodes=1
  - This ensures that all CPU cores will be on the same node
- And you need to specify that all reserved CPU cores will be used for your calculation:
  #SBATCH --ntasks=1
- The number of CPU cores for the calculations are given by
  #SBATCH --cpus-per-task=20

https://docs.csc.fi/computing/running/creating-job-scripts-puhti/

Requesting CPUs for MPI jobs

- You can define the number of nodes:
  #SBATCH --nodes=2
- And you need to specify that all reserved CPU cores are to be used for your calculation:
  #SBATCH --ntasks=80

https://docs.csc.fi/computing/running/creating-job-scripts-puhti/

A note about logical and physical CPU cores

- A CPU has multiple individual cores that execute the calculations
- Some CPUs support hyperthreading
  - There are two logical cores for every physical core
  - It is not necessary more efficient to use it on a server
- On Puhti:
  20 physical cores per CPU
  2 CPUs per node
  -> 40 cores per node

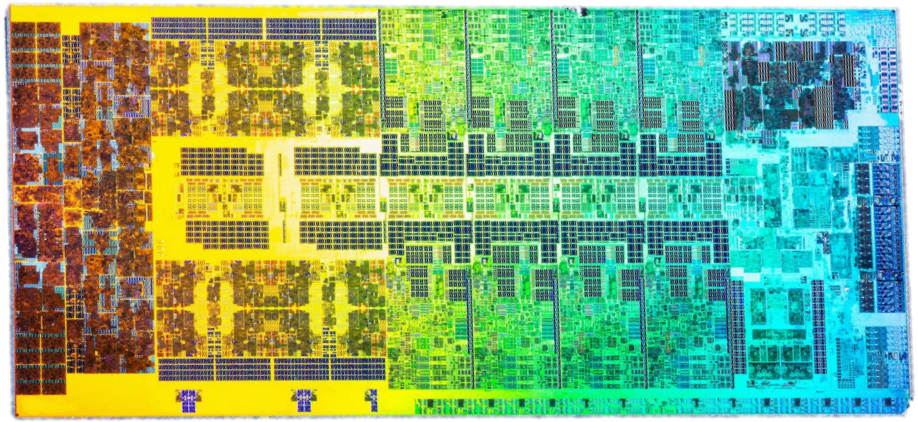


Photo by Fritzchens Fritz on Flickr: http://flic.kr/p/2nXpyah

Executing parallel jobs

- A job can be run in parallel using the command srun
  ```
  srun myprog
  ```
- Often, software will only run in serial mode unless you specify the srun command!

How to use the hardware efficiently?

- Check out the CSC training materials and courses:
  - https://www.csc.fi/training
- Materials from previous courses:
  - https://docs.csc.fi/support/training-material/
- CSC Computing Environment Moodle course:
  - https://e-learn.csc.fi/enrol/index.php?id=76

You can always ask us in case of doubt!

servicedesk@csc.fi