Hands-on Liger supercomputer

- Pierre-Emmanuel Guérin
- Davide Rovelli
- Hugues Digonnet

supercomputing.ec-nantes.fr / @cnscfr

CENTRALE NANTES SUPER COMPUTING

Topics

- System overview
  - What is a supercomputer?
  - Inside Liger
- Hands-on
  - First steps
  - Demonstration and questions

CENTRALE NANTES SUPER COMPUTING

System overview

What is a supercomputer ?

A supercomputer is a cluster of computers linked with a high speed network.

The performance is commonly measured in **FL**oating-point

Operations **P**er **S**econds (**FLOPS**).

HPC = **H**igh **P**reformance **C**omputing → the main discipline exploiting supercomputing

What is a supercomputer ?

- Many servers = many processors
- The processors are connected by a high-speed internal network (Infiniband)
- To exploit the number of processors, programs need to run in **parallel** (written with MPI, OpenMP etc.)
- All nodes are connected to a parallel storage (GPFS, Lustre ...) which needs special libraries
- Programs have to be submitted to the batch system (SLURM, LSF ...), a “supercomputer scheduler”

What a supercomputer is not

- It has overclocked high-speed processors ? **NO**
- It has a big internal RAM ? **NO**
- It runs Windows ? **NO**
- CPU runs at higher frequency than a desktop PC ? **NO**
- It will run my software without changes ? **NO**
- It will run my software with millions of threads ? **NO**
- It will run my old trusted executable ? **NO**
- It will run Excel spreadsheets ? **NO**

Some reasons to buy a “big computer”

What types of big problem might require a "Big Computer" ?

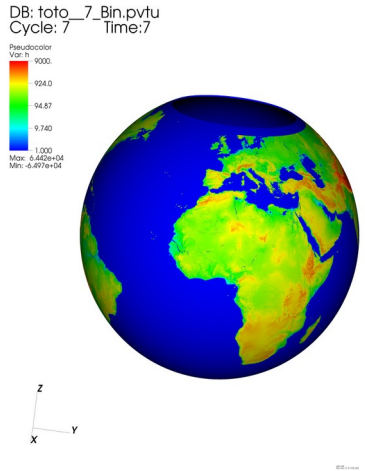
- **Compute intensive** : a single problem requiring a large amount of computation.
- **Memory intensive** : a single problem requiring a large amount of memory.
- **Data intensive** : a single problem operating on a large amount of data.
- **High throughput** : Many unrelated problems to be executed over a long period.

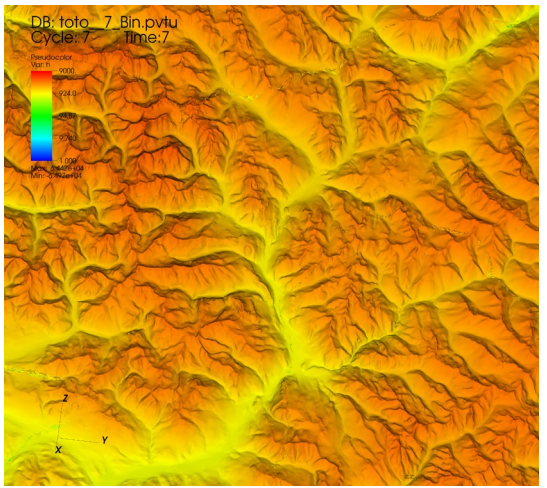
Typical applications

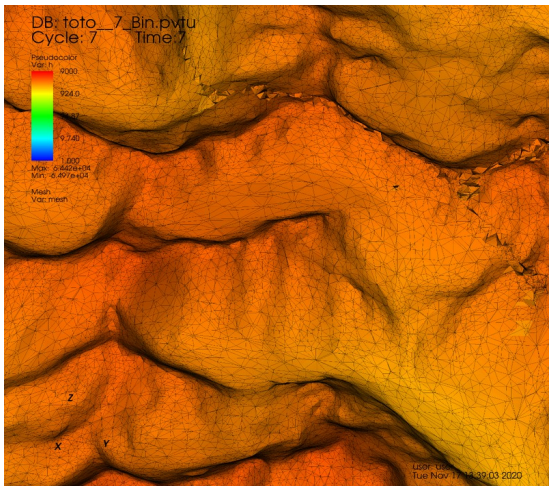
They are used for different applications such as

- digital simulation
- research
- weather forecasting
- nuclear domain
- ...everything that requires a lot of calculation!

