# IFB-core cluster in practice

Magali Hennion  Frédéric Jarlier

October 2024





# Table of Contents

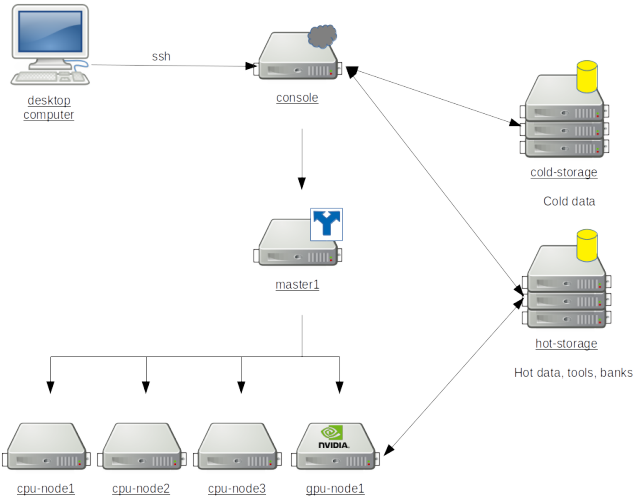


The slides are adapted from previous presentations by Alix Silvert, Magali Hennion and Julien Rey and inspired by IFB training material.

# What is a HPC cluster for ?

- High hardware resources needs
- Long running analyses
- A lot of similar analyses
- Shared work between users
- Free your desktop from heavy tasks
- Improve reproducibility

# What is a HPC cluster ?



# How to talk to it?

**SSH** Connect to the cluster using the following line in your local terminal, replacing 'username' with your personal username. Then enter your password.

```
ssh username@core.cluster.france-bioinformatique.fr
```

**Open OnDemand web interface** Open a web browser and go to https://ondemand.cluster.france-bioinformatique.fr.
Type in your login and password. Grant access.

**sftp**

- To navigate in your remote folders with a local file manager
- To edit files with your local editor

Open a file manager and go to **sftp://core.cluster.france-bioinformatique.fr**.

# Note about security

Best practices, change regularly your password

```
[username @ clust-slurm-client2]$ ~ : passwd
```

# Where you can go, write, or execute

## User environment

/shared/home/username

## Computations (hot data)

/shared/projects/projectname

## Common banks (read-only)

/shared/bank/

# Note about quotas

The amount of disk space for projects are limited by quotas. To check available space for your projects, use the following commands.

```
[mhennion @ clust-slurm-client2 10:29]$ ~ : status_bars
                  bi4edc [################-----]     810 /    1024 GB
             edc_nanopore [################----]     400 /     500 GB
             nanopore_mod [################----]     848 /    1000 GB

                     HOME [##------------------]      14 /     100 GB
Update: 2024-07-30 10:00 - Your current default account is bi4edc - More info: status_bars --help
```

```
[mhennion @ clust-slurm-client 09:25]$ ~ : gid=$(getent group 2417_wf4bioinfo | cut -f 3 -d ':')
[mhennion @ clust-slurm-client 09:26]$ ~ : lfs quota -h -p $gid /shared/projects/2417_wf4bioinfo
Disk quotas for prj 166486 (pid 166486):
     Filesystem    used   quota   limit   grace   files   quota   limit   grace
/shared/projects/2417_wf4bioinfo
                  1.336G    250G    375G       -   15746  250000  375000       -
```

All jobs **must** be launched from a **project** directory.

# About Slurm

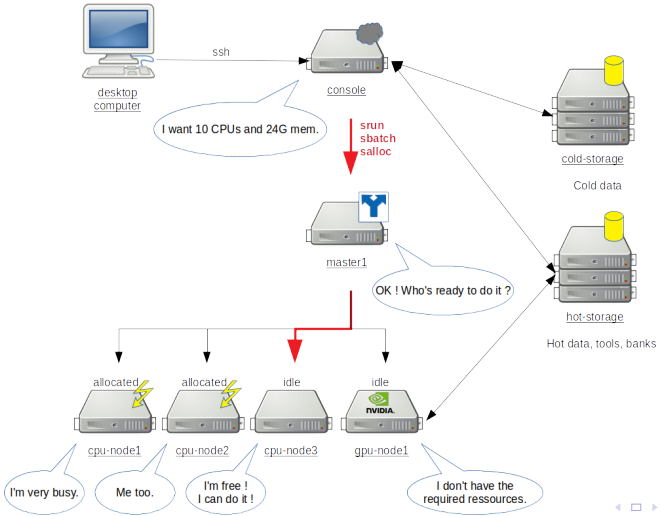

Slurm is the job scheduling system.

