Introduction to HPC resources

Robert Fridman
Dave Schulz
June 11 2024

UNIVERSITY OF CALGARY
Research Computing Services

UNIVERSITY OF CALGARY

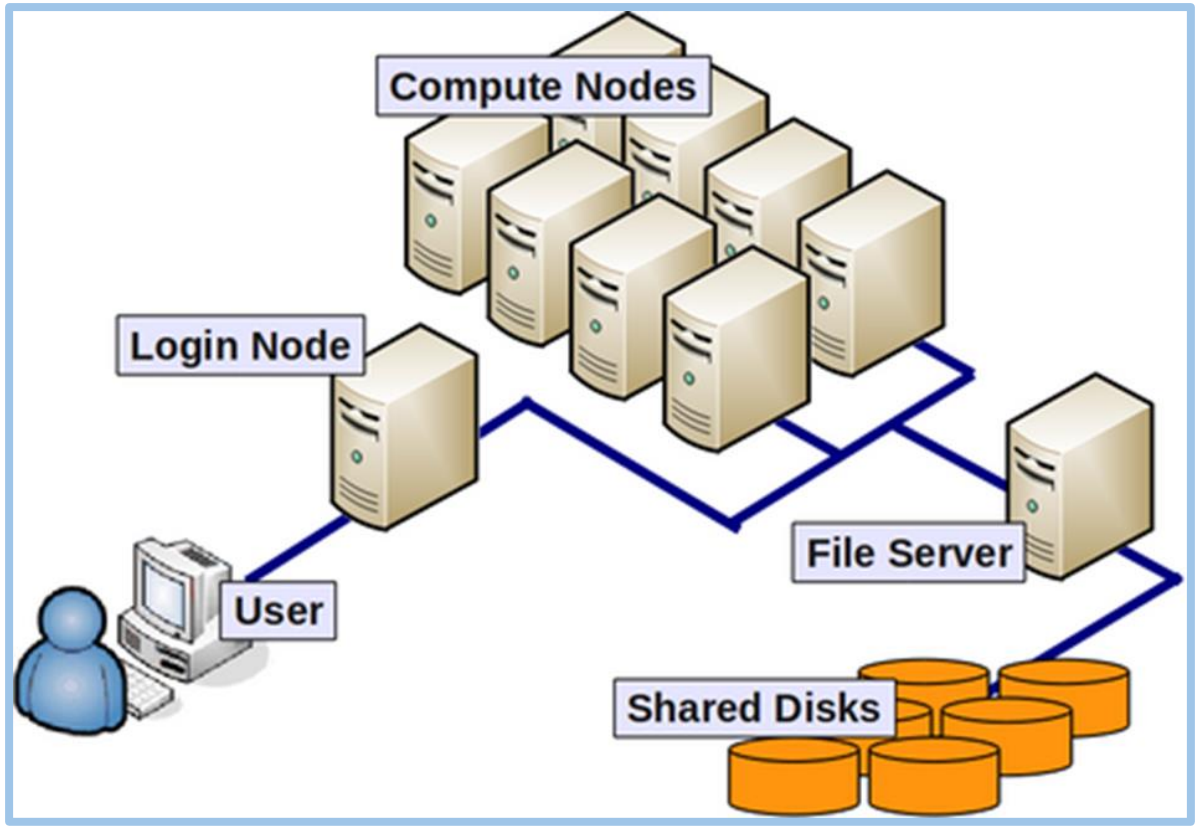
What is HPC

- High Perform Computing (HPC) is a computing paradigm that allows the resources of multiple servers to be harnessed to solve a single problem.
  - CPUs
  - Memory
  - Interconnect
  - Caches on the CPUs
- Parallel programming is a methodology of using communication libraries and data structures to allow a problem to be divided amongst separate servers, or multiple CPUs of a single server