Example: earth surface mesh







Accuracy: map each point every 200m on the entire earth, at the same time.

Adapting programs for HPC

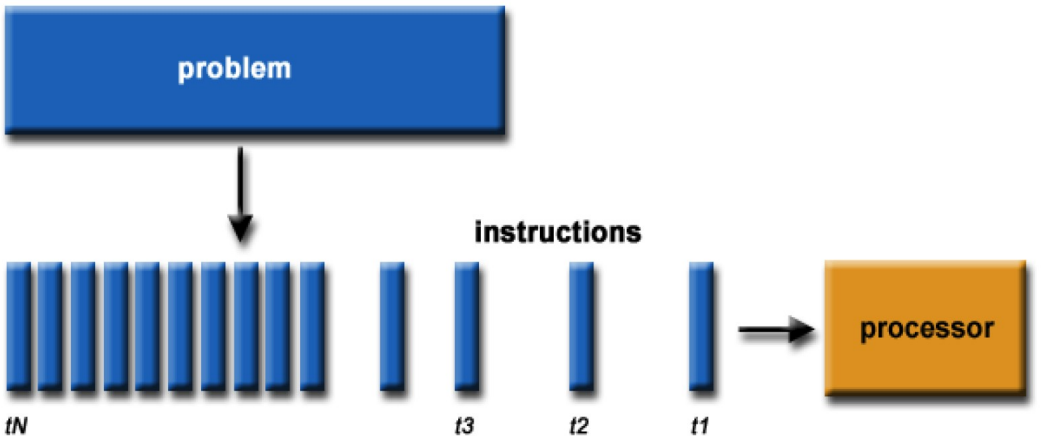
A computational problem:

- Should be able to be broken apart into discrete pieces of work that can be solved simultaneously
- Should be able to execute multiple program instructions at any moment in time
- Should be able to be solved in less time with multiple compute resources than with a single compute resource

Parallel vs Serial: Serial

Traditionally, software was written for **serial** computation:

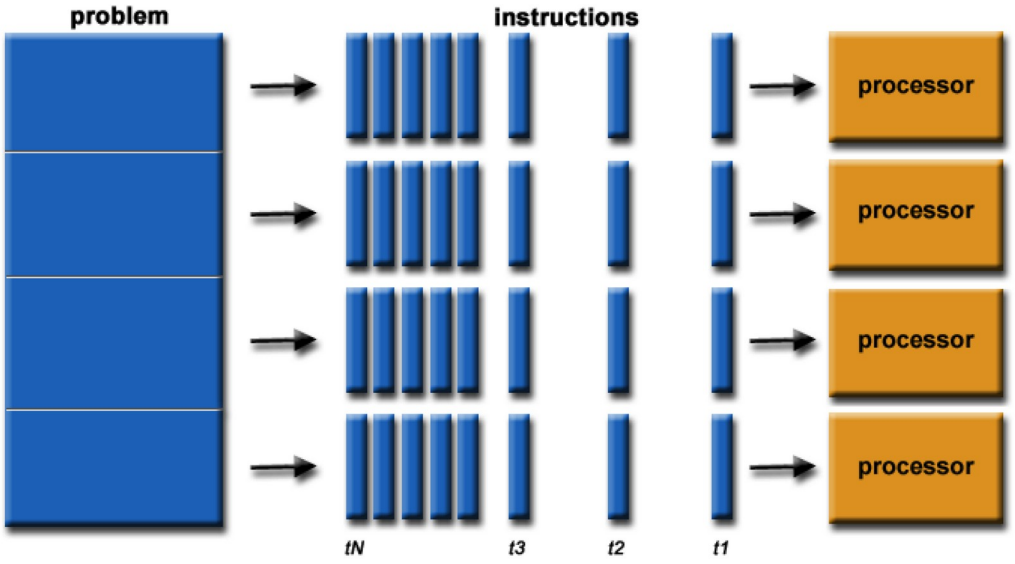
- A problem is broken into a discrete series of instructions
- Instructions are executed sequentially one after another
- Executed on a single processor
- Only one instruction may execute at any moment in time



