Scientific computing resources

Maxime KERMARQUER

Using the ICM cluster

Introduction

- Goal
  - This training gets you started with the basics of connecting and running your code / applications on the cluster
- Prerequisites
  - This training assumes knowledge about the Linux command line
    - `ls , cd , pwd , tree , cat , tail , less , vim , sed , awk , |`
    - Relative path / Absolute path
      - `data`
      - `/network/lustre/iss01/home/maxime.kermarquer/data`
    - Bash environment variables
      - `export VAR1=value`
      - `echo ${VAR1}`
- Topics
  - What is a cluster
  - Cluster storage
  - Module environment
  - Submitting jobs on the cluster with SLURM

Introduction

Agenda

1 **HPC**

What is it ?

Do I need it ?

2 **IT Infrastructure**

Computing resources ?

Storage spaces ?

3 **Cluster tools**

How do I use the cluster environment ?

4 **SLURM**

How I can compute my data ?

How does it work ?

5 **Deep learning**

Which framework are available ?

How to use it ?

1

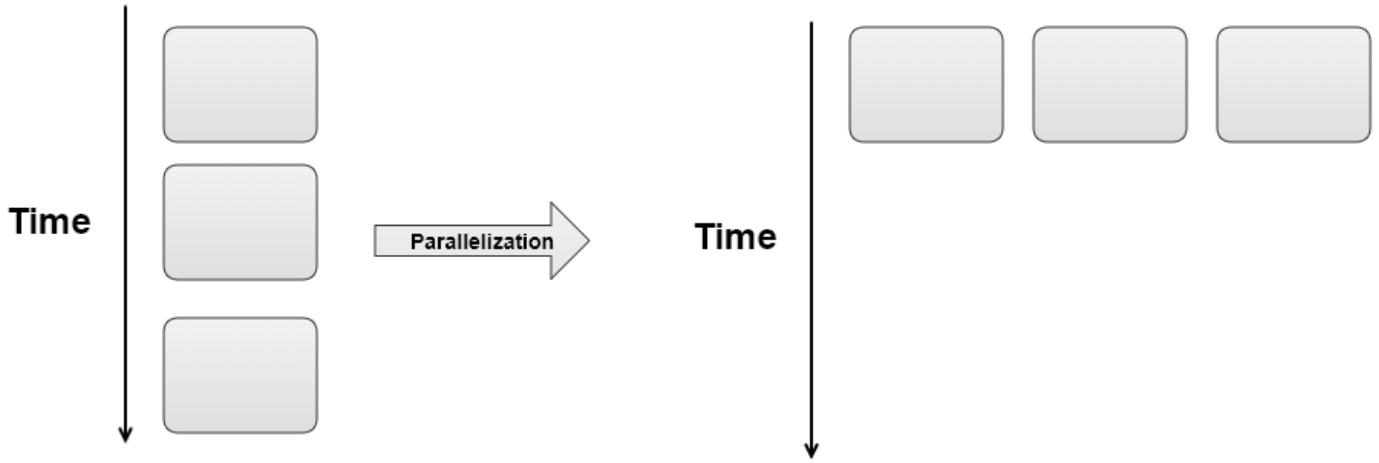
HPC

What is it ? Do I need it ?

High Performance Computing
What is it?

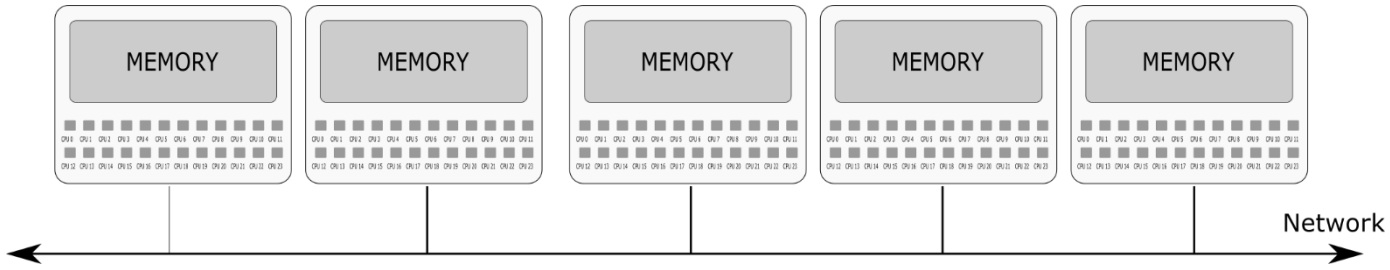
- **Serial Computing**
  - A problem is broken down into instructions
  - Instructions are executed sequentially
  - Executed on a single processor

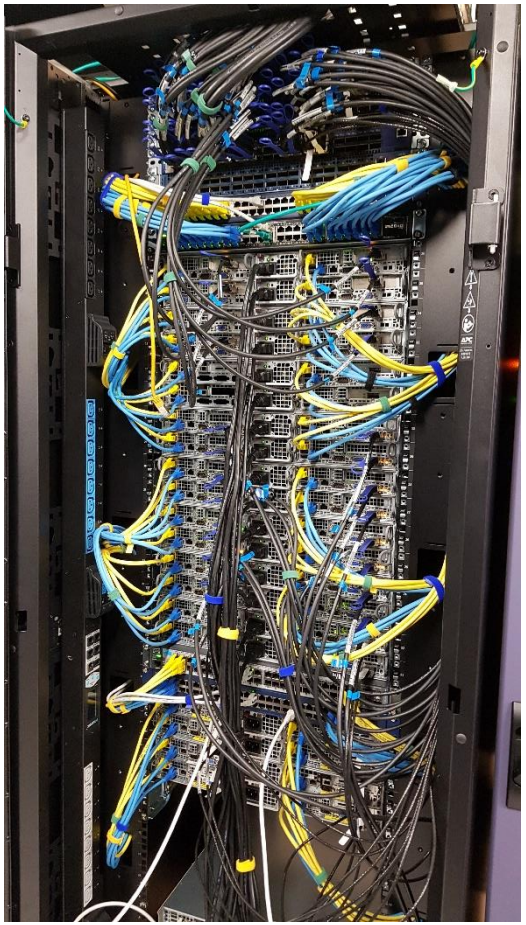
- **Parallel Computing**
  - A problem is broken into parts that can be solved in parallel
  - Each part is broken down into instructions
  - Instructions from each part execute simultaneously
  - Executed on multiple processors



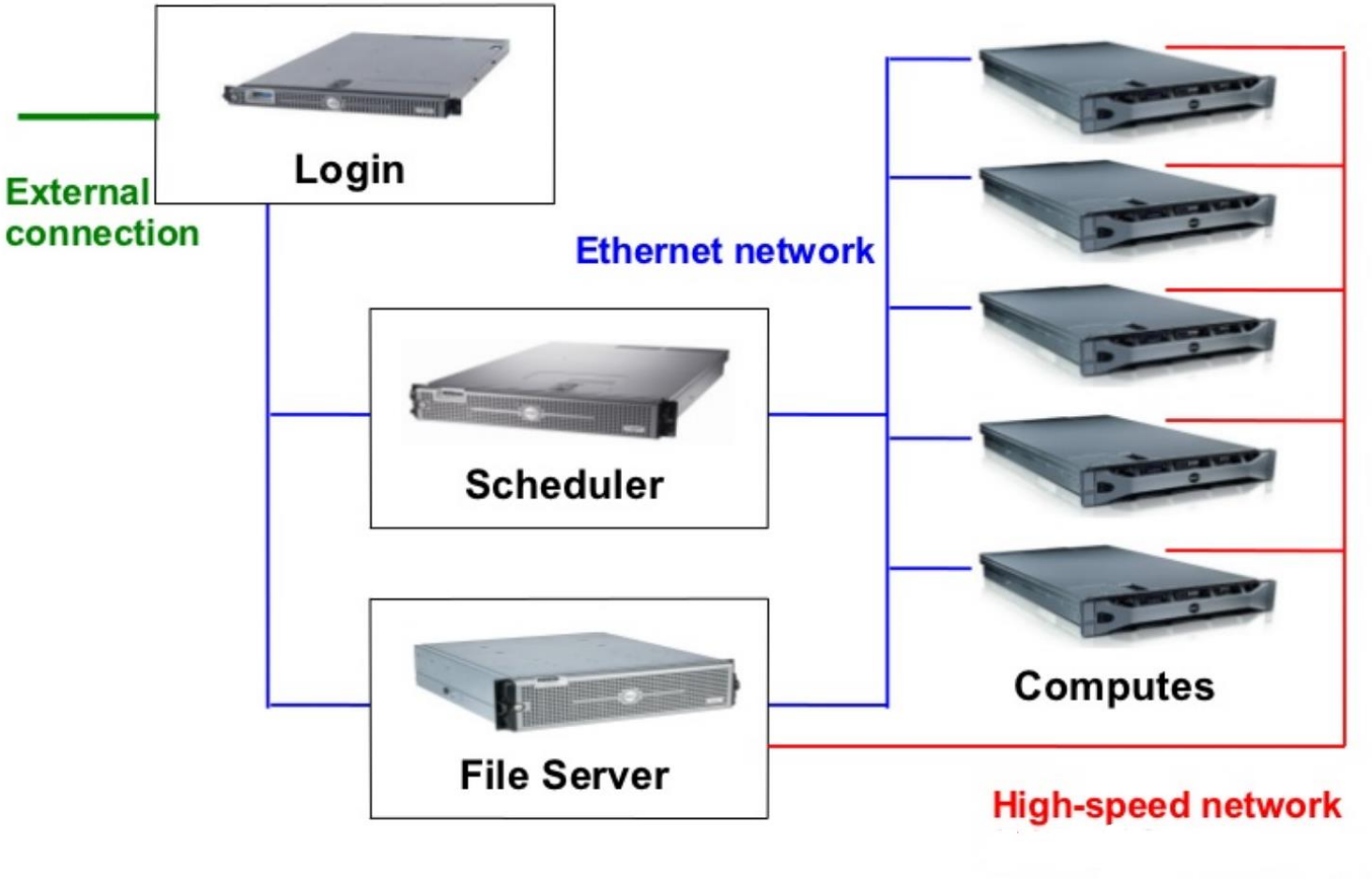
High Performance Computing
What is a cluster ?