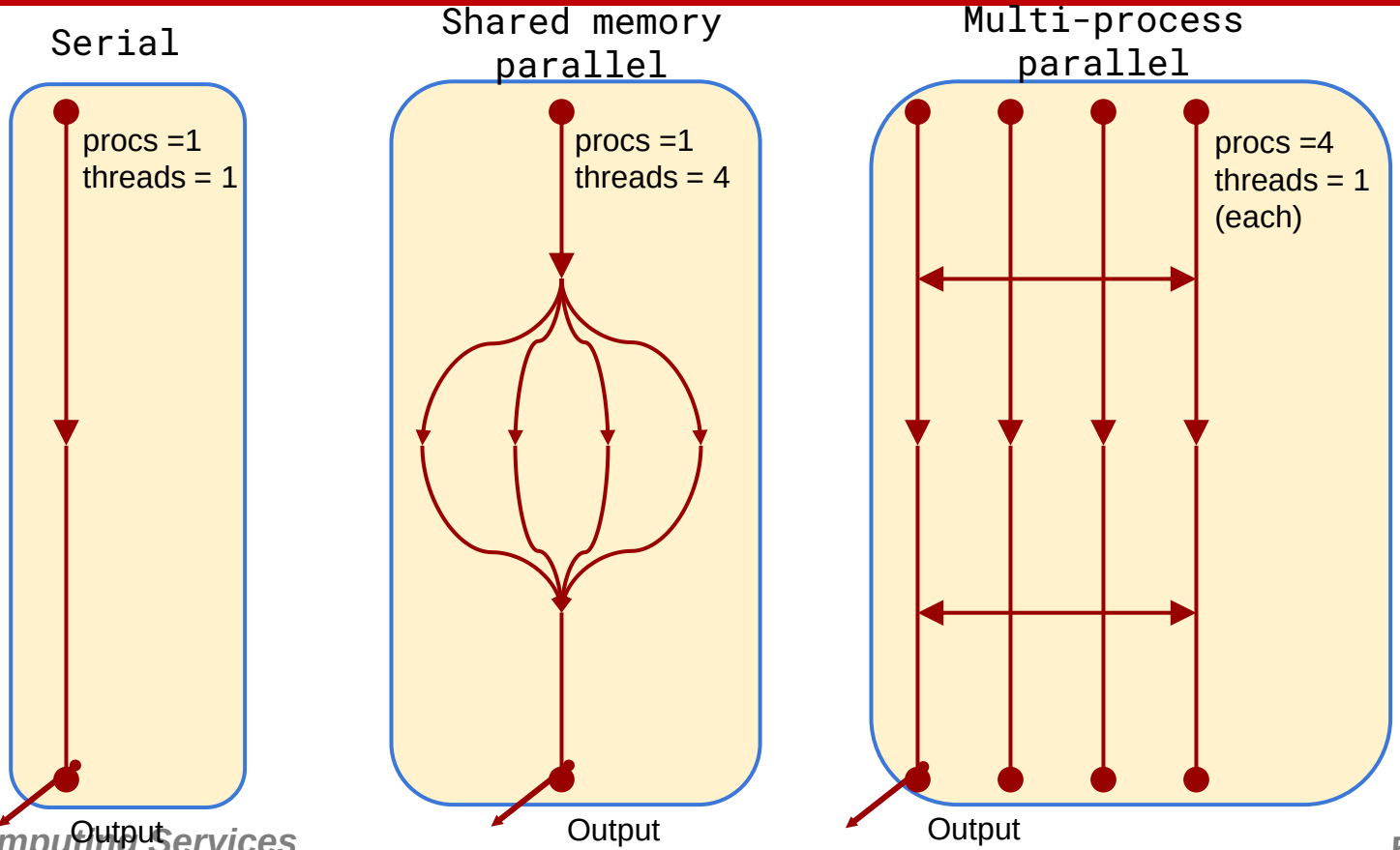
UNIVERSITY OF CALGARY

What does HPC look like?

Components of HPC



Types of Parallel Programming



UNIVERSITY OF CALGARY

Using a compute cluster

- Using **command line interface** (mostly).
- **Non-interactive** computations:
  Will run somewhere at some time.
- **calculation** vs. **job**
  Job script: requests resources and runs your calculation.
- **Postal office** model:
  Submit a job and **leave**, it will run when there are resources.
- Cluster **limits**:
  maximum run time, storage quotas, maximum number of jobs.
- Cluster **policies**.