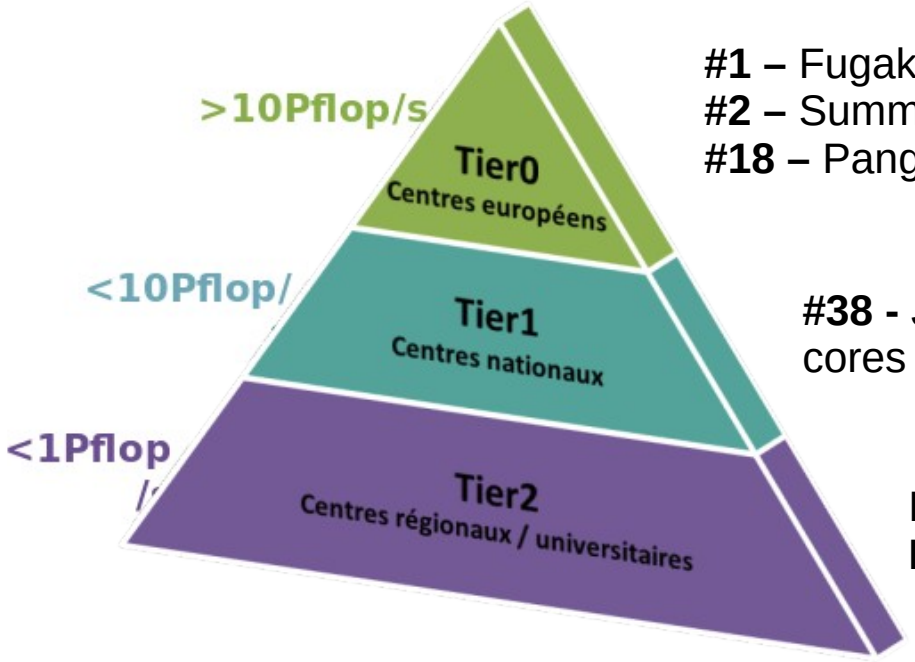
Parallel vs Serial: Parallel

Parallel computing is the key for fast computation:

- A problem is broken into discrete parts that can be solved concurrently
- Each part is further broken down to a series of instructions
- Instructions from each part execute simultaneously on different processors
- An overall control/coordination mechanism is employed



Supercomputer performance pyramid

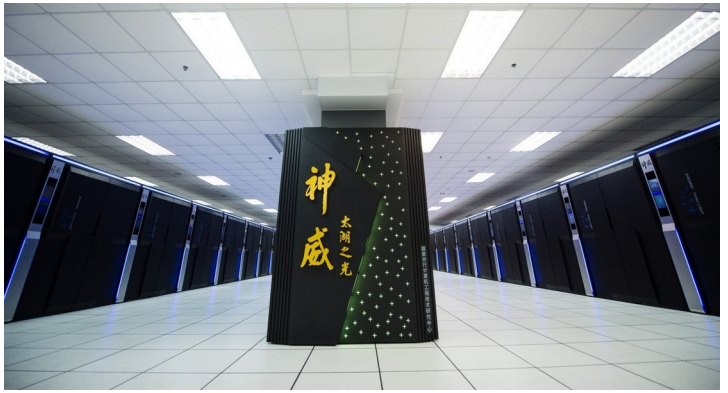


#1 – Fugaku. 442 PFLOPS, ~7M cores, JPN
#2 – Summit, 148 PFLOPS, ~2M cores, USA
#18 – Pangea III, 17 PFLOPS, 448k cores, FRA