- A **collection of computers** (nodes) connected by a network
- Lot of computers in the background for running tasks
- Most clusters run on Linux



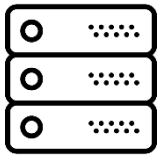


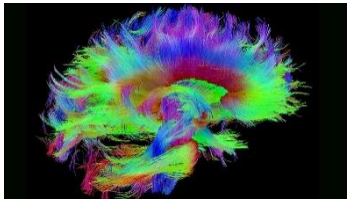
High Performance Computing
What is a cluster ?

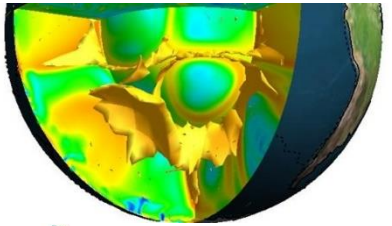


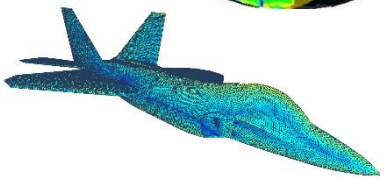
High Performance Computing
What can it do for me ?

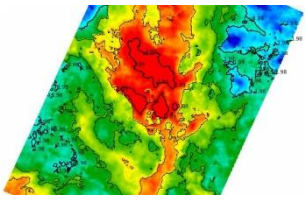
- Why use HPC ?
  - Save time
  - Solve larger problems
  - Use non-local resources

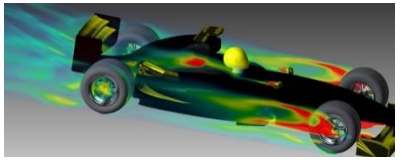


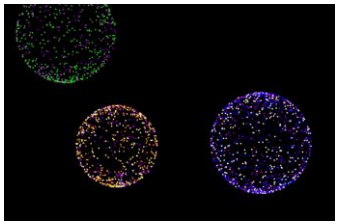












- Who is using HPC ?
  - Universities and government institutions
    - Atmosphere, Earth, Environment
    - Physics – applied, nuclear, particle
    - Mechanical Engineering
    - Chemistry, Molecular Sciences
    - Geology, Seismology
    - Medical imaging and diagnosis
    - Genetics
    - …
  - Industrial and Commercial
    - Big Data, data mining
    - Pharmaceutical design
    - Financial and economic modeling
    - …

High Performance Computing
What do I need for access ?

- A workstation on ICM network or a VPN connection open to computing resources
- Open an issue in category ***Calculateur-07*** to request an account
- SSH protocol used to connect to the cluster

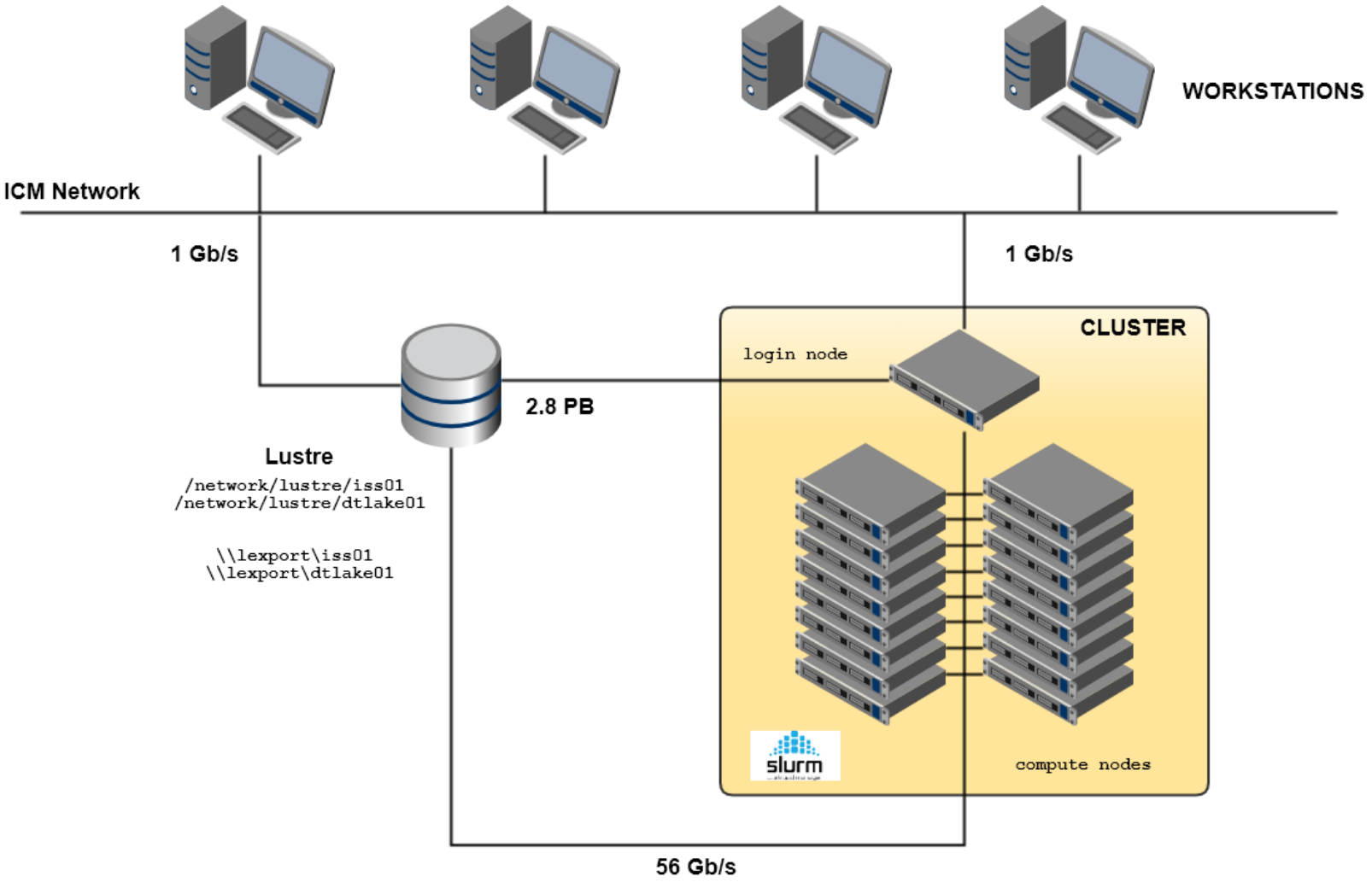
2

IT Infrastructure

What are the storage/computing resources?

IT Infrastructure

Overview



Storage

Structure

Long term archiving - Glacier

Archiving data after > 3 years or make a copy of your data from cold storage
***File retention**: from 3 years to 20 years*

Cold storage

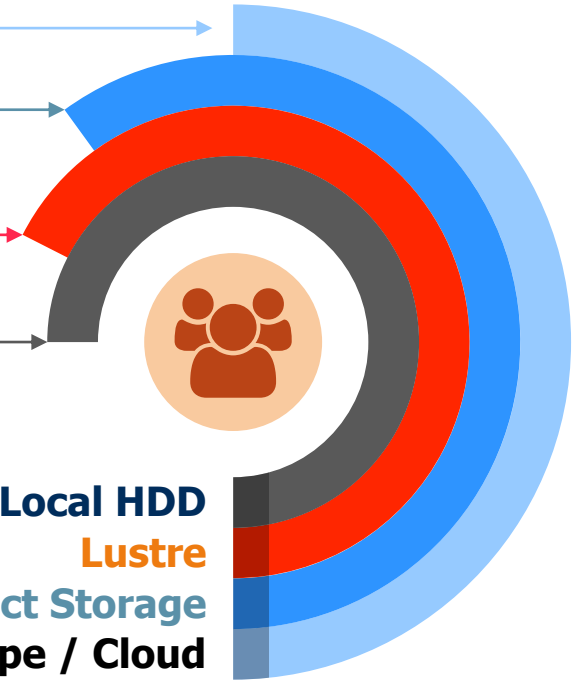
Archive or backup only the data you needs to.
***File retention:** from 1 week to 4 years maximum*

High speed storage

Main storage system, connected to all system, from workstation to servers
***File retention:** now to 2 years*

Local storage

Your workstation hard-drive: no backup, you are responsible of your data



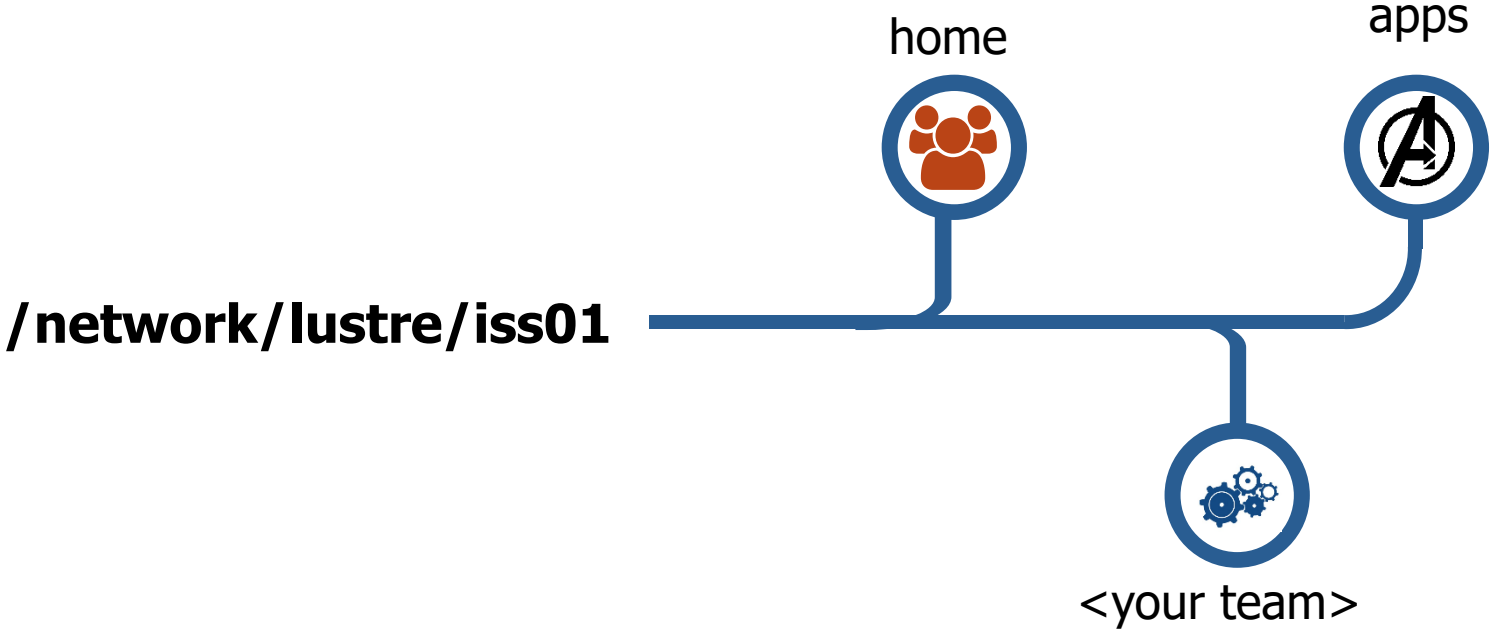
You can access Lustre through Windows (SMB), Linux (SMB/NFS) and Mac OS (NFS/SMB)