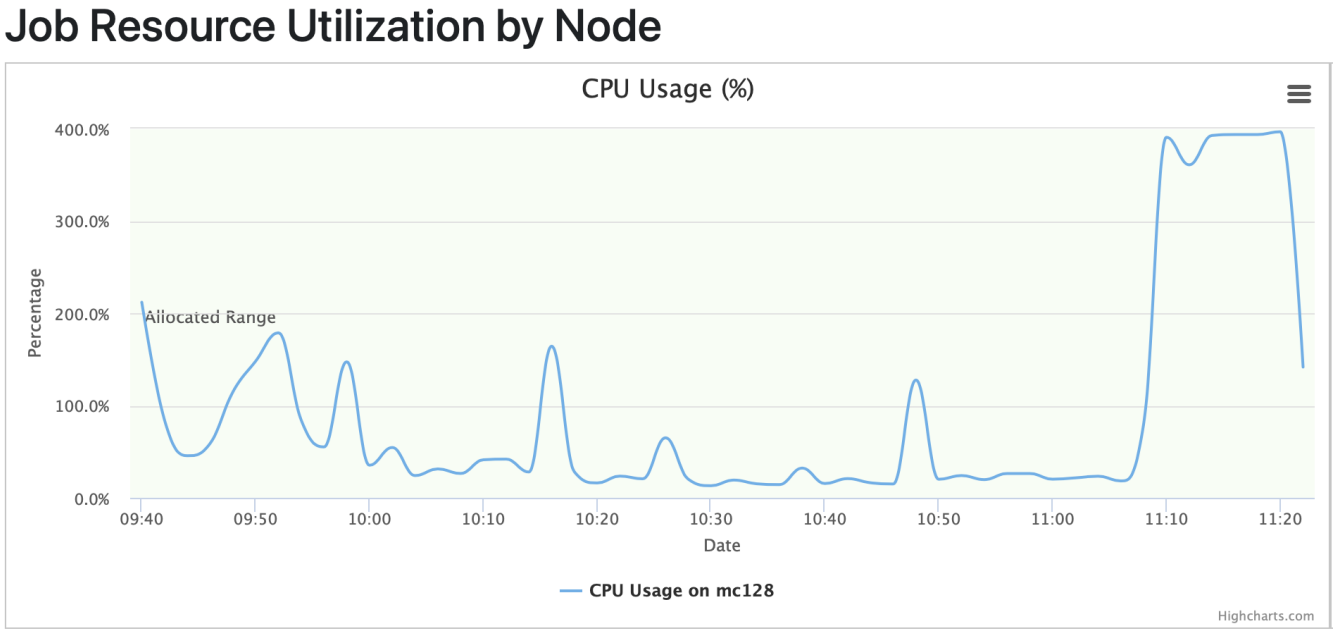
UNIVERSITY OF CALGARY

Steps to run a computation on a cluster

- Upload your **initial data** to the cluster.
- **Login** to the cluster.
- Prepare a **job script**.
- **Submit** the job script to the job scheduler.
- **Disconnect** from the cluster.
- **Login** to the cluster to periodically check on the job's progress
- **Check**, if your job has been completed.