#38 - JOLIOT-CURIE ROME, 6.9 PFLOPS 197k cores

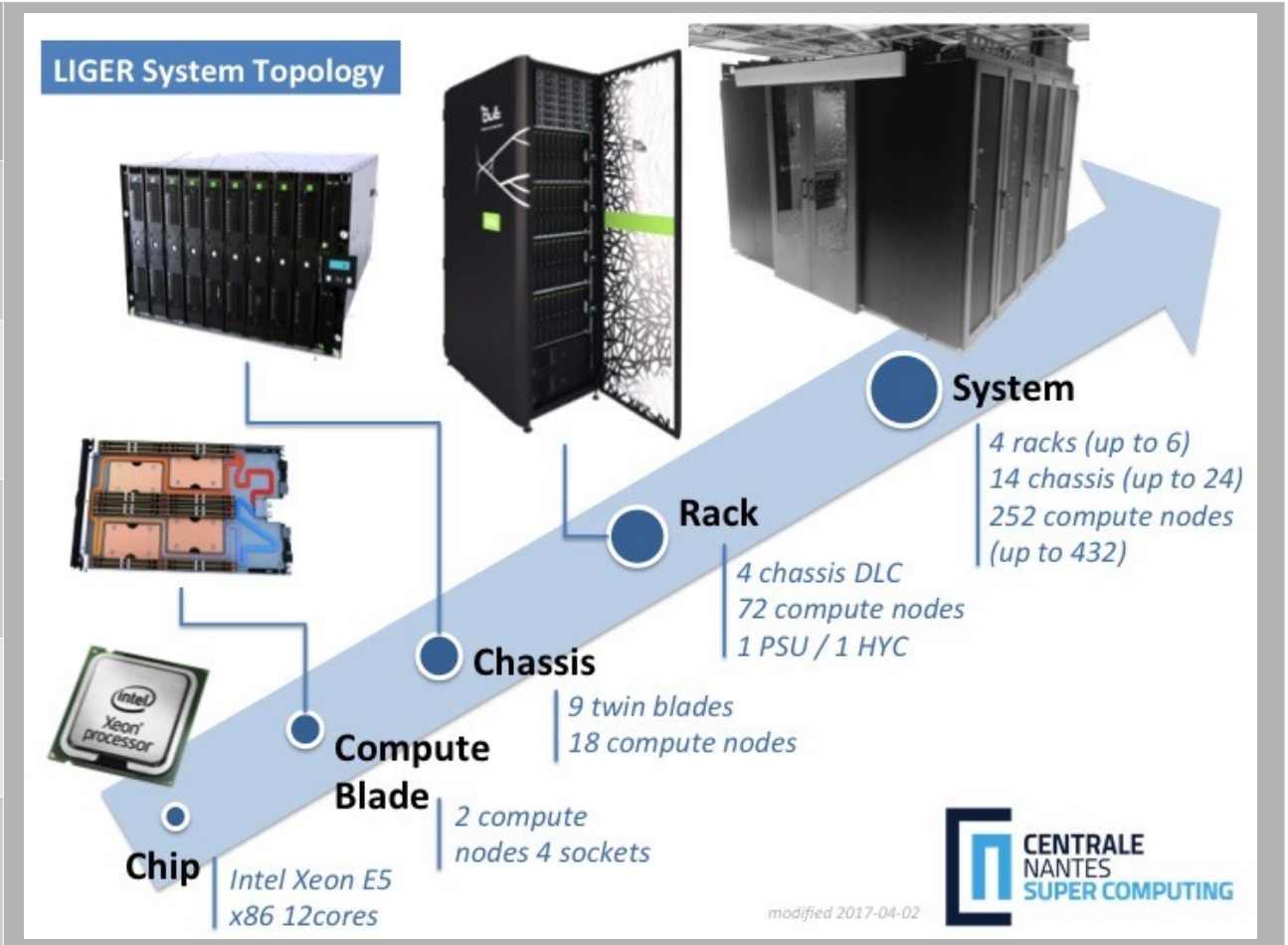
Romeo, 1 PFLOP, FRA
Liger, 0.29 PFLOP, 6k cores, FRA