Storage

ISS01

Quota: 50Go
No backup
Available on your linux workstation
Available inside the cluster

Available on your linux workstation
Available inside the cluster
Where we store applications



Where your store all files relate to your teams and project

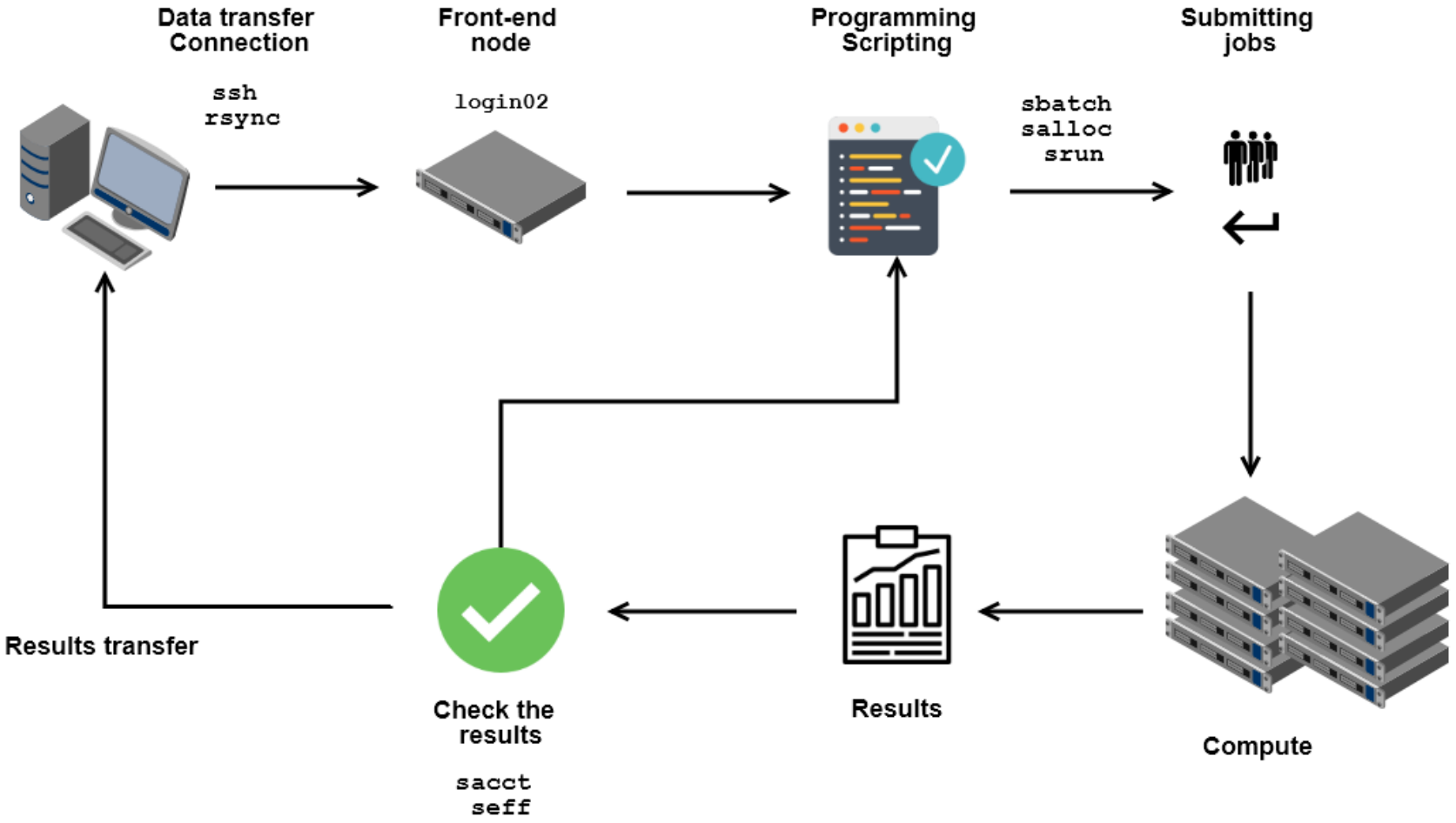
You can access Lustre through Windows (SMB), Linux (SMB/NFS) and Mac OS (NFS/SMB)

3 Cluster tools

How do I use the cluster environment?

Cluster usage

Process



Cluster usage

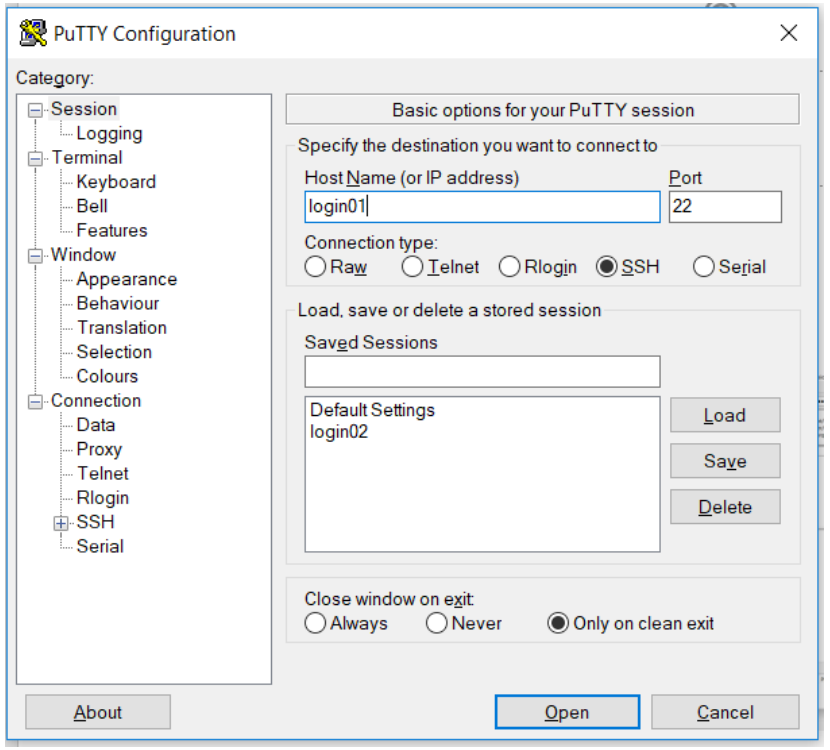
Login node

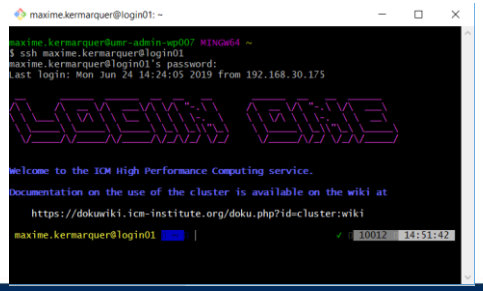
- **Front end node**
  - `login02`
- **Used for access to the cluster**
  - Login
  - Data transfers
  - Job submission
  - Editing scripts
- **Not used to compute**

Cluster usage

SSH connection

- SSH protocol used to connect to the cluster
- IP Address: **192.168.90.2**
- Linux and MacOS
  - From a terminal
  - `ssh <username>@login02`
  - `-X` option to activate Xforwarding
  - `-Y` option on Mac with XQuartz activated
- Windows
  - Putty
  - Git Bash
  - Xming for Xforwarding





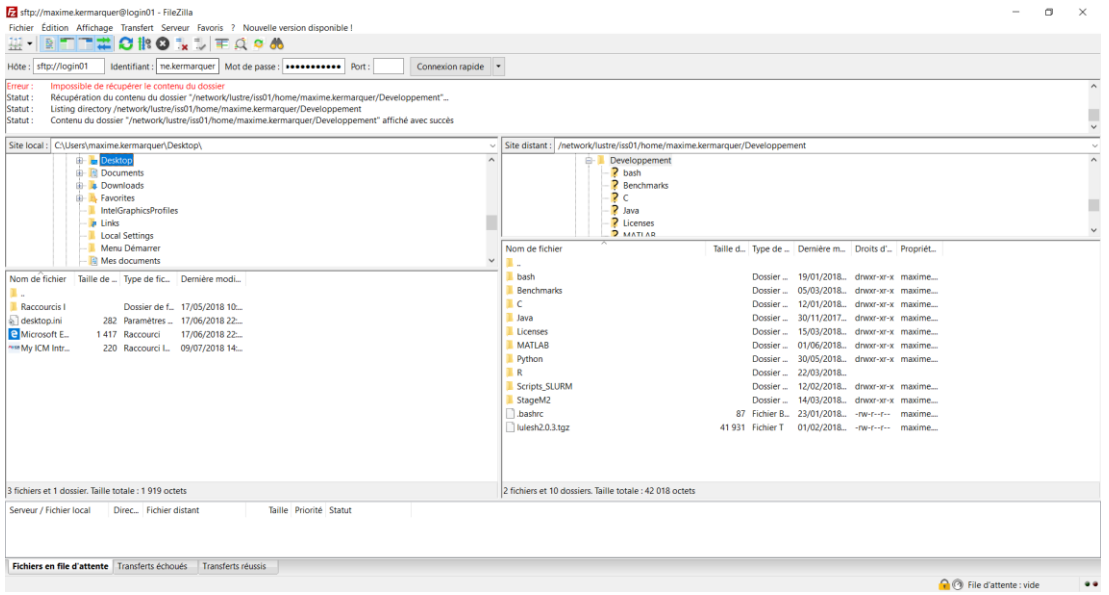
Cluster usage

Data transfer

- **File Transfer client**
  - Linux : `scp`, `rsync` commands with your ICM credentials
    - `rsync <src> <dest>`
    - Options: `-r -h --stats –progress –itemize-change`