It is what will take your code and distribute it on the computing nodes, while ensuring they have the CPU(s) and RAM that you asked for.

It requires specific commands to run (srun, sbatch, salloc, etc...).

You **must** launch jobs with Slurm.

# Flowsheet


desktop computer
ssh
console
I want 10 CPUs and 24G mem.
srun
sbatch
salloc
cold-storage
Cold data
master1
OK ! Who's ready to do it ?
hot-storage
Hot data, tools, banks
allocated
allocated
idle
idle
NVIDIA
cpu-node1
cpu-node2
cpu-node3
gpu-node1
I'm very busy.
Me too.
I'm free !
I can do it !
I don't have the
required ressources.

## srun

Launch a (simple) interactive job.

```
[mhennion @ ifb]$ srun hostname
```

Some parameters can be added to the command line:
**–partition/-p**: request a specific partition
**–account/-A**: select the (project) account
**–cpus-per-task/-c**: request that ncpus be allocated (default: 1 cpu)
**–mem-per-cpu**: specify the required memory per cpu (default: 2GB)

Example

```
[mhennion @ ifb]$ srun -A 2417_wf4bioinfo -p fast hostname
```

# sbatch

Launch more complex jobs using scripts.

myscript.sbatch

```
#!/bin/bash
#SBATCH --partition=fast
#SBATCH --account=2417_wf4bioinfo
#SBATCH --cpus-per-task=1
#SBATCH --mem-per-cpu=4GB
#SBATCH --output=resultat.log
hostname
```

Example

```
[mhennion @ clust-slurm-client2 ]$ sbatch myscript.sbatch
```

# About associations

Cluster, account and partition must match your user's associations.
To check your associations:

```
[mhennion @ ifb]$ ~ : sacctmgr show user mhennion withassoc
      User   Def Acct     Cluster    Account  Partition     QOS
---------- ----------  ---------- ---------- ---------- ----------
  mhennion     bi4edc       core edc_nanop+       long    normal
  mhennion     bi4edc       core edc_nanop+       fast    normal
  mhennion     bi4edc       core nanopore_+       long    normal
  mhennion     bi4edc       core nanopore_+       fast    normal
  mhennion     bi4edc       core       demo       fast    normal
  mhennion     bi4edc       core     bi4edc        gpu    normal
  mhennion     bi4edc       core     bi4edc       long    normal
  mhennion     bi4edc       core     bi4edc       fast    normal
```

# squeue : list submitted jobs on the cluster

```
[mhennion @ ifb]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
          40912487    bigmem Blast_co bvouadec  R    1:21:33      1 cpu-node-69
          40891092    bigmem run_flye jorjuela  R 4-21:47:49      1 cpu-node-69
          39958757      fast Reinhard  dinaton PD       0:00      1 (QOSMaxMemoryPerUser)
          39910421      fast sys/dash     jaff PD       0:00      1 (AssocGrpCPUMinutesLimit)
          40085033      fast sys/dash   nobody PD       0:00      1 (InvalidAccount)
          40383313      fast split_fa emergen- PD       0:00      1 (launch failed requeued held)
          40912840      fast sys/dash   jcaron  R      33:39      1 cpu-node-56
          40913175      fast sys/dash mlavieri  R       6:25      1 cpu-node-50
```

Some parameters can be added to filter jobs or show more infos:

–**partition**/-**p**: specify the partition to view

–**user**/-**u**: request jobs from a list of users

–**format**/-**o**: specify the information to be displayed

```
[mhennion @ ifb]$ squeue --me -p fast
[mhennion @ ifb]$ squeue -o "%10i %.9P %.8j %.12u %.15T %.15M %.15l %.6D %.20R %.6C %.10b %.10Q"
```

