Training session (UiO/Norway, SAGA cluster, SLURM job queue system)

METAPIPE PIPELINE

METABARCODING STEP BY STEP

SAGA MAIN DIRECTORIES OVERVIEW

- /cluster/**home**/my_user

  You may use this directory to practice bioinformatics, keep mapping files and periodic work diaries. Do not store big files here. Use it as your personal computer “Documents” folder.

- /cluster/**work**/users/my_user

  This is your **actual work directory**. Store and run all work files and scripts here.

  Do not run complex commands or scripts directly on the **command prompt**.

  For running anything on command prompt, you **must request resources** using the following **srun** command:

  srun --nodes=1 --ntasks=1 --mem=8G --time=02:00:00 --qos=devel --account=nn9XXXk --pty bash –I

  edit account=project code, e.g. nn9813k or nn9623k. 2 hours is the time limit, after it the session ends, and you need to request resources again.

  Pay attention to copy your commands on work.txt diaries, because you might lose your commands history.

- /cluster/**projects**/nnXXXXk

  When **more than one student** is working with the **same** sequencing dataset, store the rawdata in a dedicated directory in the projects directory, so the teammates may have access to the same files and share metadata files.

Setting up the work environment

1. Directories' structure and work `best practices`

After logging in to SAGA, type "**pwd**"

```
[my_user@login-1.SAGA ~]$ pwd
/cluster/home/my_user
```

Let's go right to your Saga 'work' directory (**cd**: change directory):

```
[my_user@login-1.SAGA ~]$ cd  /cluster/work/users/my_user
```

Create **(mkdir)** your first work directory:

```
[my_user@login-1.SAGA ~]$ mkdir my_work
[my_user@login-1.SAGA ~]$ cd my_work
[my_user@login-1.SAGA ~]$ pwd
/cluster/work/users/my_user/my_work
```

Inside 'my_work' directory, create the directory were your rawdata will be stored:

```
[my_user@login-1.SAGA ~]$ mkdir my_datasets → upload/download your rawdata, the original sequencing fastq files.
[my_user@login-1.SAGA ~]$ cd my_datasets
[my_user@login-1.SAGA ~]$ pwd
/cluster/work/users/my_user/my_work/my_datasets/
```

Download the sequencing dataset using '**wget**' directly in my_datasets directory.

The user and password can be found in the email you have received.

```
[my_user@login-1.SAGA ~]$ wget --user=lab_user --ask-password --accept "*.tar" --recursive --no-directories --no-parent
https://the/address/you/received/from/the/sequencing/facility/dataset.tar
```

Check the directory's contente by listing the files:

```
[my_user@login-1.SAGA ~]$ ls
dataset.tar
```

Setting up the work environment

Extract the sequencing dataset using '**tar**':

```
[my_user@login-1.SAGA ~]$ tar -xvf dataset.tar
[my_user@login-1.SAGA ~]$ ls
dataset_folder
[my_user@login-1.SAGA ~]$ cd  dataset_folder
[my_user@login-1.SAGA ~]$ ls
Illumina_dataset.html     Illumina_dataset_R1_001.fastq.gz     Illumina_dataset_R2_001.fastq.gz
```

Check where you are:

```
[my_user@login-1.SAGA ~]$ pwd
/cluster/work/users/my_user/my_work/my_datasets/dataset_folder
```

Extract the compressed fastq files, because we'll call these fastq files from other directories, and some tools do not work with compressed files.

```
[my_user@login-1.SAGA ~]$ gunzip Illumina_dataset_R1_001.fastq.gz
[my_user@login-1.SAGA ~]$ gunzip Illumina_dataset_R2_001.fastq.gz
[my_user@login-1.SAGA ~]$ ls
Illumina_dataset.html     Illumina_dataset_R1_001.fastq     Illumina_dataset_R2_001.fastq
```

Go back to 'my_work' directory

```
[my_user@login-1.SAGA ~]$ cd ../..  → each '../' representes go back one directory
[my_user@login-1.SAGA ~]$ pwd
/cluster/work/users/my_user/my_work
[my_user@login-1.SAGA ~]$ ls
my_datasets
```