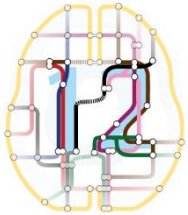
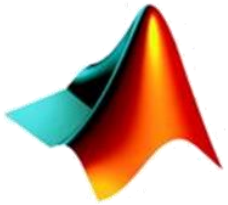
```
rsync -r data maxime.kermarquer@192.168.90.2:/network/lustre/iss01/maxime.kermarquer
```

  - Windows: Filezilla with host as `sftp://192.169.90.2` and your ICM credentials



Software

- MATLAB, R, MRtrix3, FSL, FreeSurfer, ANTs, SPM

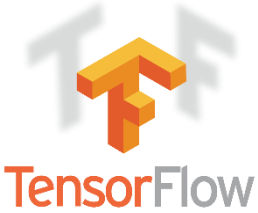













Module
Available software

- Workstation
  - One or few users
  - A small selection of software
- HPC cluster
  - Large number of users
  - Each needing a different selection of software packages
  - Often with different versions and configurations
- Organized by the `module` tool
  - Manage multiple versions and configurations of software
  - Used by many HPC centers around the world
- Environment set for software when module is loaded
- Dependency automatically load

Module Commands

- Display the available modules
  - `module avail`

```
------------------------------------------------------------------------------ Compilers ------------------------------------------------------------------------------
  cmake/3.10.2    gcc/5.5.0    gcc/6.4.0    gcc/7.3.0 (D)    intel/2018    pgi/1710

------------------------------------------------------------------------- Programming Languages -------------------------------------------------------------------------
  MATLAB/R2016a    MATLAB/R2017a    MATLAB/R2017b (L,D)    R/3.4.3    R/3.5.0 (D)    java/7    java/8 (D)    latex/3.14159265    python/bioinfo    python/2.7    python/3.6 (D)

------------------------------------------------------------------------------ Neuroimaging ------------------------------------------------------------------------------
  AFNI/18.1.05    FSL/5.0.1          FreeSurfer/6.0.0 (D)    ICA-AROMA/0.4 (D)    MNE_py27/0.16              PETPVC/1.2.1    TDT/3.991            dcm2niix/v1.0.20171215    fmriprep/1.0.11          phy/spyking_circus_folders
  ANTs/2.2.0      FreeSurfer/5.3.0    ICA-AROMA/0.3           MNE/0.16             MRtrix3/no_gui_19_03_2018  SPM/12          clinica/05_04_2018   field-trip/20180625       phy/spyking_circus_dev   phy/1.0.0                 (D)

------------------------------------------------------------------------------ Tools ------------------------------------------------------------------------------
  C3D/1.1.0    CUDA/8.0    CUDA/9.0    CUDA/9.1 (D)    ICU/58.2    ICU/60.1 (D)    ITK/4.13    fiji/20170530    libwebp/0.5.2    snakemake/4.8.0    svgo/1.0.5    zstd/1.3.3

------------------------------------------------------------------------ /usr/share/lmod/lmod/modulefiles/Core ------------------------------------------------------------------------
  lmod/7.7.18    settarg/7.7.18

Où:
  L:  Le module est chargé
  D:  Default Module
```

- Load / unload a specific module
  - `module load <modulefile>`
  - `module unload <modulefile>`

```
module load MATLAB/R2017b
```

Module Commands

- Display the available modules
  - `module list`

```
Currently Loaded Modules:
  1) CUDA/8.0              5) ITK/4.13                  9) PETPVC/1.2.1
  2) FSL/5.0.1             6) MATLAB/R2017b            10) dcm2niix/v1.0.20171215
  3) FreeSurfer/5.3.0      7) SPM/12                   11) clinica/05_04_2018
  4) ANTs/2.2.0            8) MRtrix3/no_gui_19_03_2018
```

- Display a brief description
  - `module what-is <modulefile>`

```
Version: R2017b
Keywords: MATLAB
URL: https://fr.mathworks.com/products/matlab.html
Description: MATLAB combines a desktop environment tuned for iterative analysis and design processes with a programming language that expresses matrix and array mathematics directly.
```

Module
Installation of new software

- Into your own space storage
  - Keep control over software yourself
  - No special privileges required
  - Cannot be used by others users by default
  - You can request to create module to share with all users

- Into a new module
  - Can be used by other users
  - Installation requires special privileges
  - Open an issue in ***Calculateur-07*** category

4

SLURM

How do I compute my data?

SLURM

Presentation

- SLURM stands for Simple Linux Utility for Resource Management
- Manages and shares hardware resources between users
- The most used scheduler for Linux cluster

- Resources: logical or hardware unit necessary to run software on a cluster, e.g.:
  - Walltime
  - Compute node
  - Memory
  - Accelerator cards
  - …

SLURM

Presentation

1. Users write **job description** and **needed resources**
2. SLURM finds matching resources
3. Scheduler tries to make optimal use of the resources
4. No resource for a job ⟹ wait in a queue
5. Priority determined by usage of the system in recent past and job size

SLURM

Most used commands

- Check the cluster state
  - `sinfo`
  - `cluster_state`

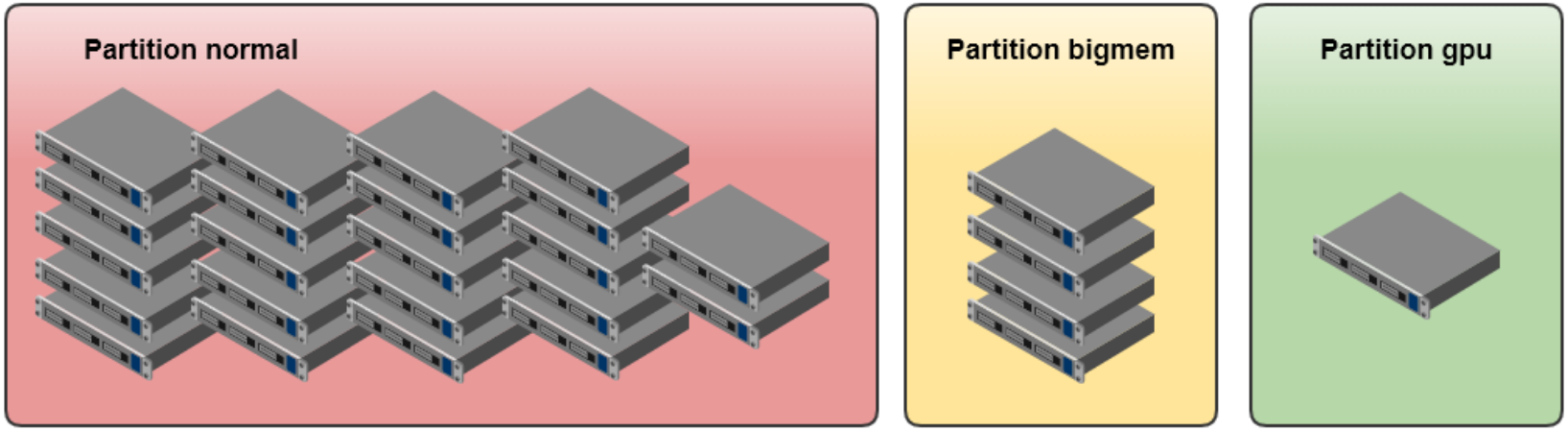
- Running jobs
  - `sbatch` to submit a job script
  - `srun` to run a job (wait the ressources)
  - `salloc` to allocate resources (wait the ressources)

- Monitoring jobs
  - `squeue` to check the jobs in queue
  - `sacct` to check the job history

Slurm

Partitions

- **29 compute nodes form the computing cluster**
  - Based on Intel Xeon E5-2680 v4 processors
  - 22 nodes with 128GB memory and 28 cores
  - 4 nodes with 256GB memory and 28 cores
  - 1 GPU node with 4 NVIDIA Tesla P100 graphics cards with nvlink



Slurm

Partitions

- A partitions/queue is a logical view of all compute node
- The partitions divide the nodes using characteristics:
  - Compute node memory
  - Number of processor
  - Accelerator cards
  - …
- *Partitions are:*

	Nodes	Core/node	Mem/node	Processor
normal	22	28	128G	Broadwell
bigmem	4	28	256G	Broadwell
gpu	1	24	128G	Broadwell

Slurm

sinfo

- `cluster_state` or `sinfo` command display nodes state

```
maxime.kermarquer@login01 ~ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
debug        up   infinite      4   idle icmw[001-004]
bigmem       up   infinite      4    mix icmb[001-004]
normal*      up   infinite      1  drain* icmb020
normal*      up   infinite      3  drain icmb[021,025,030]
normal*      up   infinite     13    mix icmb[022-024,026,028,035-036,039,041-045]
normal*      up   infinite      8  alloc icmb[027,033-034,037-038,040,046-047]
gpu          up   infinite      1   idle icmgpu01
```

- State are the following:
  - `down` : unavailable node for use
  - `mix` : node has some of its CPUs ALLOCATED while others are IDLE
  - `alloc` : node has been allocated to one or more jobs
  - `idle` : node free
  - `drain` : the node is in maintenance

SLURM

Most used commands

- Check the cluster state
  - `sinfo`
  - `cluster_state`

- Running jobs
  - `sbatch` to submit a job script
  - `srun` to run a job (wait the ressources)
  - `salloc` to allocate resources (wait the ressources)