# scancel

Kill a job

```
[mhennion @ ifb]$ scancel job_id
```

Cancel all my jobs

```
[mhennion @ ifb]$ scancel -u <my_user_name>
```

or

```
[mhennion @ ifb]$ scancel --me
```

# sacct

Display accounting data for your running and completed jobs:

```
[mhennion @ ifb]$ sacct --format=JobID,JobName,Start,Elapsed,CPUTime,NCPUS,NodeList,MaxRSS,ReqMeM,State
JobID           JobName                Start    Elapsed    CPUTime      NCPUS        NodeList     MaxRSS     ReqMem      State
------------ ---------- ------------------- ---------- ---------- ---------- --------------- ---------- ---------- ----------


40913048     DiffRmd-c+ 2024-07-30T10:29:16   00:00:50   00:00:50          1      cpu-node-1                    2013M  COMPLETED
40913048.ba+       batch 2024-07-30T10:29:16   00:00:50   00:00:50          1      cpu-node-1   1346696K            COMPLETED
40913049     DiffRmd-c+ 2024-07-30T10:29:28   00:00:59   00:00:59          1   cpu-node-104                    2013M  COMPLETED
40913049.ba+       batch 2024-07-30T10:29:28   00:00:59   00:00:59          1   cpu-node-104   1313508K            COMPLETED
40913087     FinalRepo+ 2024-07-30T10:30:45   00:00:29   00:00:29          1    cpu-node-57                     500M  COMPLETED
40913087.ba+       batch 2024-07-30T10:30:45   00:00:29   00:00:29          1    cpu-node-57      3272K            COMPLETED
40913090     CompRepor+ 2024-07-30T10:31:15   00:00:18   00:00:18          1    cpu-node-59                     500M  COMPLETED
40913090.ba+       batch 2024-07-30T10:31:15   00:00:18   00:00:18          1    cpu-node-59      1460K            COMPLETED
40913604       WGBSflow 2024-07-30T11:01:48   00:00:37   00:00:37          1    cpu-node-50                     500M CANCELLED+
40913604.ba+       batch 2024-07-30T11:01:48   00:00:38   00:00:38          1    cpu-node-50    105656K           CANCELLED
```

# reportseff

## Report a job efficiency

```
[mhennion @ ifb]$ module load reportseff
[mhennion @ ifb]$ reportseff 40913604
     JobID    State       Elapsed  TimeEff   CPUEff   MemEff
  40913604  CANCELLED     00:00:37   0.0%     16.2%    20.6%
```

## Analyse a folder of slurm outputs

```
[mhennion @ ifb]$ reportseff --format "+Start,CPUTime,NCPUS,NodeList,MaxRSS,ReqMeM" --modified-sort slurm_output/
```