They look like that

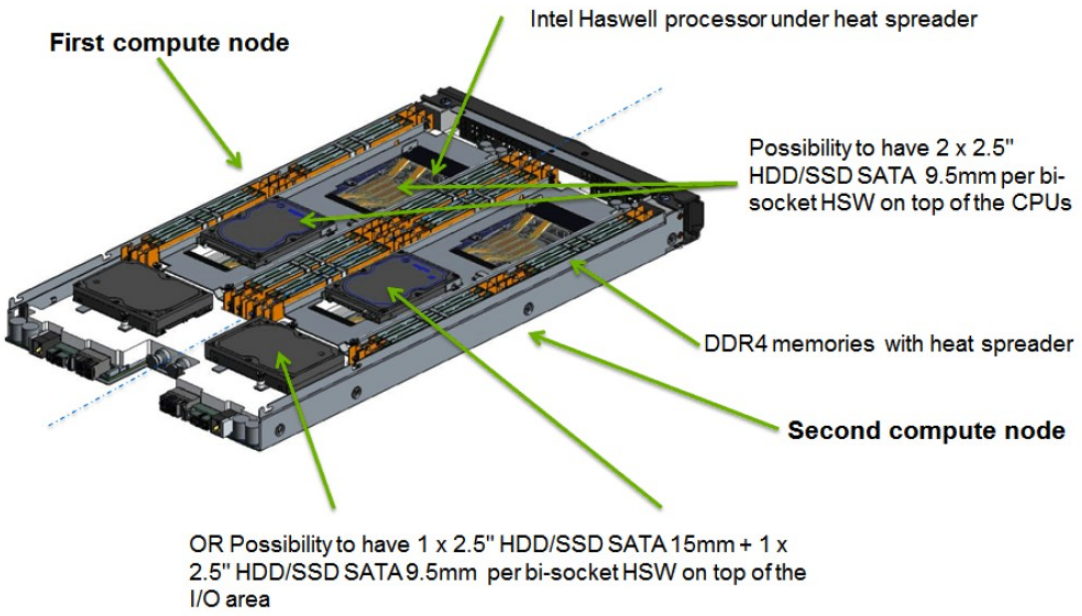


Liger: our sumpercomputer

280 TFLOPS
6384 cores
28 GPU
56 Gbit/s
266 nodes
+36 TB



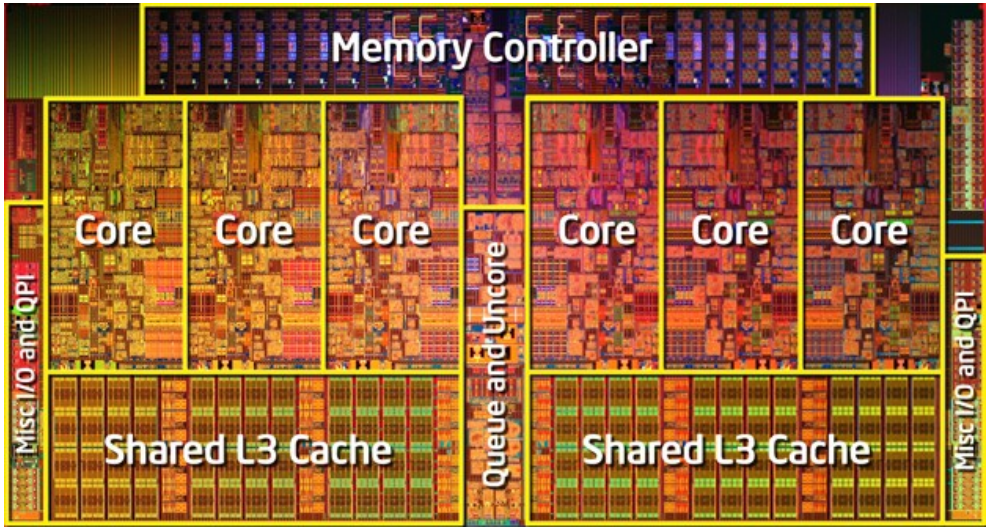
Liger: inside a blade



Terminology:

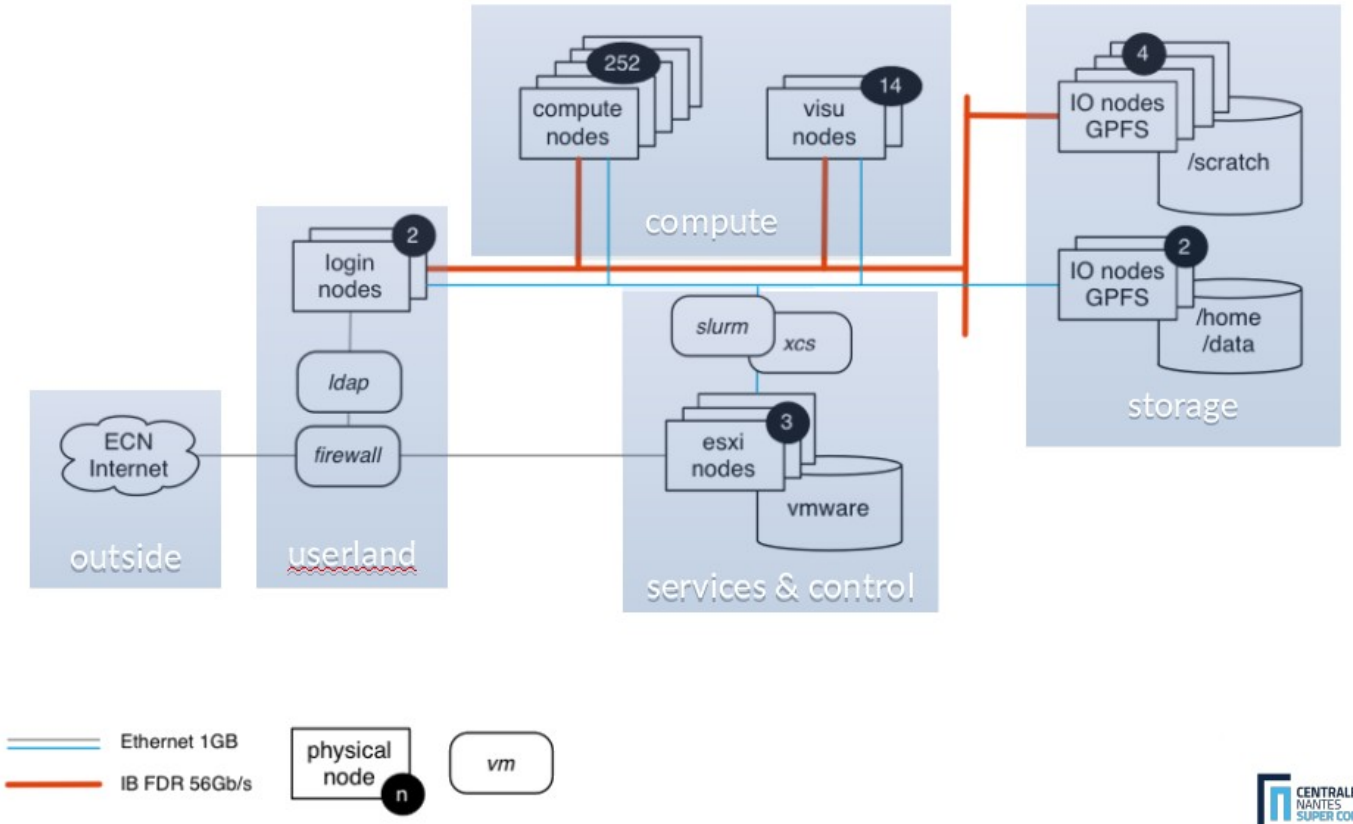
- **Node:** a single “computer”, server or machine. Among other elements it contains
- **2 CPUs:** the electronic unit performing basic arithmetic operations. It contains
- **12 Cores:** basic unit of computation. A core can run one process.

Liger: inside a CPU



532 CPUs in Liger.

Liger: system topology



Liger in numbers

- 1 day compute on Liger = 17.5 years on 1 proc
- 1 day compute on 1 proc = 13.5 secs on Liger
- As hot as 30 houses using heaters at the same time
- As fast as 10k modern smartphones (working in parallel of course!)
- The 1st world is as fast as 400 Liger
- 5th fastest regional machine in France (Tier2) and the fastest in Bretagne/Pays de la Loire

Hands-on

Request an account

- Know your Rights & Duties as a Liger User
  - https://supercomputing.ec-nantes.fr/charter
  - READ, AGREE & SIGN charters listed above
  - Send ONLY pages with your signature to your teacher.
  - Preferred Scan+PDF to paper.
- Use SVP to send the documents and for any technical problem
  - https://svp.ec-nantes.fr (supercomputing)

Types of job

- **Visualisation**
  - Launch visual applications through the web portal
  - Use remote visualisation to use the app on Liger
  - *Matlab*, *Blender*, *Paraview*, *Fiji etc…*
- **Pure computation**
  - Connect to Liger via CLI (command line interface)
  - Run programs directly in the system
  - *Python*, *C++*, *Rust*, *optimised scientific framework*, *non-GUI tools etc.*

Types of job

- **Visualisation**
  - Launch visual applications through the web portal
  - Use remote visualisation to use the app on Liger
  - *Matlab*, *Blender*, *Paraview*, *Fiji etc…*

Focus: requires Linux shell knowledge

- **Pure computation**
  - Connect to Liger via CLI (command line interface)
  - Run programs directly in the system
  - *Python*, *C++*, *Rust*, *optimised scientific framework*, *non-GUI tools etc.*

