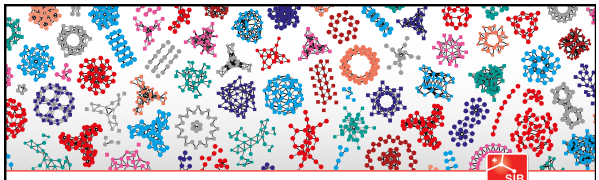


SIB Swiss Institute of Bioinformatics

High-Performance Computing - UNIL Infrastructure Training

Robin Engler (SIB/Vital-IT), Sebastien Moretti (SIB/Vital-IT, UNIL/DEE)
Grégoire Rossier (SIB Vital-IT & SIB Training)

Lausanne, 20 September 2019 www.sib.swiss

Who are the participants?

- 22x FBM: DEE, CIG, DBC, DMF, DNF
- 13x FGSE: IDYST, IGD, ISTE
- 4x HEC: DESI, Finance
- 2x CHUV
- 1x FDCA
- 1x IP
- 1x ISS
- 1x SLI
- 1x Décanat – Lettres
- 1x Hopital Jules Gonin

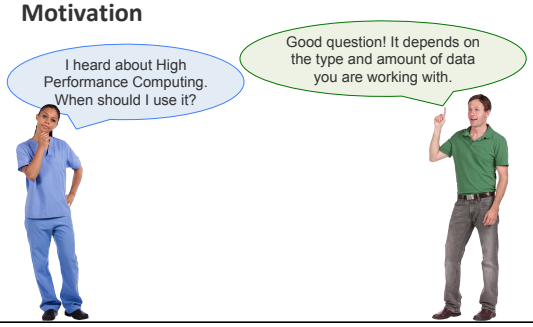
Objectives of the course

- **Basic knowledge** on high performance computing.
- Computer **clusters at UNIL**: overview, usage and tips.
- **Support** and **cost model**: overview.

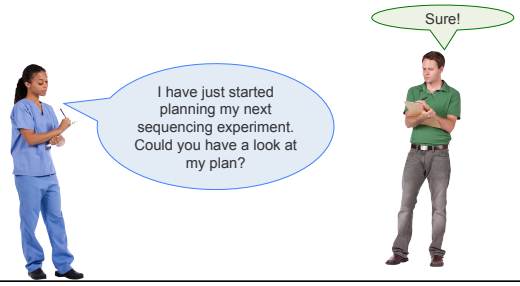
Outline

1. Motivation
2. Presentation of the DCSR
3. General High Performance Computing concepts
4. UNIL HPC cluster infrastructure overview
5. Cluster storage locations
6. Slurm commands

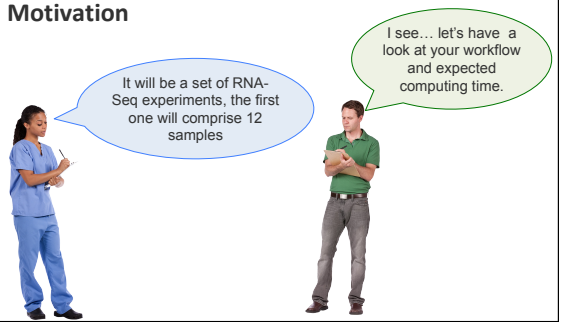
Motivation



Motivation

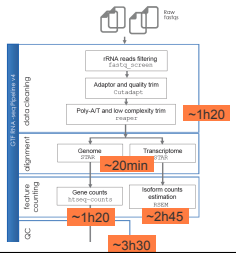


Motivation



RNA-seq experiment with 12 samples

(source: Genomic Technologies Facility, UNIL)



Requirements:
Memory: 32GB
#CPU: 8

Detail per sample (~25M single-reads)