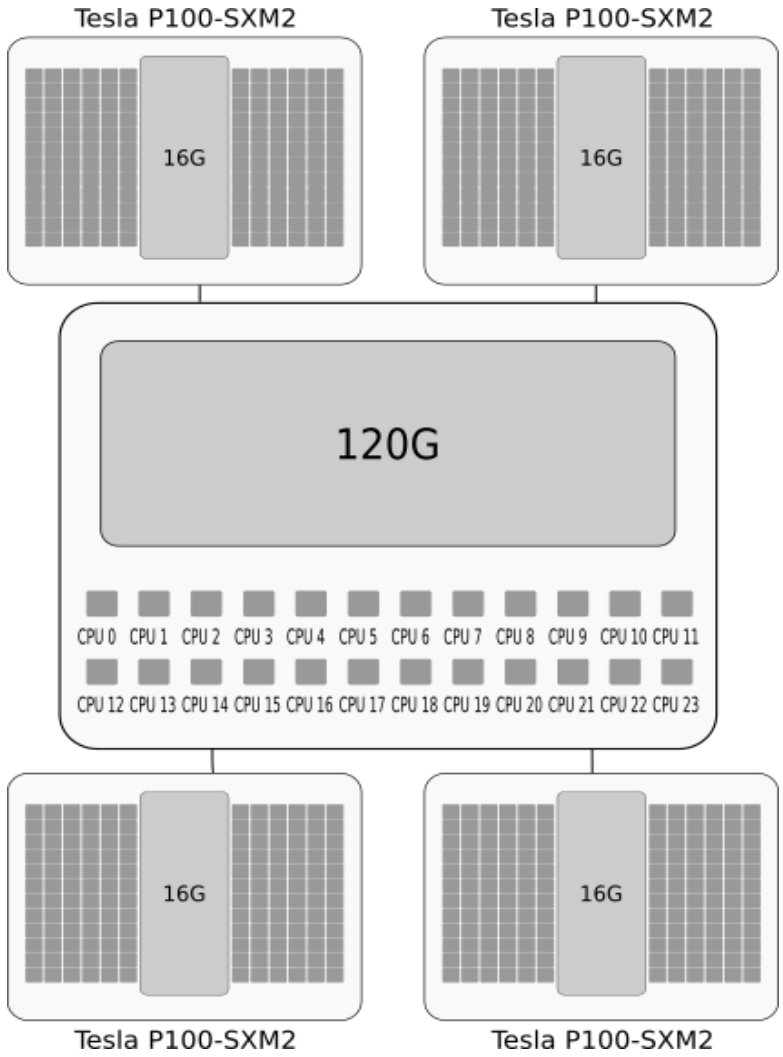
- Monitoring jobs
  - `squeue` to check the jobs in queue
  - `sacct` to check the job history

Running jobs

Commands to submit jobs

- 3 ways allow to submit jobs with SLURM
  - `sbatch`
  - `srun`
  - `salloc + srun / ssh`
- Each job is identified by a **job id** useful to follow its progress
- You need to specify a set of resources in which your job will run
- Exceeding the limit resources will, stop your application/job

Running jobs
GPU server example



Running jobs

Resources options

- Type of compute node used **normal**, **bigmem**, **gpu**

`--partition=<partition_name>` or `-p <partition_name>`

- Time limit of a job with this format for time **HH:MM:SS** or **days-hours:minutes:seconds**

`--time=<time>` or `-t <time>`

- The memory size usable by the job with this format **1024** , **1024MB** or **1G**

`--mem=<count>`

- The number of CPUs per process

`--cpus-per-task=<count>` or `-c <number>`

- The number of GPUs (only for gpu partition)

`--gres=gpu:<count>`

Running jobs

Resources options

- The job name (the default command name)

```
--job-name=<jobname> or -J <jobname>
```

- To choose the working directory

```
--chdir=<path_workdir>
```

- To redirect the error output

```
--error=<filename pattern> or -e <filename pattern>
```

- To redirect the standard output

```
--output=<filename pattern> or -o <filename pattern>
```

Running jobs

sbatch

- A job script is shell file, starts with `#!/bin/bash` followed by `#SBATCH` and SLURM resources options

- Sample script:

```
#!/bin/bash
#SBATCH --mem=2G
#SBATCH --nodes=1
#SBATCH --partition=normal
#SBATCH --time=01:00:00


hostname
```

SLURM resources options

Your commands

- To submit a job with `sbatch` command
  - `sbatch <script_name>.sh`

Running jobs

MATLAB example

```
#!/bin/bash
#SBATCH --job-name=run_MATLAB
#SBATCH --partition=normal
#SBATCH --time=03:00:00
#SBATCH --mem=10G
#SBATCH --cpus-per-task=1
#SBATCH --chdir=/network/lustre/iss01/home/maxime.kermarquer
#SBATCH --output=file_output_%j.log
#SBATCH --error=file_output_%j.log

module load MATLAB
matlab -nojvm -nodesktop -nodisplay < mycode.m
```

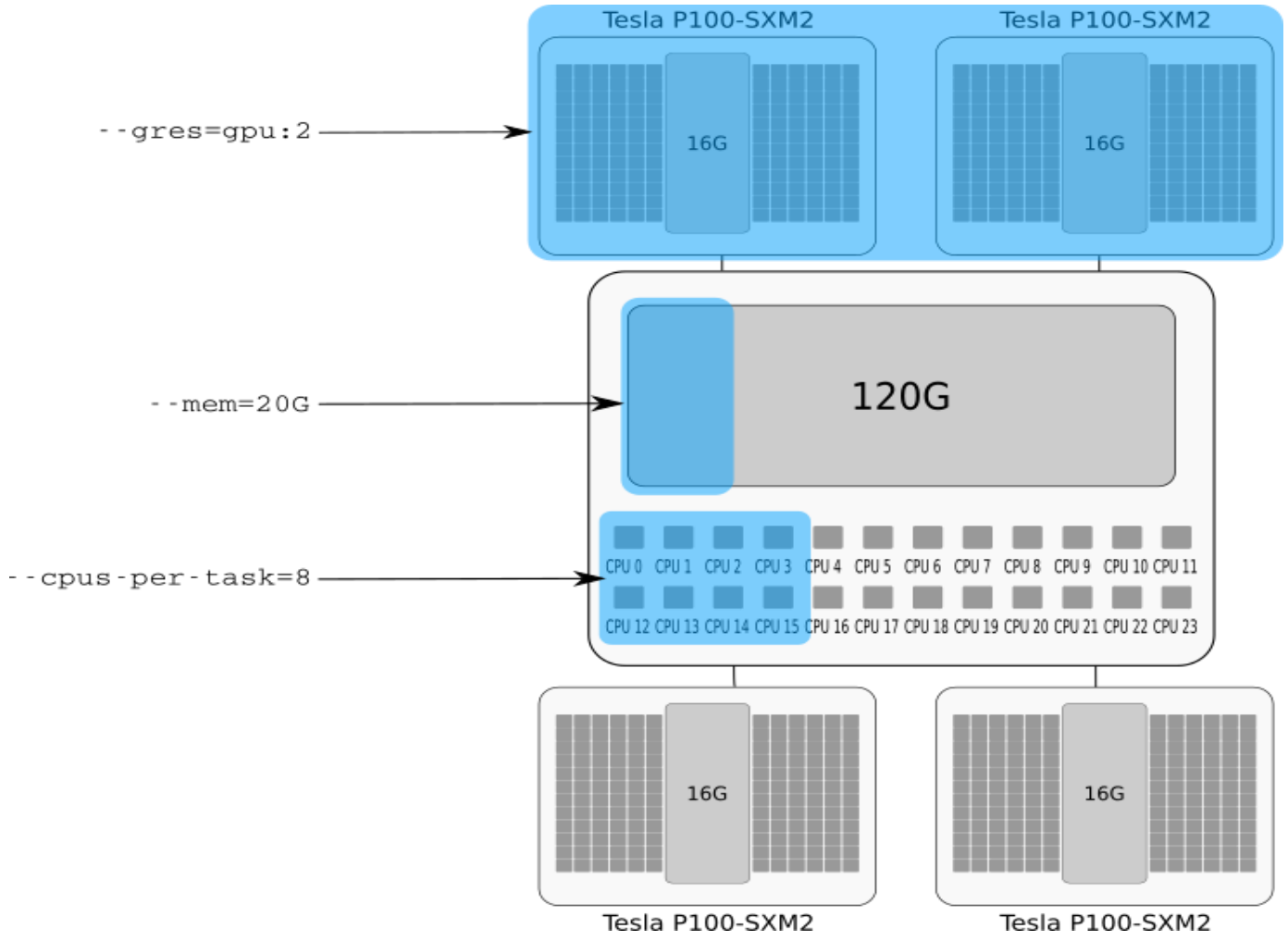
You can find other examples from `gitlab` at
https://gitlab.com/maxime.kermarquer/examples-slurm-scripts

Running jobs
GPU allocation example

```
#!/bin/bash
#SBATCH --partition=gpu
#SBATCH --time=02:00:00
#SBATCH --mem=20G
#SBATCH --gres=gpu:2
#SBATCH --cpus-per-task=8
#SBATCH --workdir=.
#SBATCH --output=tensorflow_gpu_job_%j.log
#SBATCH --error=tensorflow_gpu_job_%j.log
#SBATCH --job-name=test_tensorflow

module load python/2.7
module load CUDA/9.0
python tensorflow_example.py
```

Running jobs
GPU allocation example



Running jobs

Slurm environment variables

- SLURM defines some environment variable you can use to tune your script:

Variable	Function	Example
SLURM_JOBID	Job number	4242
SLURM_SUBMIT_DIR	Workdir	/network/iss01/home/maxime.kermarquer
SLURM_JOB_NAME	Job name	run_MATLAB
SLURM_NODELIST	Node list	lmb021
SLURM_CPUS_PER_TASK	CPUs by tasks	2
SLURM_MEM_PER_NODE	Memory	4096M

Running jobs

Slurm environment variables – Usage example

```
#!/bin/bash
#SBATCH --partition=normal
#SBATCH --cpus-per-task=8
#SBATCH --mem=32G
#SBATCH --time=20:00:00
#SBATCH --chdir=.
#SBATCH --output=outputs-slurm/fmriprep-20.0.0-test-%j.txt
#SBATCH --error=outputs-slurm/fmriprep-20.0.0-test-%j.txt
#SBATCH --job-name=fmriprep-20.0.0-test

module load fmriprep

export BIDS_DIR=../bids-example/ds001_BIDS
export OUTPUT_DIR=outputs
export PARTICIPANT=participant
export WORK_DIRECTORY=workdir
export OPTIONS="--verbose --work-dir ${WORK_DIRECTORY}  --nthreads ${SLURM_CPUS_PER_TASK}
--mem_mb ${SLURM_MEM_PER_NODE}"


fmriprep ${BIDS_DIR} ${OUTPUT_DIR} ${PARTICIPANT} --work-dir ${WORK_DIRECTORY} ${OPTIONS}
```

Running jobs

srun

- You can directly run a command on the cluster without write a script with `srun` command.
  - `srun [options to define needed resource] [your command]`
- By example

```
srun -p gpu -t 02:00:00 --mem=20G --gres=gpu:1 -c 1
-J "run_tensorflow"    python tensorflow_example.py
```

- This command blocks your terminal until your command ends or is cancelled.
- The workdir is the current directory.