Create a dedicated directory for running the METAPIPE pipeline.
The following directories' structure is very important to keep the correct pipeline's chain of commands.

```
[my_user@login-1.SAGA ~]$ mkdir METAPIPE
[my_user@login-1.SAGA ~]$ cd METAPIPE
[my_user@login-1.SAGA ~]$ pwd
/cluster/work/users/my_user/my_work/METAPIPE
```

Setting up the work environment

Create the directory for the (optionally) first METAPIPE step, the merge of the paired-end Illumina sequences:

```
[my_user@login-1.SAGA ~]$ mkdir 1_merge
[my_user@login-1.SAGA ~]$ cd 1_merge
[my_user@login-1.SAGA ~]$ pwd
/cluster/work/users/my_user/my_work/METAPIPE/1_merge
```

Create simbolic links using '**ln –s**' to call your rawdata stored in my_datasets directory, avoiding big files duplicates.

Create simbolic links for every big file you need as input in other directories

```
[my_user@login-1.SAGA ~]$ ln -s  /cluster/work/users/my_user/my_work/my_datasets/llumina_sequences_R1_001.fastq .
[my_user@login-1.SAGA ~]$ ln -s  /cluster/work/users/my_user/my_work/my_datasets/llumina_sequences_R1_001.fastq .
[my_user@login-1.SAGA ~]$ ls
llumina_sequences_R1_001.fastq llumina_sequences_R2_001.fastq
```

Check the simbolic links using **ls -lh**:

```
[my_user@login-1.SAGA ~]$ ls –lh
lrwxrwxr-x 1 my_user nn9XXXk   64 Sep 30 15:43 llumina_sequences_R1_001.fastq -> /cluster/work/users/my_user/my_work/my_datasets/llumina_sequences_R1_001.fastq
lrwxrwxr-x 1 my_user nn9XXXk   64 Sep 30 15:43 llumina_sequences_R2_001.fastq -> /cluster/work/users/my_user/my_work/my_datasets/llumina_sequences_R1_001.fastq
permissions  user  project  date  time  files  where the files are actually stored
```

Go back to 'my_work' directory and create the directory for the second METAPIPE step, the demultiplexing:

```
[my_user@login-1.SAGA ~]$ cd ../
[my_user@login-1.SAGA ~]$ pwd
/cluster/work/users/my_user/my_work/ METAPIPE
[my_user@login-1.SAGA ~]$ ls
my_datasets     1_merge
[my_user@login-1.SAGA ~]$ mkdir 2_demulti
[my_user@login-1.SAGA ~]$ cd 2_demulti
[my_user@login-1.SAGA ~]$ pwd
/cluster/work/users/my_user/my_work/METAPIPE/2_demulti
```

Setting up the work environment

If the sequencing dataset is shared between teammates:

Create a dedicated directory and download the sequencing dataset on the **project's directory**:

```
[my_user@login-1.SAGA ~]$  cd /cluster/projects/nnXXXk
[my_user@login-1.SAGA ~]$ pwd
/cluster/projects/nnXXXk
[my_user@login-1.SAGA ~]$ ls
user1     user2     user3     dataset1     dataset2     dataset3
```

Create a dedicated directory to the new sequencing dataset (suggested unambiguous name):

```
[my_user@login-1.SAGA ~]$ mkdir PlantID_Illumina_or_Ion_marker1_marker2_rawdata
[my_user@login-1.SAGA ~]$ cd PlantID_Illumina_or_Ion_marker1_marker2_rawdata
```

Check where you are:

```
[my_user@login-1.SAGA ~]$ pwd
/cluster/projects/nnXXXk/PlantID_Illumina_or_Ion_marker1_marker2_rawdata
```

Download the sequencing dataset and extract the data as shown previously.