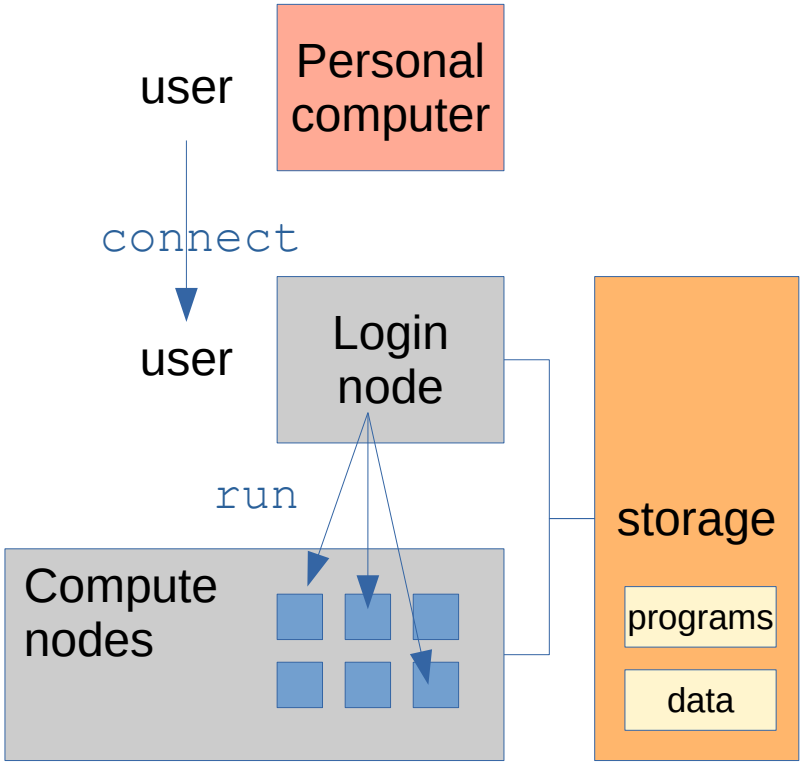
Connect to Liger

- Client tool to connect on remote console:
  - Windows : PowerShell, putty, cygwin, mobaxterm
  - Mac/Linux : xterm, xquartz (only mac)
- Use a VPN to connect to Centrale Nantes network
- SSH secure protocol

```
$ ssh myUsername@liger.ec-nantes.fr
```

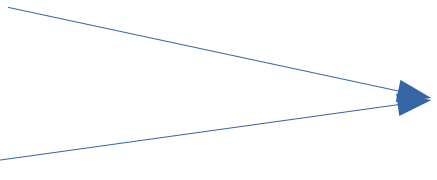
Liger : User environment

- You have 3 directories
- You can compile and test codes on login nodes
- You can use available softwares/libraries
- And you can submit jobs on nodes.



User Env : Filesystems & storage

- /scratch
  - 815 TB, 1 000 000 files quota per user
  - Your directory is $SCRATCHDIR
  - Computations and temporary files
- /home
  - 30 TB, 5GB quota soft per user
  - Your directory is $HOME
  - Sources files
- /data
  - 45 TB, quota per group={100GB and 2 million files}
  - Your project directory is $DATADIR
  - Permanent projects data and group sharing data



User space

Project space

Move files to Liger

- SCP (or WinSCP for Windows): secure copy
  - Example: tranfer program to */home*

```
$ scp ./Desktop/program.c LIGER-ID@liger.ec-nantes.fr:~
```

  - WinSCP: GUI, same principle
- Download directly on Liger: git, wget etc.
  - Example: clone git repository on scratch

```
$ git clone https://repo.git $SCRATCHDIR
```

Load programs: *modules*

- Your environment is initally empty: no programs installed
- Modules is a tool to load or unload software packages.
  - List available software

```
$ module avail
```

  - Load python

```
$ module load python
```

Job submission

- Compute resources are managed by a scheduler:
  - Liger uses **SLURM**
- Jobs are submitted to the scheduler
  - The scheduler choose available nodes (job running)
  - Or the computation is queued (job pending)

Job submission

- With slurm commands you can run program on compute nodes.
  - Tell the SLURM what to run
  - SLURM will find the available resources and run the program

```
$ srun PROGRAM # run a job in the foreground
```

```
$ sbatch SCRIPT # run a job in the background
```

Readings

- Tutorials & Documentation available at
  - supercomputing.ec-nantes.fr/publications/tutorials
  - supercomputing.ec-nantes.fr/publications/tutorials