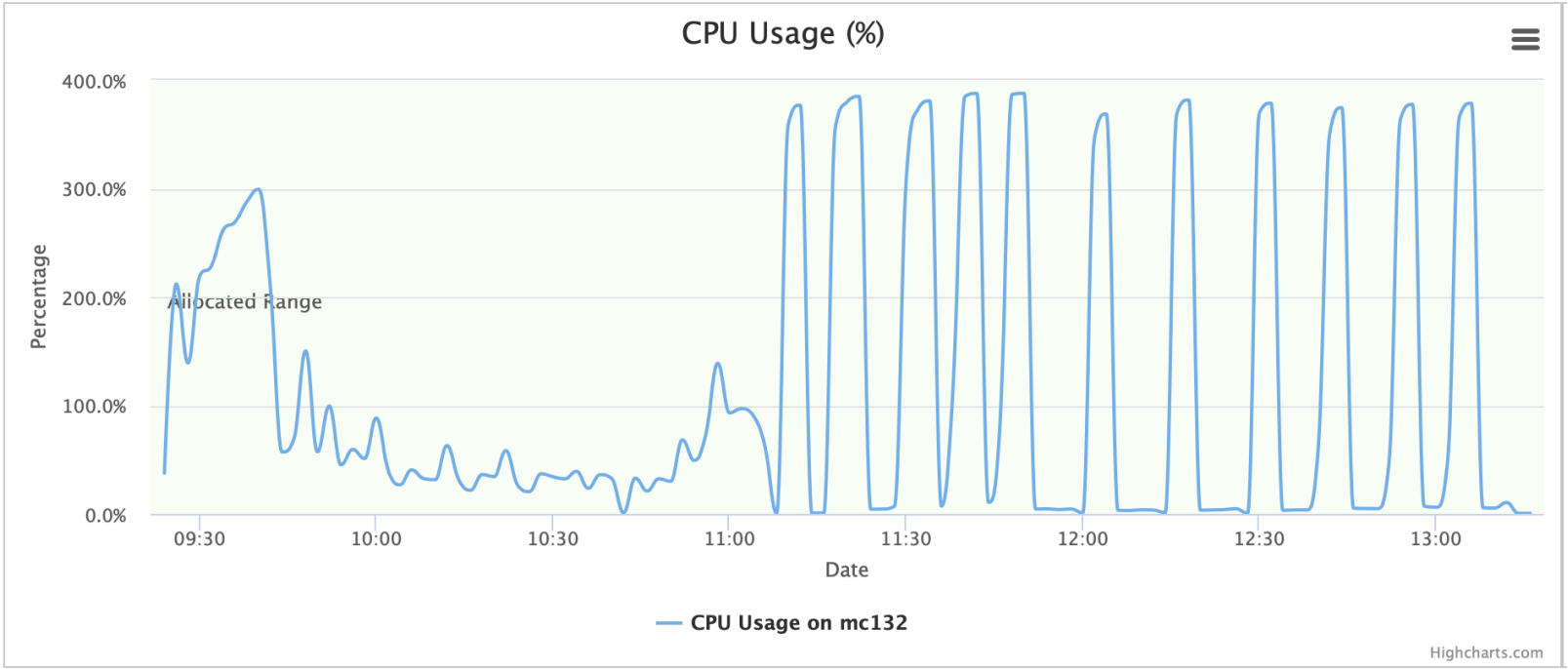
Inefficiencies

- Just because you ask for multiple CPUs does not mean that the job will utilize all of them.