In your 'work' METAPIPE/1_merge directory, create the simbolic links calling the files from the project's directory:

```
[my_user@login-1.SAGA ~]$ cd /cluster/work/users/my_user/my_work/METAPIPE/1_merge
[my_user@login-1.SAGA ~]$ pwd
/cluster/work/users/my_user/my_work/METAPIPE/1_merge
[my_user@login-1.SAGA ~]$ ln -s  /cluster/projects/nnXXXk/PlantID_Illumina_or_Ion_marker1_marker2_rawdata/Illumina_sequences_R1_001.fastq .
[my_user@login-1.SAGA ~]$ ln -s  /cluster/projects/nnXXXk/PlantID_Illumina_or_Ion_marker1_marker2_rawdata/Illumina_sequences_R1_001.fastq .
[my_user@login-1.SAGA ~]$ ls -lh
lrwxrwxr-x 1 my_user nn9XXXk   64 Sep 30 15:43 Illumina_sequences_R1_001.fastq -> /cluster/projects/nnXXXk/PlantID_Illumina_or_Ion_marker1_marker2_rawdata/Illumina_sequences_R1_001.fastq
lrwxrwxr-x 1 my_user nn9XXXk   64 Sep 30 15:43 Illumina_sequences_R2_001.fastq -> /cluster/projects/nnXXXk/PlantID_Illumina_or_Ion_marker1_marker2_rawdata/Illumina_sequences_R1_001.fastq
```

running jobs

Linux basics:

- **cd -> change directory**
- **pwd -> print working directory**
- **ls -> list**
- **cp -> copy**
- **mv -> rename or actually move**
- **mkdir -> make directory**
- **head -> print 10 first lines**
- **tail -> print 10 last lines**
- **more -> print the whole file**

Text editor:

- **vi**
- **type 'i' to write**
- **type 'esc :wq!' to close and save**
- **type 'esc :q!' to close without saving it**
- **VIM, nano…**

Slurm basics:

- **launch a job --> sbatch run_my_job.slurm**
- **keep tracking -> squeue -u my_user**
  - **if error, check the .out file**
- **module avail vsearch**
  - **VSEARCH/2.9.1-foss-2018b**
- **module load VSEARCH/2.9.1-foss-2018b**

- MPI (message passing protocol) → distributed memory systems
- applications running on multiple computers (nodes) sharing (intermediate) results
- Partition: allocations of resources, queue
- Most of our tools can shared memory among cores (CPUs), but no truly paralelize, distribution over nodes.
- Think a node as your laptop, two nodes, two laptops pile up
- Think a “--thread” in a command line as 1 core (CPU).
- At most 40 in each node, at most 4G each.

Single node jobs (Python, Perl, R scripts...)

```
#SBATCH --account=MyProject
#SBATCH --job-name=MyJob
#SBATCH --time=72:00:00 → Job will be killed by SLURM after time has run out
#SBATCH –ntasks=8 → Number of processes
#SBATCH --ntasks-per-node=4 → Number of processes per node, max depending on number of CPU's
#SBATCH --cpus-per-task=10 → CPU cores per task, this is the number of threads
#SBATCH –mem-per-cpu=4G → Minimum memory (RAM) per node, e.g. 16G.
```

2 compute nodes, because 8/4 = 2

OpenMP (Open Multi-Processing): share the memory between all processing units (CPU cores) within one node
Start a parallel job for a shared memory system on only one node
to run 8 threads on a single compute node