Running jobs

salloc + srun

- If you use `srun` command, the resources allocated for the job will be released at his end.

- Using `salloc` command allows to keep the allocation at the end of the command and to start an other command without waiting for the allocation of resources.

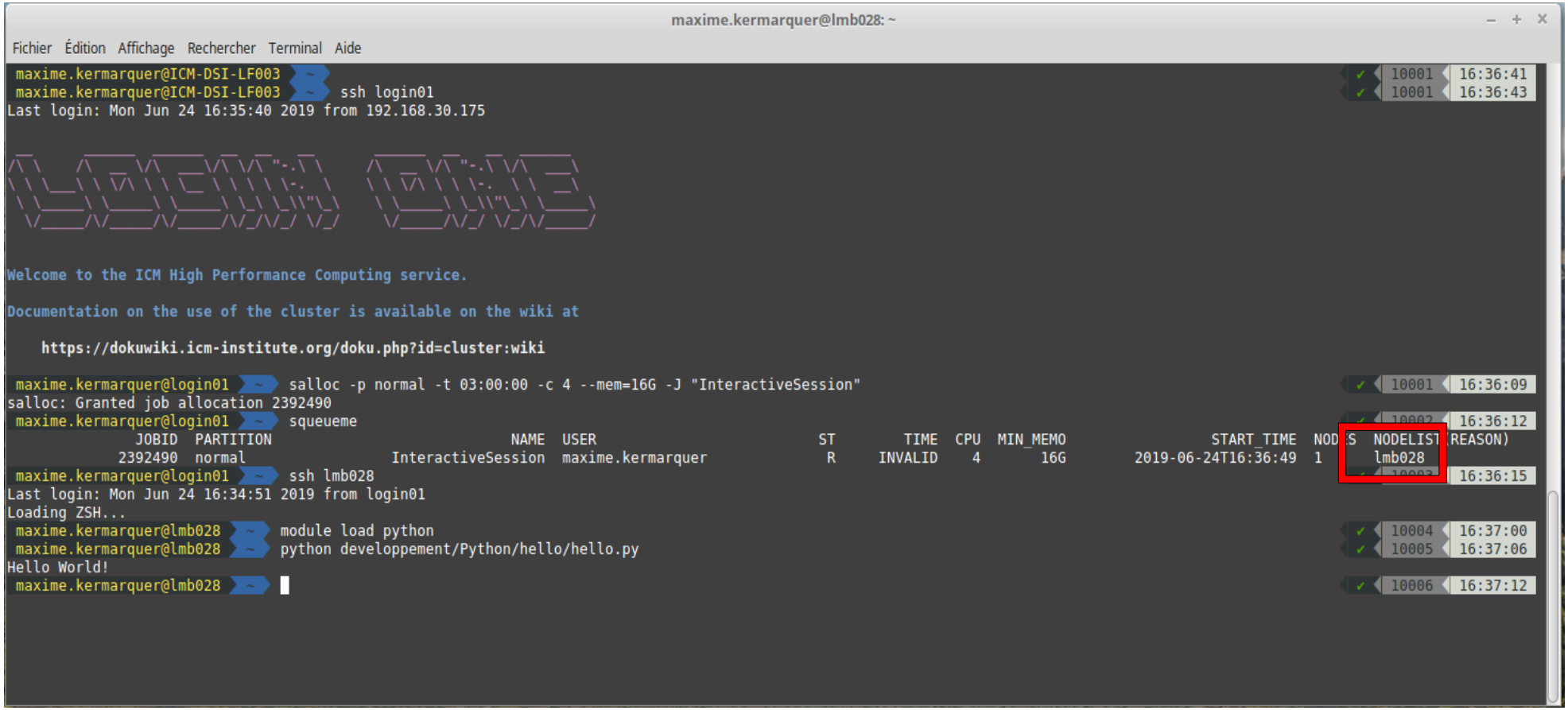
- `exit` command release the allocation

- By example:

```
salloc -p gpu -t 02:00:00 --mem=20G --gres=gpu:1 -N 1 -n 1 -c 1
srun python script1.py
srun python script2.py
exit
```

Running jobs

salloc + ssh



SLURM

Most used commands

- Check the cluster state
  - `sinfo`
  - `cluster_state`

- Running jobs
  - `sbatch` to submit a job script
  - `srun` to run a job (wait the ressources)
  - `salloc` to allocate resources (wait the ressources)

- Monitoring jobs
  - `squeue` to check the jobs in queue
  - `sacct` to check the job history

Monitoring jobs

squeue

- The `squeue` command displays the status of your jobs. It gives information on :
  - The job ID
  - The job partition
  - The name of the job
  - The user
  - The state
  - The running time
  - …

- Can be filtered on:

```
squeue -u $USER      # To display his own jobs, for new users you can use the alias "squeueme"
squeue -t R            # To display the running jobs
squeue -t PD           # To display the pending jobs
squeue -u $USER -t R   # To display his own running jobs
```

Monitoring jobs

squeue

JOBID	PARTITION	NAME	USER	ST	TIME	CPU	MIN_MEMO	SUBMIT_TIME	NODES	NODELIST(REASON)
1278770	gpu	preprocessing	elina.thibeausutre	PD	0:00	1	10G	2018-11-27T16:59:24	1	(Resources)
1282652	normal	snakejob.bam2bw.2.sh	justine.guegan	R	4:29:54	6	80000M	2018-11-28T09:53:44	1	lmb027
1282915	normal	snakejob.mirdeep2.5.sh	thomas.gareau	R	46:47	1	8G	2018-11-28T13:36:52	1	lmb026
1282913	normal	snakejob.mirdeep2.18.sh	thomas.gareau	R	47:23	1	8G	2018-11-28T13:36:16	1	lmb026
1282923	normal	snakejob.mirdeep2.17.sh	thomas.gareau	R	38:34	1	8G	2018-11-28T13:45:04	1	lmb027
1282921	normal	snakejob.mirdeep2.14.sh	thomas.gareau	R	39:10	1	8G	2018-11-28T13:44:28	1	lmb027
1282941	normal	snakejob.mirdeep2.25.sh	thomas.gareau	R	30:08	1	8G	2018-11-28T13:53:30	1	lmb047
1282933	normal	snakejob.mirdeep2.6.sh	thomas.gareau	R	32:45	1	8G	2018-11-28T13:50:54	1	lmb030
1282946	normal	snakejob.mirdeep2_inputs.32.sh	thomas.gareau	R	26:55	1	20G	2018-11-28T13:56:44	1	lmb047
1282951	normal	snakejob.mirdeep2.11.sh	thomas.gareau	R	20:31	1	8G	2018-11-28T14:03:08	1	lmb026
1282950	normal	snakejob.mirdeep2.3.sh	thomas.gareau	R	20:44	1	8G	2018-11-28T14:02:55	1	lmb026
1282959	normal	snakejob.mirdeep2_inputs.33.sh	thomas.gareau	R	15:20	1	20G	2018-11-28T14:08:19	1	lmb030
1282957	normal	snakejob.mirdeep2.23.sh	thomas.gareau	R	16:43	1	8G	2018-11-28T14:06:56	1	lmb026
1282955	normal	snakejob.mirdeep2.24.sh	thomas.gareau	R	17:43	1	8G	2018-11-28T14:05:56	1	lmb026
1282954	normal	snakejob.mirdeep2.13.sh	thomas.gareau	R	17:56	1	8G	2018-11-28T14:05:43	1	lmb026
1282967	normal	snakejob.mirdeep2_inputs.41.sh	thomas.gareau	R	10:31	1	20G	2018-11-28T14:13:07	1	lmb047
1282966	normal	snakejob.mirdeep2.22.sh	thomas.gareau	R	11:06	1	8G	2018-11-28T14:12:32	1	lmb030
1282964	normal	snakejob.mirdeep2.15.sh	thomas.gareau	R	12:06	1	8G	2018-11-28T14:11:32	1	lmb047
1282963	normal	snakejob.mirdeep2_inputs.34.sh	thomas.gareau	R	13:31	1	20G	2018-11-28T14:10:07	1	lmb028
1282961	normal	snakejob.mirdeep2_inputs.37.sh	thomas.gareau	R	14:20	1	20G	2018-11-28T14:09:19	1	lmb030
1282971	normal	snakejob.mirdeep2.1.sh	thomas.gareau	R	5:17	1	8G	2018-11-28T14:18:22	1	lmb047
1282970	normal	snakejob.mirdeep2.2.sh	thomas.gareau	R	6:06	1	8G	2018-11-28T14:17:33	1	lmb047
1282969	normal	snakejob.mirdeep2.4.sh	thomas.gareau	R	9:30	1	8G	2018-11-28T14:14:08	1	lmb047
1282975	normal	snakejob.mirdeep2.10.sh	thomas.gareau	R	1:17	1	8G	2018-11-28T14:22:21	1	lmb047
1282973	normal	snakejob.mirdeep2.20.sh	thomas.gareau	R	1:28	1	8G	2018-11-28T14:22:10	1	lmb047
1282974	normal	snakejob.picard_rnaseq_metrics	thomas.gareau	R	1:28	1	4G	2018-11-28T14:22:10	1	lmb047
1282972	normal	snakejob.mirdeep2.21.sh	thomas.gareau	R	4:53	1	8G	2018-11-28T14:18:45	1	lmb047
1277638	gpu	deformetrica benchmark	benoit.martin	R	23:03:01	24	120G	2018-11-27T15:20:38	1	lmgpu01
1282649	normal	InteractiveSession	marieconstance.corsi	R	5:36:49	28	120G	2018-11-28T08:46:50	1	lmb046