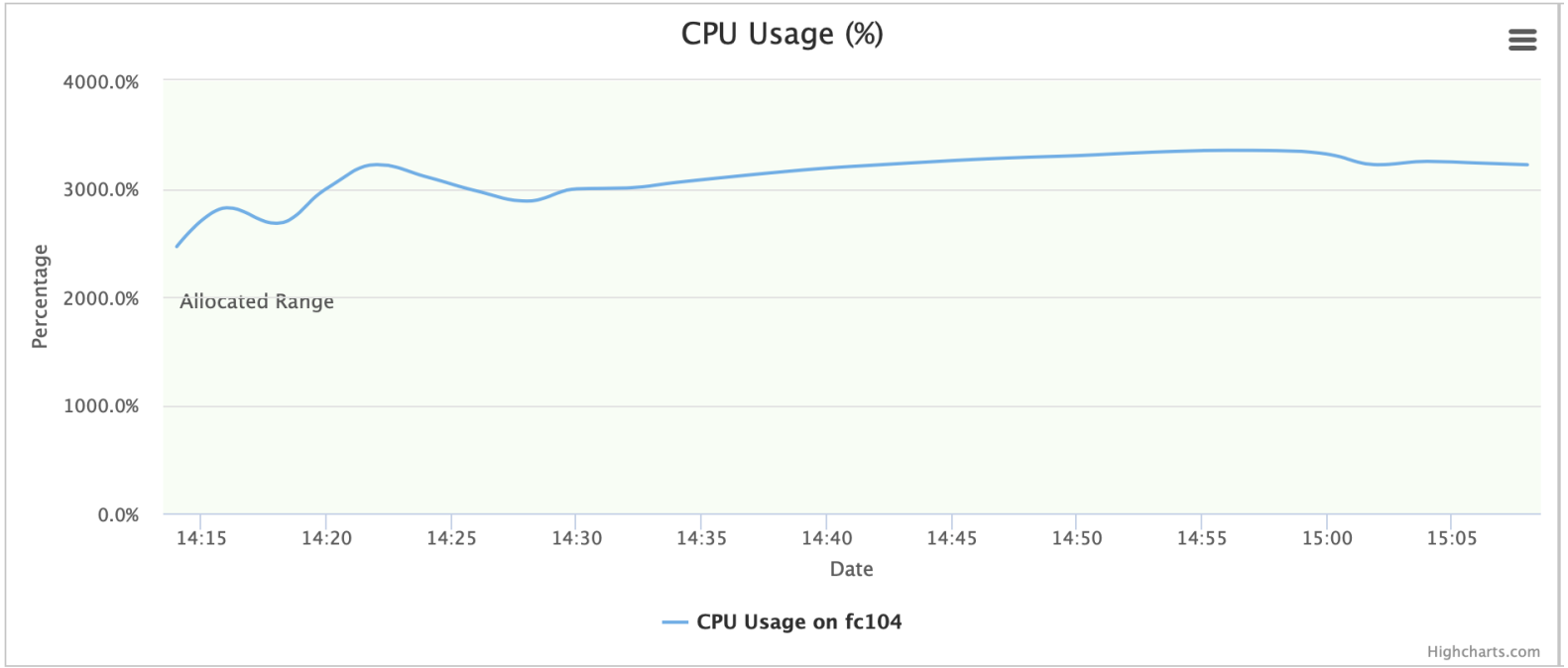
UNIVERSITY OF CALGARY

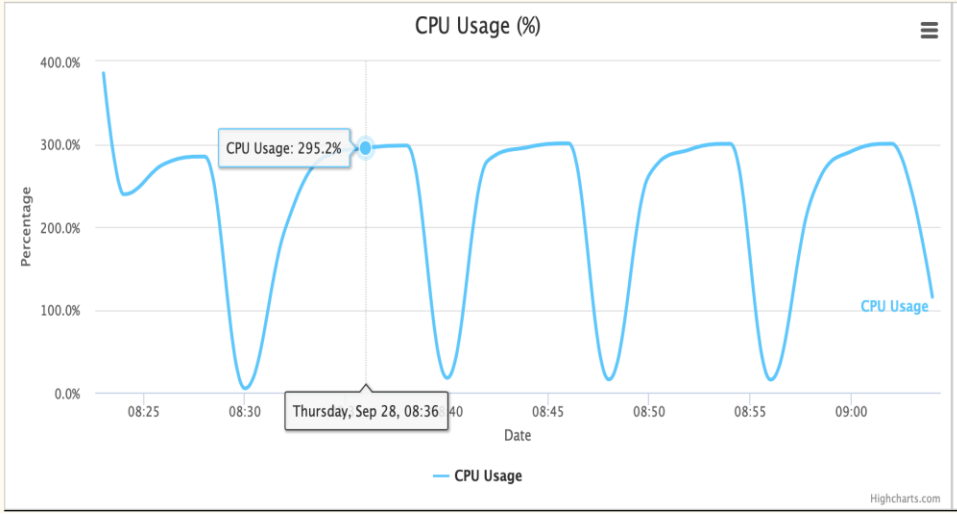
Job Resource Utilization by Node

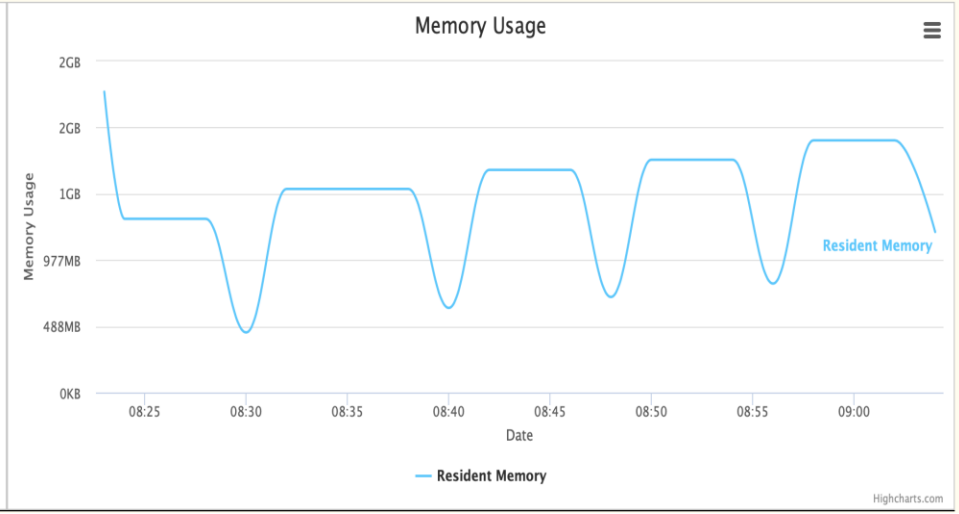


UNIVERSITY OF CALGARY

Job Resource Utilization by Node




CPU Usage (%)
400.0%
300.0%
200.0%
100.0%
0.0%
Percentage
CPU Usage: 295.2%
Thursday, Sep 28, 08:36
08:25
08:30
08:45
08:50
08:55
09:00
Date
CPU Usage
CPU Usage
Highcharts.com


Memory Usage
2GB
2GB
1GB
977MB
488MB
0KB
Memory Usage
Resident Memory
08:25
08:30
08:35
08:40
08:45
08:50
08:55
09:00
Date
Resident Memory
Highcharts.com

UNIVERSITY OF CALGARY

The job scheduler - SLURM

- The **job scheduler** is a software that manages jobs on a compute cluster. Typically it has a **job manager** and a **resource manager**.
- maintains a **database of jobs**,
- **enforces policies** regarding limits and priorities,
- ensures resources are **not overloaded**,
  for example by only assigning each CPU core to one job at a time,
- **decides** which jobs to run and on which compute nodes,
- **starts** them on those nodes, and
- **cleans up** after each job finishes.

If you want to **run something** on a cluster you have to **talk to the scheduler**.

UNIVERSITY OF CALGARY

Functions of a job scheduler

- Provides **fairness** of use: your priority in the scheduling queue is based on how much you have used the resource in the past. Wait times in the queue generally depend on how many resources you are asking for.
- You request resources for your job and the scheduler knows where to find them.
- Maintains the **queue** of jobs that is based on history of use.