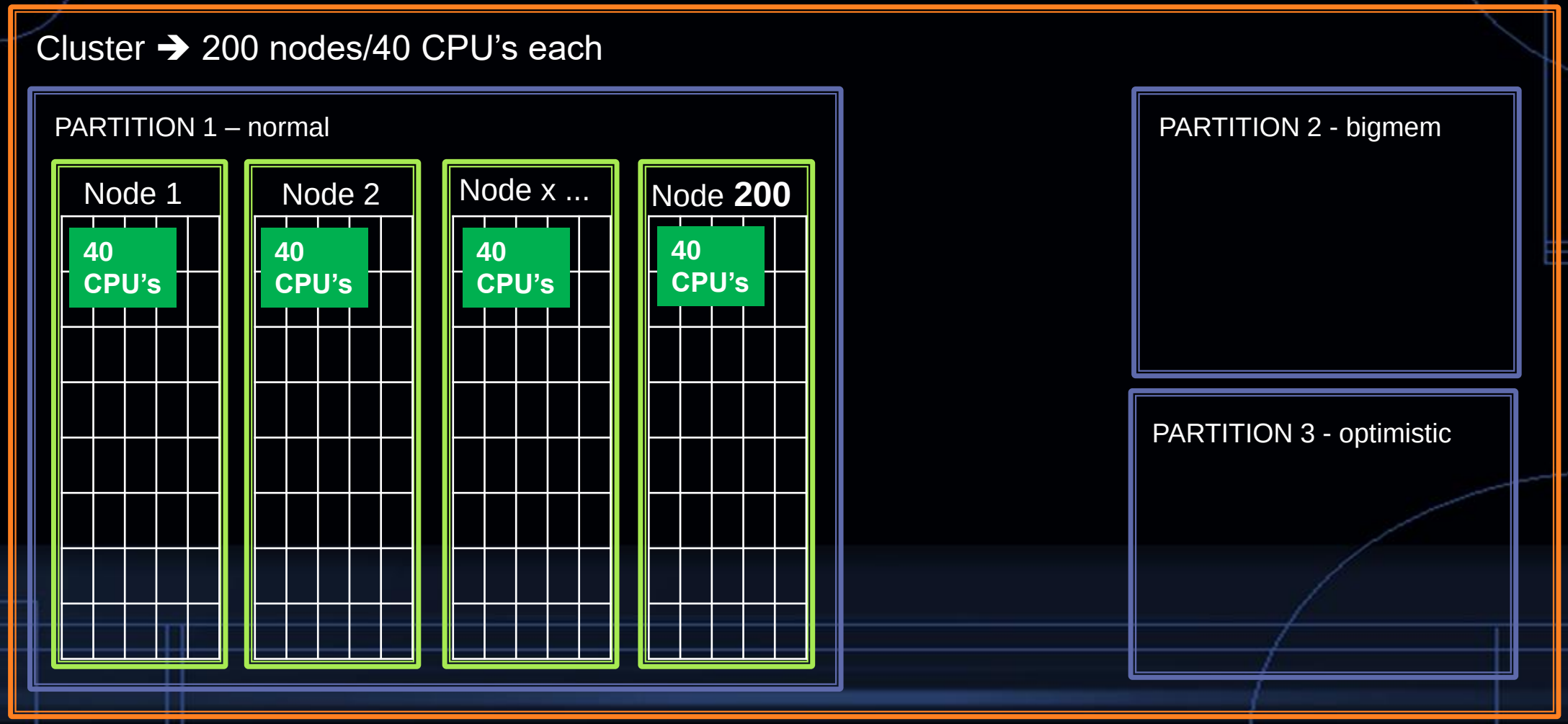
```
#SBATCH --account=MyProject
#SBATCH --job-name=MyJob
#SBATCH --time=72:00:00
#SBATCH –ntasks=8
#SBATCH --cpus-per-task=1
#SBATCH --ntasks-per-node=8
#SBATCH --mem 2G
```

Threads, such as those you can create with PThreads and OpenMPI, allow you to make use of multiple cores on the same compute node.

Single run

```
#SBATCH --account=MyProject
#SBATCH --job-name=MyJob
#SBATCH --time=72:00:00
#SBATCH --mem=4G
```