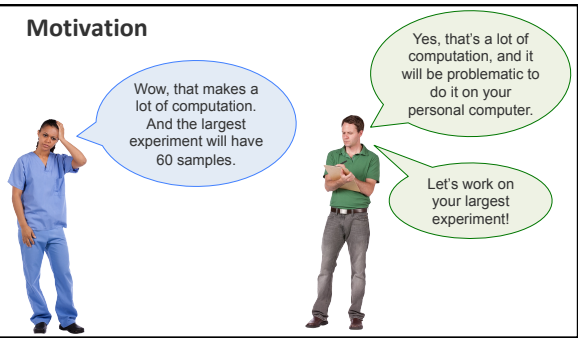
```
Resource usage summary:

    CPU time :                   185679.00 sec.
    Max Memory :                 27525.09 MB
    Average Memory :             6459.94 MB
    Total Requested Memory :     32000.00 MB
    Delta Memory :               4474.91 MB
    (Delta: the difference between total requested memory and
    actual max usage.)
    Max Swap :                   48706 MB

    Max Processes :              14
    Max Threads :                151
```

Execution time: ~9h40

Motivation



RNA-seq experiment with a larger number of samples

- 1 species
- 5 conditions
- Triplicates
- 4 time points

→ 60 samples

High coverage: 50 x 10^6 reads / sample (>10 GB)

→ >600 GB data before analysis

Motivation

For this project, you definitely need an HPC cluster, which usually offers such features:

Features	Typical HPC
Dataset size (including intermediate data)	Several TB
Memory size	Up to 1500 GB
Number of cores	Up to 128
Number of simultaneous jobs	Up to 200
Software availability	Large range
Software versioning	Several
Hardware maintenance	Yes
Hardware failure	Less sensitive
Data backup	Multiple & secure

When is usage of a cluster justified?

- **size issues** : computation time / memory / storage space.
- programs optimized with execution in **parallel**.
- data **security** & **privacy**

Outline

1. Motivation
2. Presentation of the DCSR
3. General High Performance Computing concepts
4. UNIL HPC cluster infrastructure overview
5. Cluster storage locations
6. Slurm commands

DCSR - Missions

Division calcul et soutien à la recherche / Scientific Computing and Research Support Unit

- provide **computation** and **storage** resources to the UNIL community
- provide transverse **expertise** to the UNIL faculties and its affiliated institutes

https://www.unil.ch/ci/dcsr-en

DCSR - Structure

- Technical and sysadmin specialists:
  - ✓ Operate, maintain and evolve infrastructure
- Scientific advisors / consultants:
  - ✓ Support specific research project about resources usage
  - ✓ Consulting services

https://www.unil.ch/ci/dcsr-en

Outline

1. Motivation
2. Presentation of the DCSR
3. General High Performance Computing concepts
4. UNIL HPC cluster infrastructure overview
5. Cluster storage locations
6. Slurm commands

General computing concepts

- **Program**: set of instructions intended to achieve a goal and stored in one (or more) file(s). *Spreadsheet (Excel)*
- **Process**: an executing program, identified by a unique process identifier (PID). *Program execution by user A*
- **Thread**: a single unit of execution (task) within a process. Multiple threads can exists within a single process and can be executed concurrently or in parallel (multi-threaded process). *Printing Saving*

General computing concepts

each **process** has one or more threads.

each **thread** belongs to one process only.

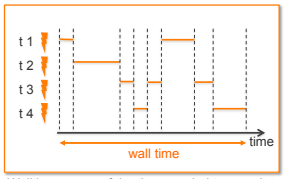
General computing concepts – concurrent vs parallel

To execute threads in parallel (and save time), you need:

- Multi-core processors
- Programs written to allow parallelism

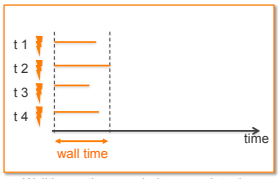
General computing concepts – concurrent vs parallel

Process with 4 ***concurrent threads***.
All run on the same CPU, alternately.



Walltime = sum of the time needed to complete all 4 threads individually

Process with 4 ***parallel threads***.
Each thread runs on a different CPU.