- Works continuously to manage the jobs and resources, freeing you from spending time manually finding where to run your job. Your jobs will start automatically when resources become available.
- Enforces reasonable limits on jobs.

SLURM commands

- Submit job:
  $ **sbatch** job_script.slurm
- Delete job:
  $ **scancel** Job_ID
- Check the queue:
  $ **squeue**
  $ **squeue** -u UserName
- Allocate an interactive node:
  $ **salloc** …

- Get job information:
  $ **scontrol** show job Job_ID
- Information about nodes:
  $ **scontrol** show nodes
  $ **sinfo** -N
- Accounting on finished jobs:
  $ **sacct** -j Job_ID
  $ **seff** Job_ID

UNIVERSITY OF CALGARY

SLURM limits on ARC

- **Run times**:
  - up to **7 days** on general resources;
  - up to **1 day** for special resources (big memory, GPUs).
- Max number of **jobs** in the system:
  - **4000**
- Max number of **CPUs** running:
  - depends on **partition** (may change without warning).
  - per-Job and per-User limits.
  - `arc.limits`

UNIVERSITY OF CALGARY

Submitting a SLURM job

```
$ sbatch my_job.slurm
Submitted batch job 5542551

$ squeue -j 5542551
JOBID    USER      STATE    PARTITION TIME_LIMIT  TIME  NODES TASKS CPUS
5542551 drozmano RUNNING  parallel    1:00:00   0:02   1     1         1

$ sacct -j 5542551 -o jobid,state,maxrss,maxvmsize
   JobID           State         MaxRSS  MaxVMSize
------------ ---------- ---------- ----------
5542551          TIMEOUT
5542551.ext+  COMPLETED          0          4360K
5542551.0      CANCELLED      302932K    295428K
```