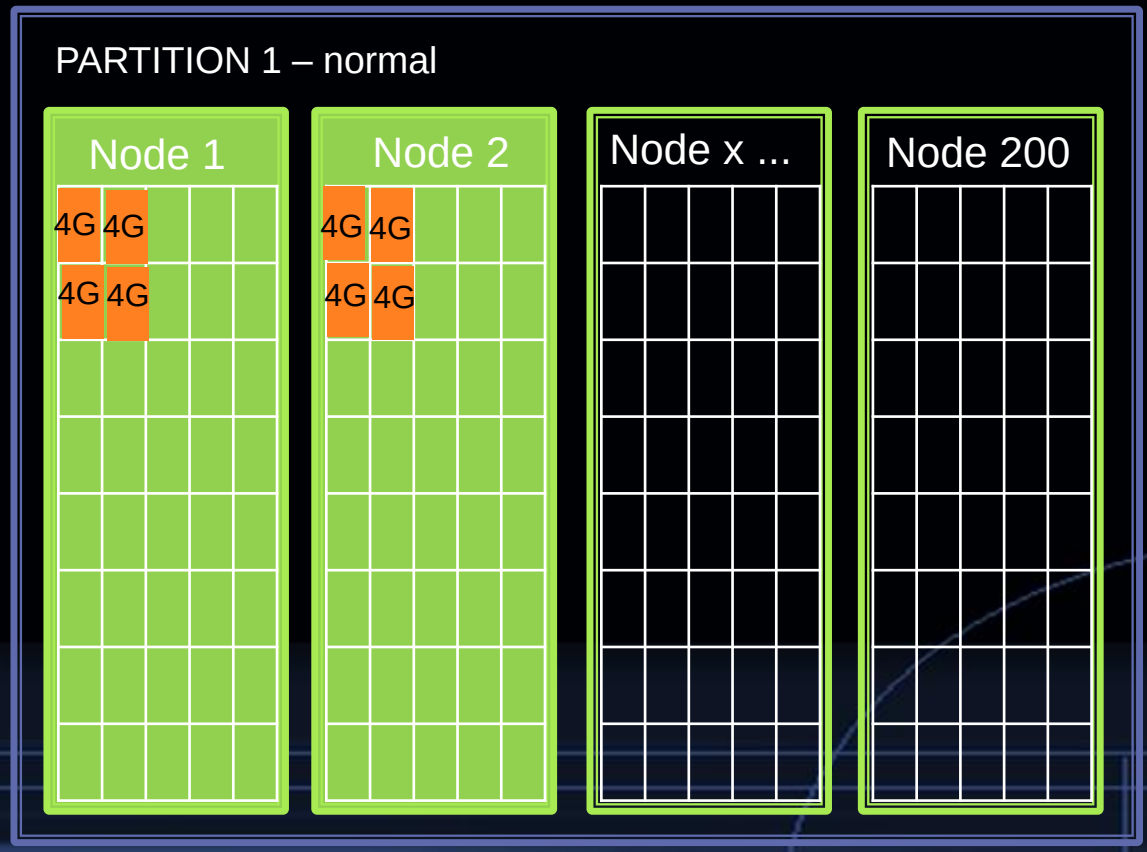
Remember to use geometric sequence with common ratio 2 for RAM and tasks: 1, 2, 4, 8, 16...


Cluster ➔ 200 nodes/40 CPU's each
PARTITION 1 – normal
Node 1
40 CPU's
Node 2
40 CPU's
Node x ...
40 CPU's
Node 200
40 CPU's
PARTITION 2 - bigmem
PARTITION 3 - optimistic

Cluster

```
#!/bin/bash
#SBATCH --account=MyProject
#SBATCH --job-name=MyJob
#SBATCH --time=2-48:0:0
#SBATCH --mem-per-cpu=4G
#SBATCH --ntasks=8
#SBATCH --tasks-per-node=4
SBATCH –cpus-per-task=1
```

This job will get 2 nodes (8/4=2), and run 4 processes on each node, using 1 cpu by task
16G RAM by node



4G 4G

4G 4G

Cluster

```
#!/bin/bash
#SBATCH --account=MyProject
#SBATCH --job-name=MyJob
#SBATCH --time=2-48:0:0
#SBATCH --mem-per-cpu=4G
#SBATCH --ntasks=8
#SBATCH --cpus-per-task=10
#SBATCH --tasks-per-node=4
```

This job will get 2 nodes (8/4=2), and run 4 processes on each of them, **each process getting 10 cpus. All in all, that will be two whole nodes on Saga.**
40G RAM by node