Walltime = time needed to complete the longest thread

General computing concepts – parallel vs distributed

Parallel computing

- Computation on a single computer
- Software able to run multiple threads in parallel
- Machines with a large number of CPUs, and large amounts of memory

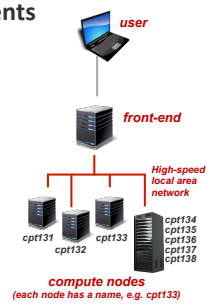
Distributed computing

- Several different machines are connected on the same network
- Resources (CPUs, memory) distributed across these machines

Compute cluster components

Hardware

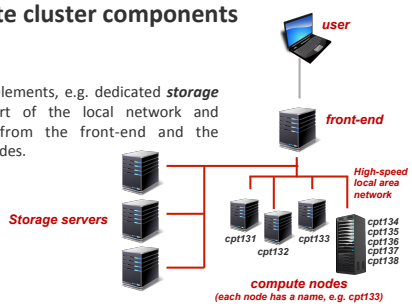
- a **front-end** machine
  - users connect/login
  - job submission
- **compute nodes**
  - no users' access
  - computation



Compute cluster components

Hardware

Additional elements, e.g. dedicated ***storage servers***, part of the local network and accessible from the front-end and the compute nodes.



Compute cluster components

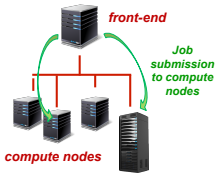
Example of cluster with 146 compute nodes

Nodes name	Archit.	CPUs	Speed	RAM	Total CPUs
cpt001 to cpt064	Intel	8	3 GHz	16 GB	64 * 8 = 512
cpt065 to cpt080	Intel	4	3 GHz	16 GB	16 * 4 = 64
cpt081 to cpt096	Intel	4	3 GHz	8 GB	16 * 4 = 64
cpt097 to cpt103	AMD	64	2.3 GHz	512 GB	7 * 64 = 448
cpt104,105	AMD	48	2.3 GHz	256 GB	2 * 48 = 96
cpt106 to cpt138	Intel	8	2.8 GHz	16 GB	32 * 8 = 256
cpt139 to cpt146	Intel	16	2.5 GHz	24 GB	8*16 = 128

Compute cluster components

Software

Job Scheduler
=
Local **R**esource **M**anagement **S**ystem
(LRMS)
=
Workload Manager



Job scheduler

Automate and optimize resource allocation when there is more work than resources.

- **Optimize the assignment of jobs to individual compute nodes**, based on job hardware requirements and current hardware availability.
- **Management of job priorities**, based on user profiles and user job-specific parameters. For instance users that already have jobs running will see their priority decreased.
- **Monitoring of job execution and enforcement of resource limits**. Jobs that exceed certain set limits, e.g. taking too much time or using too much memory, are terminated. Users are notified when a job completed.
- **Accounting of resource usage** of individual users – used to generate PI's monthly bill.

Job scheduler

Examples of popular job scheduler software

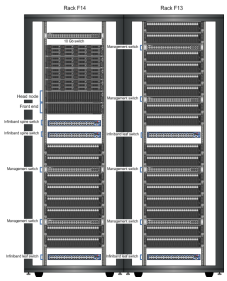
- **LSF** (Load Sharing Facility)
- **Slurm** (Simple Linux Utility for Resource Management)

Outline

1. Motivation
2. Presentation of the DCSR
3. General High Performance Computing concepts
4. UNIL HPC cluster infrastructure overview
5. Cluster storage locations
6. Slurm commands

WALLY

- `wally-front1.unil.ch` (your UNIL login)
- 1536 cores
- 91 nodes with 64 GB RAM and 16 core CPU
- 5 nodes with 96 GB RAM and 16 core CPU
- One front end with 768 GB RAM
- Recommended for standard jobs