Monitoring jobs

scontrol

- To display all job information
  - `scontrol show job <job id>`

```
JobId=1284207 JobName=hello_world_matlab
   UserId=maxime.kermarquer(20108) GroupId=utilisa. du domaine(10513) MCS_label=N/A
   Priority=7994 Nice=0 Account=dsi QOS=qos1
   JobState=RUNNING Reason=None Dependency=(null)
   Requeue=1 Restarts=0 BatchFlag=1 Reboot=0 ExitCode=0:0
   RunTime=00:00:04 TimeLimit=01:00:00 TimeMin=N/A
   SubmitTime=2018-11-28T18:48:57 EligibleTime=2018-11-28T18:48:57
   StartTime=2018-11-28T18:48:58 EndTime=2018-11-28T19:48:58 Deadline=N/A
   PreemptTime=None SuspendTime=None SecsPreSuspend=0
   LastSchedEval=2018-11-28T18:48:58
   Partition=normal AllocNode:Sid=login01:18177
   ReqNodeList=(null) ExcNodeList=(null)
   NodeList=lmb030
   BatchHost=lmb030
   NumNodes=1 NumCPUs=1 NumTasks=1 CPUs/Task=1 ReqB:S:C:T=0:0:*:*
   TRES=cpu=1,mem=10G,node=1,billing=1
   Socks/Node=* NtasksPerN:B:S:C=0:0:*:* CoreSpec=*
   MinCPUsNode=1 MinMemoryNode=10G MinTmpDiskNode=0
   Features=(null) DelayBoot=00:00:00
   Gres=(null) Reservation=(null)
   OverSubscribe=OK Contiguous=0 Licenses=(null) Network=(null)
   Command=/network/lustre/iss01/home/maxime.kermarquer/hello_world/job.sh
   WorkDir=/network/lustre/iss01/home/maxime.kermarquer/hello_world/.
   StdErr=/network/lustre/iss01/home/maxime.kermarquer/hello_world/./file_output_1284207.err
   StdIn=/dev/null
   StdOut=/network/lustre/iss01/home/maxime.kermarquer/hello_world/./file_output_1284207.out
   Power=
```

Monitoring jobs

scancel

- To cancel jobs in queue or running jobs, you can use `scancel` command.

- Can be filtered on:

```
scancel  13082016                  # To delete the job 13082016
scancel -u $USER                   # To delete all your jobs
scancel --name=JobName             # To delete job with the name JobName
scancel --state=PENDING            # To delete all jobs in the queue
scancel --state=RUNNING            # To delete all RUNNING jobs
scancel --nodelist=lmb024,lmb025   # To delete all your jobs running on nodes lmb024 and lmb025
```

Monitoring jobs

sacct

- `sacct` command give information for past and running job

Submit	Partition	User	JobID	JobNa	Timelimit	Elapsed	ReqMem	MaxRSS	AveRSS	State
2018-11-27T16:48:22	normal	gizem.temiz	1278336_54	bedp+	1-00:00:00	08:27:14	1Gn			COMPLETED
2018-11-28T02:35:33			1278336_54.batch	batch		08:27:14	1Gn	0.17G	0.17G	COMPLETED
2018-11-27T16:48:22	normal	gizem.temiz	1278336_56	bedp+	1-00:00:00	08:27:24	1Gn			COMPLETED
2018-11-28T02:40:04			1278336_56.batch	batch		08:27:24	1Gn	0.17G	0.17G	COMPLETED
2018-11-27T16:48:22	normal	gizem.temiz	1278336_57	bedp+	1-00:00:00	08:29:17	1Gn			COMPLETED
2018-11-28T02:40:16			1278336_57.batch	batch		08:29:17	1Gn	0.17G	0.17G	COMPLETED
2018-11-27T16:48:22	normal	gizem.temiz	1278336_58	bedp+	1-00:00:00	08:28:44	1Gn			COMPLETED
2018-11-28T02:40:28			1278336_58.batch	batch		08:28:44	1Gn	0.17G	0.17G	COMPLETED

- Can be filtered on:
  - Time (beginning and end)
  - Partition / accounting
  - Job ID
- You can display various information:
  - Number of processor / node / tasks
  - Memory size / Wall-time
  - …

Jobs array

Options

- Job arrays offer a mechanism for submitting and managing collections of similar jobs quickly and easily.

- All jobs must have the same initial options (e.g. size, time limit, …)

- Option: `--array`:
  - `sbatch --array=0-50`
  - `sbatch --array=1,3,5,7`
  - `sbatch --array=1-50%2`
  - `sbatch --array=1-50:10`

- Environment variable that can be use in our script:
  - `SLURM_ARRAY_JOB_ID`
  - `SLURM_ARRAY_TASK_ID`

Jobs array

Example

```
maxime.kermarquer@login01 ~/job_array $ tree
.
├── outputs
├── run_array.sh
└── scripts
    ├── script_1.py
    ├── script_2.py
    └── script_3.py

2 directories, 4 files
maxime.kermarquer@login01 ~/job_array $ sbatch --array=1-3 run_array.sh
Submitted batch job 1284209
maxime.kermarquer@login01 ~/job_array $ squeueme
JOBID       PARTITION             NAME     ST       TIME  CPU  MIN_MEMO  NODES  NODELIST(REASON)
1284218_1   bigmem          run_array.sh      R       0:04    1        2G  1      hmb001
1284218_2   bigmem          run_array.sh      R       0:04    1        2G  1      hmb001
1284218_3   bigmem          run_array.sh      R       0:04    1        2G  1      hmb001
```

Jobs array

Example

```
#!/bin/bash
#SBATCH --partition=bigmem
#SBATCH --time=02:00:00
#SBATCH --mem=2G
#SBATCH --ntasks=1
#SBATCH --error=outputs/job_%N_%A_%a.log
#SBATCH --output=outputs/job_%N_%A_%a.log

echo "SLURM_JOBID: " $SLURM_JOBID
echo "SLURM_ARRAY_JOB_ID: " $SLURM_ARRAY_JOB_ID
echo "SLURM_ARRAY_TASK_ID: " $SLURM_ARRAY_TASK_ID

module load python/2.7
python scripts/script_${SLURM_ARRAY_TASK_ID}.py
```

- Coordinate: `SLURM_ARRAY_JOB_ID + SLURM_ARRAY_TASK_ID`
- Cancel a job array, e.g.:
  - `scancel 1284218`
- Cancel only some jobs, e.g.:
  - `scancel 1284218_1`
  - `scancel 42_[1-2]`

Jobs array

Example

- Arborescence

```
├── matlab-scripts
│   └── test_array.m
├── outputs
├── parameters
│   └── parameters-path.txt
├── patients
│   ├── 01
│   ├── 02
│   ├── 03
│   ├── 04
│   ├── 05
│   ├── 06
│   ├── 07
│   ├── 08
│   ├── 09
│   └── 10
└── run_array_parameters-path.sh
```

- `run_array_parameters-path.sh` content

```
#!/bin/bash
#SBATCH --partition=bigmem
#SBATCH --time=02:00:00
#SBATCH --mem=2G
#SBATCH --ntasks=1
#SBATCH --chdir=/network/lustre/iss01/home/maxime.kermarquer/demo/job_array/parameters
#SBATCH --error=outputs/job_%A_%a.log
#SBATCH --output=outputs/job_%A_%a.log
#SBATCH --array=1-10

module load MATLAB
export PARAMETER=$(sed -n ${SLURM_ARRAY_TASK_ID}p parameters/parameters-path.txt)
if [ ! -z $PARAMETER ]; then
    echo "Use parameter : ${PARAMETER}"
    matlab -nojvm -nodisplay -nodesktop < matlab-scripts/test_array.m
fi
```

- `Test_array.m` content

```
res = getenv('PARAMETER');

fprintf("Hello! I'm a MATLAB script and I get the parameter : %s \n", res );
```

- `parameters-path.txt` content

```
/network/lustre/iss01/home/maxime.kermarquer/demo/job_array/parameters/patients/01
/network/lustre/iss01/home/maxime.kermarquer/demo/job_array/parameters/patients/02
/network/lustre/iss01/home/maxime.kermarquer/demo/job_array/parameters/patients/03
....
```

Jobs array

Example

- Job array run 10 MATLAB scripts with 10 different paths by using `sed` shell command and `getenv` MATLAB function.

JOBID	PARTITION	NAME	USER	ST	TIME	CPU	MIN_MEMO	START_TIME	NODES	NODELIST(REASON)
126567_3	bigmem	run_array_parameters-path.sh	maxime.kermarquer	CG	0:11	1	2G	2019-10-04T18:11:01	1	hmb002
126567_9	bigmem	run_array_parameters-path.sh	maxime.kermarquer	CG	0:11	1	2G	2019-10-04T18:11:01	1	hmb002
126567_10	bigmem	run_array_parameters-path.sh	maxime.kermarquer	CG	0:10	1	2G	2019-10-04T18:11:01	1	hmb002
126567_1	bigmem	run_array_parameters-path.sh	maxime.kermarquer	R	0:11	1	2G	2019-10-04T18:11:01	1	hmb002
126567_2	bigmem	run_array_parameters-path.sh	maxime.kermarquer	R	0:11	1	2G	2019-10-04T18:11:01	1	hmb002
126567_4	bigmem	run_array_parameters-path.sh	maxime.kermarquer	R	0:11	1	2G	2019-10-04T18:11:01	1	hmb002
126567_5	bigmem	run_array_parameters-path.sh	maxime.kermarquer	R	0:11	1	2G	2019-10-04T18:11:01	1	hmb002
126567_6	bigmem	run_array_parameters-path.sh	maxime.kermarquer	R	0:11	1	2G	2019-10-04T18:11:01	1	hmb002
126567_7	bigmem	run_array_parameters-path.sh	maxime.kermarquer	R	0:11	1	2G	2019-10-04T18:11:01	1	hmb002
126567_8	bigmem	run_array_parameters-path.sh	maxime.kermarquer	R	0:11	1	2G	2019-10-04T18:11:01	1	hmb002

- Allow for example to run same MATLAB script on different patient.

Use parameter :
/network/lustre/iss01/home/maxime.kermarquer/demo/job_array/parameters/patients/01

< M A T L A B (R) >
Copyright 1984-2017 The MathWorks, Inc.
R2017b (9.3.0.713579) 64-bit (glnxa64)
September 14, 2017

For online documentation, see http://www.mathworks.com/support
For product information, visit www.mathworks.com.