| JobID | State | Elapsed | TimeEff | CPUEff | MemEff | Start | CPUTime | NCPUS | NodeList | MaxRSS | ReqMem |
|---|---|---|---|---|---|---|---|---|---|---|---|
| multiqc--41907184.out | COMPLETED | 00:00:07 | 2.3% | 57.1% | 1.6% | 2024-09-24T12:23:17 | 00:00:07 | 1 | cpu-node-45 | 3184K | 191M |
| fastqc-sample=SRR3099585_chr18-41907179.out | COMPLETED | 00:00:25 | 8.3% | 64.0% | 1.1% | 2024-09-24T12:22:47 | 00:00:25 | 1 | cpu-node-38 | 3284K | 286M |
| fastqc-sample=SRR3105697_chr18-41907180.out | COMPLETED | 00:00:15 | 5.0% | 80.0% | 1.1% | 2024-09-24T12:22:47 | 00:00:15 | 1 | cpu-node-39 | 3260K | 286M |
| fastqc-sample=SRR3099587_chr18-41907181.out | COMPLETED | 00:00:14 | 4.7% | 85.7% | 1.1% | 2024-09-24T12:22:47 | 00:00:14 | 1 | cpu-node-40 | 3200K | 286M |
| fastqc-sample=SRR3105698_chr18-41907183.out | COMPLETED | 00:00:13 | 4.3% | 69.2% | 1.1% | 2024-09-24T12:22:47 | 00:00:13 | 1 | cpu-node-40 | 3192K | 286M |
| fastqc-sample=SRR3099586_chr18-41907178.out | COMPLETED | 00:00:12 | 4.0% | 66.7% | 1.1% | 2024-09-24T12:22:47 | 00:00:12 | 1 | cpu-node-45 | 3208K | 286M |
| fastqc-sample=SRR3105699_chr18-41907182.out | COMPLETED | 00:00:12 | 4.0% | 66.7% | 1.1% | 2024-09-24T12:22:47 | 00:00:12 | 1 | cpu-node-40 | 3260K | 286M |
| multiqc--41907169.out | OUT_OF_MEMORY | 00:00:06 | 2.0% | 16.7% | 3.3% | 2024-09-24T12:21:06 | 00:00:06 | 1 | cpu-node-45 | 3240K | 95M |
| fastqc-sample=SRR3099585_chr18-41907159.out | COMPLETED | 00:00:33 | 11.0% | 54.5% | 75.7% | 2024-09-24T12:20:25 | 00:00:33 | 1 | cpu-node-38 | 221708K | 286M |
| fastqc-sample=SRR3105697_chr18-41907160.out | COMPLETED | 00:00:19 | 6.3% | 73.7% | 1.1% | 2024-09-24T12:20:25 | 00:00:19 | 1 | cpu-node-39 | 3212K | 286M |
| fastqc-sample=SRR3105698_chr18-41907163.out | COMPLETED | 00:00:17 | 5.7% | 82.4% | 1.1% | 2024-09-24T12:20:25 | 00:00:17 | 1 | cpu-node-40 | 3224K | 286M |
| fastqc-sample=SRR3099587_chr18-41907161.out | COMPLETED | 00:00:15 | 5.0% | 80.0% | 1.1% | 2024-09-24T12:20:25 | 00:00:15 | 1 | cpu-node-40 | 3192K | 286M |
| fastqc-sample=SRR3105699_chr18-41907162.out | COMPLETED | 00:00:14 | 4.7% | 71.4% | 1.1% | 2024-09-24T12:20:25 | 00:00:14 | 1 | cpu-node-40 | 3244K | 286M |
| fastqc-sample=SRR3099586_chr18-41907158.out | COMPLETED | 00:00:15 | 5.0% | 53.3% | 1.1% | 2024-09-24T12:20:25 | 00:00:15 | 1 | cpu-node-45 | 3304K | 286M |
| multiqc--41907141.out | COMPLETED | 00:00:24 | 0.3% | 25.0% | 0.6% | 2024-09-24T12:15:57 | 00:00:24 | 1 | cpu-node-38 | 3228K | 500M |
| fastqc-sample=SRR3105698_chr18-41907138.out | COMPLETED | 00:00:18 | 0.2% | 72.2% | 0.6% | 2024-09-24T12:15:35 | 00:00:18 | 1 | cpu-node-34 | 3244K | 500M |
| fastqc-sample=SRR3105699_chr18-41907137.out | COMPLETED | 00:00:16 | 0.2% | 68.8% | 0.6% | 2024-09-24T12:15:35 | 00:00:16 | 1 | cpu-node-34 | 3228K | 500M |
| fastqc-sample=SRR3099586_chr18-41907133.out | COMPLETED | 00:00:15 | 0.2% | 66.7% | 0.6% | 2024-09-24T12:15:35 | 00:00:15 | 1 | cpu-node-39 | 3144K | 500M |
| fastqc-sample=SRR3099585_chr18-41907134.out | COMPLETED | 00:00:15 | 0.2% | 73.3% | 0.6% | 2024-09-24T12:15:35 | 00:00:15 | 1 | cpu-node-40 | 3272K | 500M |
| fastqc-sample=SRR3099587_chr18-41907136.out | COMPLETED | 00:00:14 | 0.2% | 71.4% | 0.6% | 2024-09-24T12:15:35 | 00:00:14 | 1 | cpu-node-40 | 3208K | 500M |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |

# Job arrays

Job arrays offer a mecanism for launching a lot of tasks at the same time. Each task of the job will have the environment variable $SLURM_ARRAY_TASK_ID set to its array index value.

myscript.sbatch

```
#!/bin/bash
#SBATCH --partition=fast
#SBATCH --account=2417_wf4bioinfo
#SBATCH --output=resultat_%a.log
#SBATCH --array=1-3
case "$SLURM_ARRAY_TASK_ID" in
   1) fruit='orange';;
   2) fruit='apple';;
   3) fruit='banana';;
esac
echo $fruit
```

# Job arrays

```
[mhennion @ ifb]$ sbatch myscript.sbatch
```

Results:

```
[mhennion @ ifb]$ ls
resultat_1.log  resultat_2.log  resultat_3.log
[mhennion @ ifb]$ tail resultat_*
==> resultat_1.log <==
orange

==> resultat_2.log <==
apple

==> resultat_3.log <==
banana
```

# Job arrays

Use Bash lists

myscript.sbatch

```
#!/bin/bash
#SBATCH --partition=fast
#SBATCH --account=2417_wf4bioinfo
#SBATCH --output=resultat_%a.log
#SBATCH --array=0-200%50
#SBATCH --cpus-per-task=2
INPUTS=(*.fq.gz)
fastqc ${INPUTS[$SLURM_ARRAY_TASK_ID]}
```

# Job arrays

```
#SBATCH --array=0-15
```

= 16 jobs ($SLURM_ARRAY_TASK_ID: from 0 to 15 included).

```
#SBATCH --array=10-16:2
```

= 4 jobs ($SLURM_ARRAY_TASK_ID: 10,12,14,16).

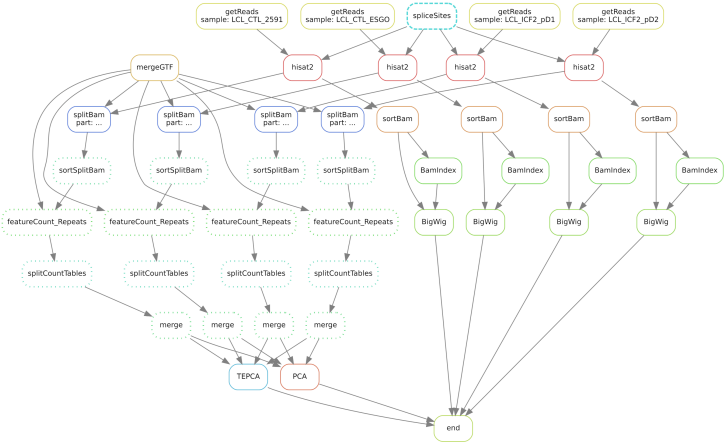
```
#SBATCH --array=2,3-7:2,11,13
```

= 6 jobs ($SLURM_ARRAY_TASK_ID: 2,3,5,7,11,13).

```
#SBATCH --array=1-10000%32
```

= 10 000 jobs, max 32 simultaneous jobs

# Complex workflows




Use workflow managers such as Snakemake or Nextflow.

# Modules

Tools are installed on the cluster in virtual environments

- each tool has its own dependencies (libraries) and it's not possible to make them all coexist in the same environment
- reproducibility: keep several versions of a tool

Conda environments


Containers (Apptainer)

# Modules

They can be loaded with the module command.
Look for the different versions of multiqc:

```
[mhennion @ ifb]$ module avail multiqc
multiqc/1.3  multiqc/1.6  multiqc/1.7  multiqc/1.9
```

Load an environment:

```
[mhennion @ ifb]$ module load multiqc/1.9
```

List loaded environments:

```
[mhennion @ ifb]$ module list
Currently Loaded Modulefiles:
  1) multiqc/1.9    2) blast/2.13.0
```

# Useful resources

To find out more, the SLURM manual : `man sbatch` or
https://slurm.schedmd.com/sbatch.html

Ask for help or signal problems on the cluster : https://community.france-bioinformatique.fr

IFB cluster documentation:
https://ifb-elixirfr.gitlab.io/cluster/doc/

# Thanks





Alix Silvert
Julien Rey
Olivier Kirsh