UNIVERSITY OF CALGARY

SLURM job submission example

```
#!/bin/bash
# -----------------------------------------------------------------
# An example PBS script for running a job on a compute cluster
# -----------------------------------------------------------------
#SBATCH --job-name=test-job
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=1
#SBATCH --cpus-per-task=1
#SBATCH --mem=1gb
#SBATCH --time=0-06:00:00

#SBATCH --partition=lattice
# -----------------------------------------------------------------
echo "Starting run at: `date`"
# -----------------------------------------------------------------
module load python
python my_code.py input.dat

# -----------------------------------------------------------------
echo "Job finished with exit code $? at: `date`"
# -----------------------------------------------------------------
```

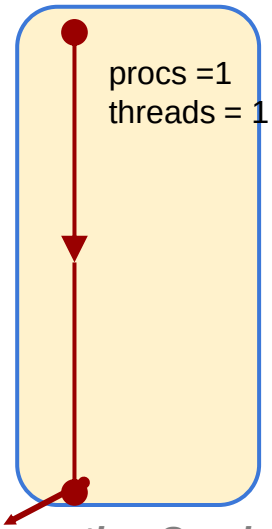
Monitoring SLURM jobs

```
$ squeue and squeue-long
$ arc.job-info job_ID
$ squeue
$ squeue-long -u username
$ squeue-long -j job_ID
$ sacct -j job_ID -o jobid,state,...
$ seff job_ID
```

SLURM resource selection

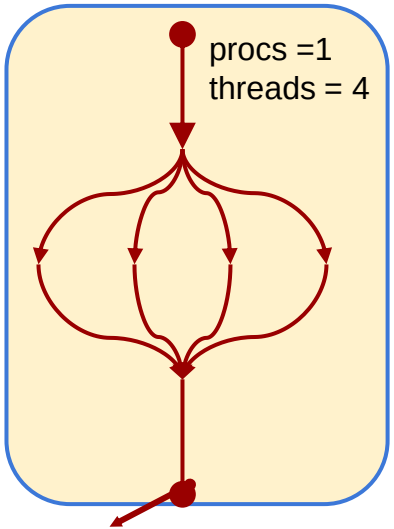
```
--nodes=1
--ntasks=1
--cpus-per-task=1
```

Serial



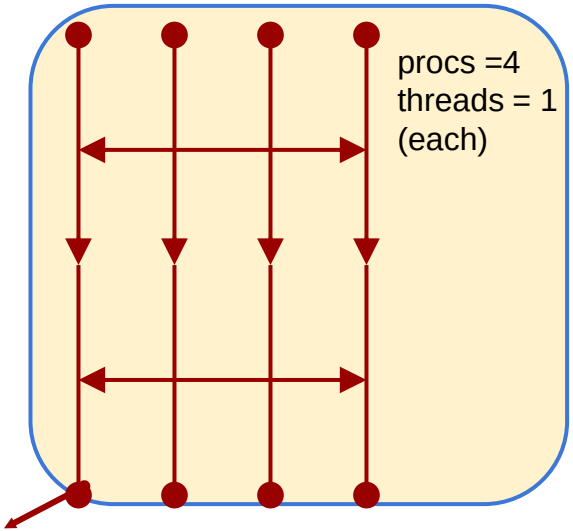

```
--nodes=1
--ntasks=1
--cpus-per-task=4
```

Shared memory parallel




```
--nodes=4
--ntasks-per-node=1
--cpus-per-task=1
```

Multi-process parallel



UNIVERSITY OF CALGARY

SLURM interactive jobs

```
[drozmano@arc ~]$ arc.nodes
....
[drozmano@arc ~]$ salloc -N1 -n1 -c8 --mem=32gb -t 3:00:00

salloc: Granted job allocation 5542532
salloc: Waiting for resource configuration
salloc: Nodes fc104 are ready for job

[drozmano@fc104 ~]$
```

UNIVERSITY OF CALGARY

Questions!

Reach us at : support@hpc.ucalgary.ca

Survey

Complete our survey for a chance to win a $50 gift card

Your feedback will help us improve our future sessions.

https://rcs.ucalgary.ca/survey

Questions?

support@hpc.ucalgary.ca