>> >> >> >> Hello! I'm a MATLAB script and I get the parameter :
/network/lustre/iss01/home/maxime.kermarquer/demo/job_array/parameters/patients/**01**
>> >> >> >> >>

Use parameter :
/network/lustre/iss01/home/maxime.kermarquer/demo/job_array/parameters/patients/02

< M A T L A B (R) >
Copyright 1984-2017 The MathWorks, Inc.
R2017b (9.3.0.713579) 64-bit (glnxa64)
September 14, 2017

For online documentation, see http://www.mathworks.com/support
For product information, visit www.mathworks.com.

>> >> >> >> Hello! I'm a MATLAB script and I get the parameter :
/network/lustre/iss01/home/maxime.kermarquer/demo/job_array/parameters/patients/**02**
>> >> >> >> >>

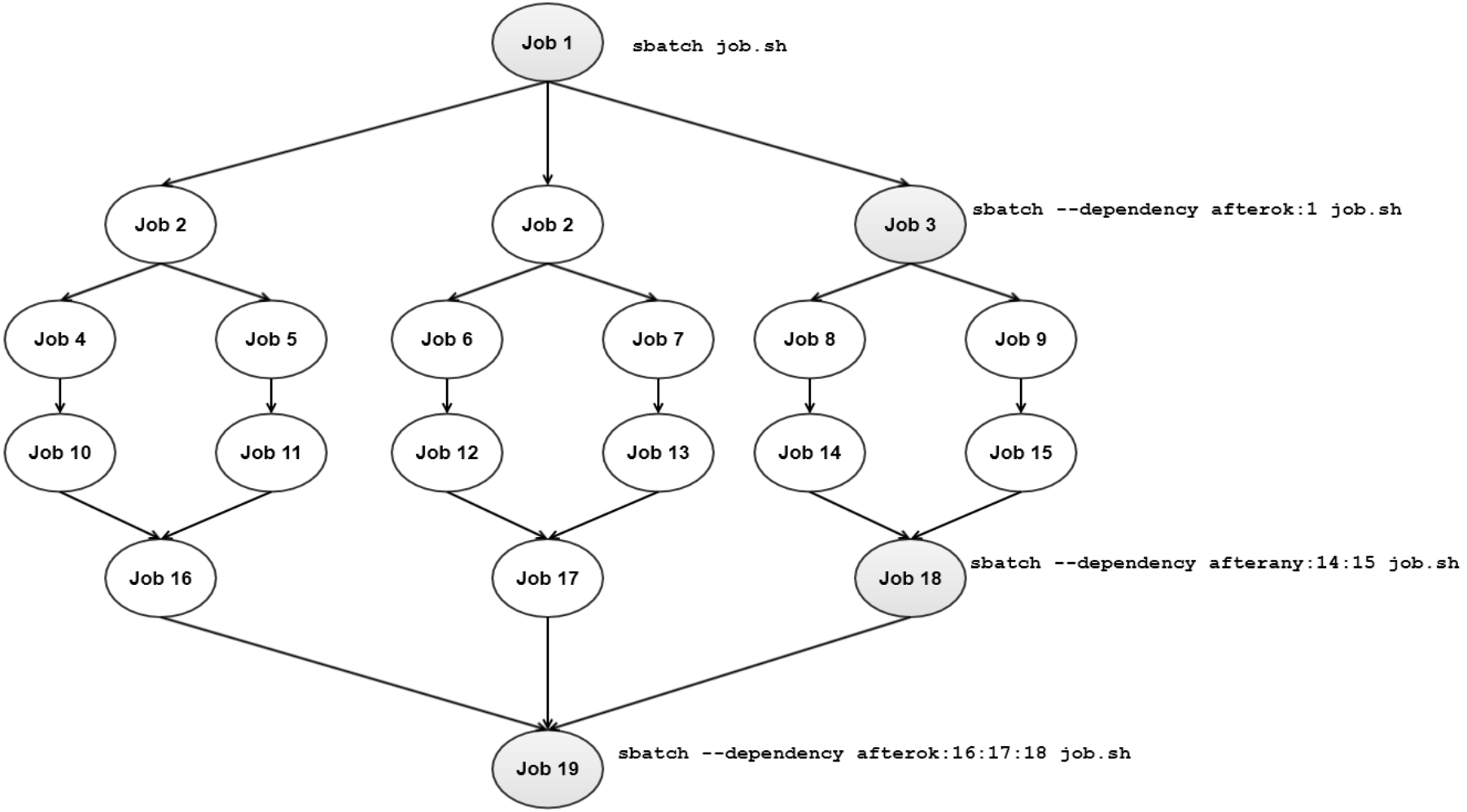
Jobs dependency

Options

- SLURM gives the possibility to set up pipeline of jobs with dependencies between them
- `sbatch` command with `--dependency` option, a type of dependency and a jobid
- Type of dependency
  - Begin execution after the specified jobs have begun execution.
    - `after`
  - Begin execution after the specified jobs have terminated.
    - `afterany`
  - Begin execution after the specified jobs have successfully executed
    - `afterok`
  - Begin execution after the specified jobs have terminated with failed state
    - `afternotok`

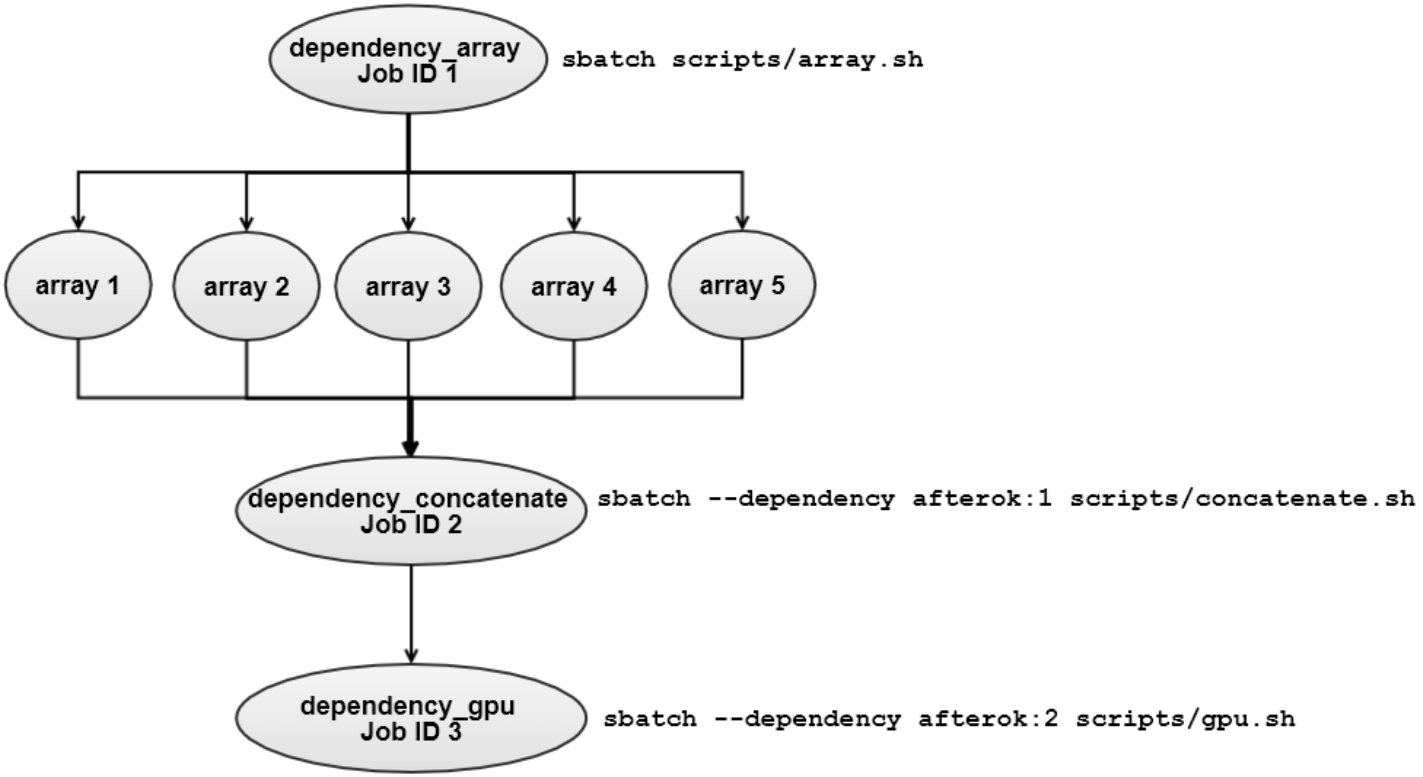
Jobs dependency

Dependency graph



Jobs dependency

Example



Scheduling policy

Priority

- Priority are determined with the following expression:

```
Priority =
(PriorityWeightAge)        * (age_factor)         +
(PriorityWeightFairshare)  * (fair-share_factor)  +
(PriorityWeightJobSize)    * (job_size_factor)
```

- Slurm's fair-share is a floating point number between 0,0 and 1,0.
  - Reflects the amount of computing resources the user's jobs have consumed.
  - Higher is the value, higher is the placement in the queue of jobs waiting to be scheduled.

- The limits per user are **300 cores and 1200G** but we can raise them for special requests.

Scheduling policy

Classification of jobs

- To share equitably the use of the cluster, jobs are classify for `normal` partition

	Class 1	Class 2	Class 3	Class 4	Class 5
% Disponible	100,00%	80,00%	50,00%	30,00%	20,00%
Nb cores	700	560	350	210	140
Memory (Mb)	3194880	2555904	1597440	958464	638976
Memory (Gb)	3120	2496	1560	936	624
Nodes	25	20	12	7	5

	≤ 4h	≤ 1 semaine	≤ 2 semaine	≤ 3 mois
4G				
8G				
16G				
32G				
64G				
122G				

5 Deep learning

Which framework are available ?

Deep learning
Resources

- To perform Deep Learning ICM, you can use the GPU node with NVIDIA Tesla P100 cards

- In python modules are installed the DL frameworks

TensorFlow™ PYTORCH NVIDIA CUDA ANACONDA®

Conclusion

Conclusion
Contact and materials

- The slides will be available
- The wiki (only accessible from the ICM network)
https://dokuwiki.icm-institute.org/doku.php?id=cluster:wiki
- If you need help, you can request
  - Open an issue in ***Calculateurs-07*** category
  - By mattermost, direct message to Maxime Kermarquer or computer cluster channel
  - By email : maxime.kermarquer@icm-institute.org

Thank you for your attention !