Demonstration

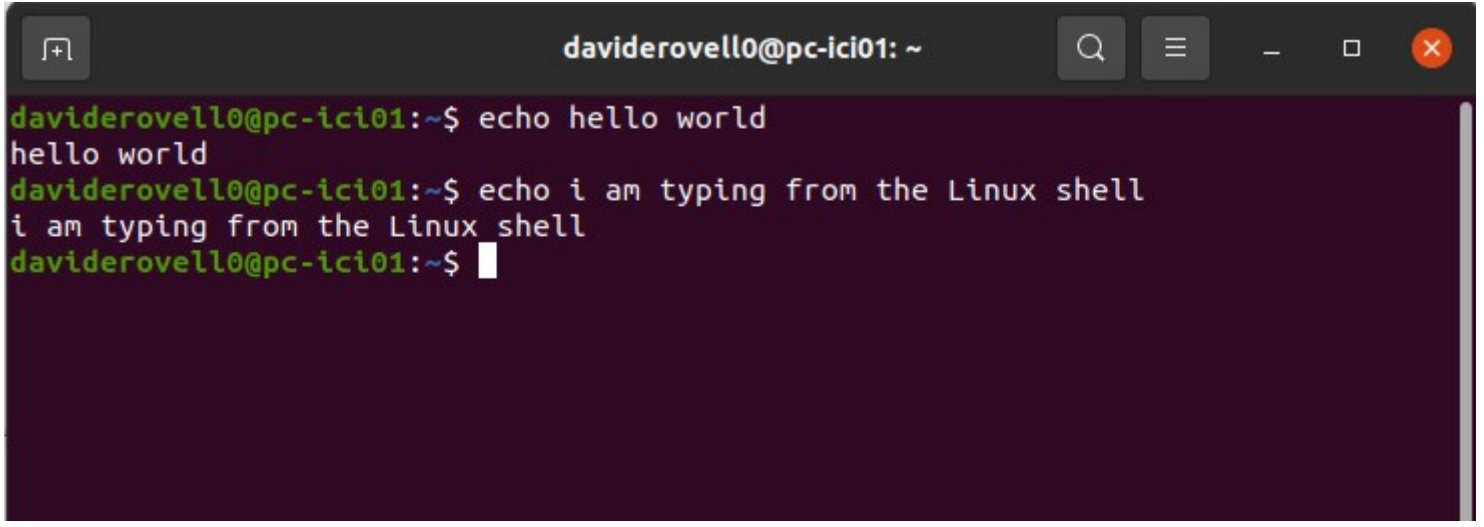
- Remote vizualisation
- Liger CLI
  - My* commands
  - quotas
  - modules
  - running a job (slurm scripts examples)


Questions?

Linux shell basics

The Linux shell - terminal

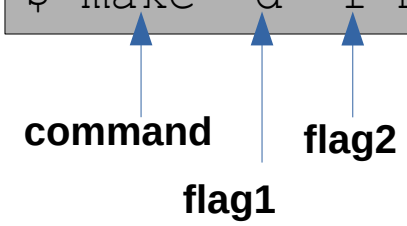
- No Graphical User Interface
- Issue commands through a CLI: command lone interface



Issuing commands

- A command is a program that corresponds to a string of text. Use return to send a command, ctrl-C to interrupt it.
- A command can have **options**, set through **flags**.

```
$ make -d -f Makefile
```



- The "**-h**" flag shows a help guide for most commands

Navigating directories

- *pwd* – shows which directory you are in
- *ls* – list the files in the current directory
- *cd* – change to another directory

The base folder (top of the tree) is represented by “**/**”

The current folder is represented by “**.**”

The parent folder is represented by “**..**”

Editing files

- *cp* – copy a file to another location
- *mv* – move the file to another location (used for renameing as well)
- *rm* – remove a file, **-r** flag for recursive and folders

General rule: all commands are executed in the current folder (*pwd*), to execute a command in another folder use its path:

/absolute/path/to/file *relative/path/to/file*

File operations

- Text editors: *nano*, *vi*, *gedit (requires GUI)*
  - Relies on a lot of key combinations, can be hard at the beginning. Use an editor wherever possible
- View file content: *cat*, *less etc*

```
$ cat your_file.txt
```

Run programs

- *gcc* – C / C++ compiler
- *python3* – run a Python script
- Javac – run a Java program
- ...any installed program. Install with package manager:
  - Ubuntu, Debian: *apt*
  - RHEL: *yum*

Useful resources

There’s much much more!

- https://supercomputing.ec-nantes.fr/publications/tutorials
- https://projects.ncsu.edu/hpc/Documents/unixtut/
- http://swcarpentry.github.io/shell-novice/