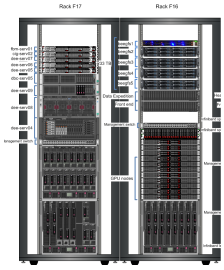


http://wally-head.unil.ch/ganglia/

AXIOM

- `axiom-front1.unil.ch` (your UNIL login)
- 1936 cores
- 33 nodes with up to 768 GB RAM (1.5 TB soon) and up to 128 (256 soon) core CPU
- One front end with 768 GB RAM
- Recommended for high RAM jobs

More nodes very soon...



http://wally-head.unil.ch/ganglia/

Connect to front end machines

```
ssh <UNIL username>@wally-front1.unil.ch

ssh <UNIL username>@axiom-front1.unil.ch
```

Slurm

- Free and open source job scheduler
- Used in many of the world's largest HPC centers

slurm
workload manager

Slurm - partitions

- Group of compute nodes to which a job is assigned by the user.
- A node can belong to more than one partition.

Slurm – partitions specificities

- **Job priority value**, influences how quickly the job starts running.
- **How long** a job can run (runtime limit).
- **How much resources** can be requested (e.g. CPUs, memory).
- **How many jobs** a user can run simultaneously.
- **Who can access** the partition.

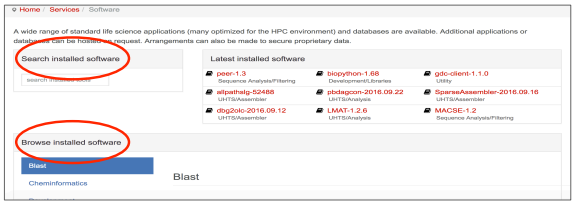
Slurm - partitions on Wally and Axiom

Partition name (queue)	Applies to	CPU	RAM	Job run time	Concurrent job limit
normal [default]	/scratch/wally/	1 cpu [def] 16 cpu [max]	2 GB/CPU [def] 64 GB [max]	12h [def] 24h [max]	No limit [might change in the future]
long [lower priority]				5 days [def] 10 days [max]	
ax-normal [default]	/scratch/axiom	1 cpu [def] 128 cpu [max]	2 GB/CPU [def] 768 GB [max]	12h [def] 24h [max]	No limit [might change in the future]
ax-long [lower priority]				5 days [def] 10 days [max]	

[def] = default, [max] = maximum

Default is half the max time !

Bioinformatics software on Vital-IT website



https://www.vital-it.ch/services/software

Bioinformatics software on UNIL cluster

General path

/software/<category(ies)>/<package>/<version>/

category(ies): scientific/technical domain(s)

- UHTS, Phylogeny, MassSpec, Development, R, etc.

package: package name

version: package version

Bioinformatics software on UNIL cluster

```
/software/UHTS/Aligner/bowtie2/2.3.4.1
/software/UHTS/Quality_control/fastqc/0.11.7
/software/Phylogeny/paml/4.9i
/software/Development/java/1.8.0_222
/software/R/3.5.1
```

Using software on UNIL cluster

RULE: to use a bioinformatics software, you need to

1) load the Vital-IT software stack
2) add the bioinformatics software

```
module load Bioinformatics/Software/vital-it

module add R/3.5.1
```

Using software on UNIL cluster

Command to load/unload a software: on Vital-IT website

Using software on UNIL cluster

List available module files

```
$ module avail
```

List modules that are currently loaded

```
$ module list
```

Using software on UNIL cluster

Load software

```
$ module add <module path>
```

Unload software

```
$ module rm <module path>
```

```
$ module purge (removes all modules)
```

Using software on UNIL cluster - versioning

- Version might change over time
- Several versions available for some software
- Might make your experiments invalid
- Check version(s) available before you start experiments

```
$ bowtie2 --version
… version 2.3.0
```

vit_soft

- Utility to retrieve information about software using the terminal.
- Need first to load the Vital-IT software stack.

```
$ vit_soft -search <XYZ>     text search for packages (case insensitive)
           -package <XYZ>    get package information
           -module <XYZ>     get module commands for a package
           -binary <ABC>     search package providing this binary
           -latest           get latest installed packages
           -help             more details
```

vit_soft - examples

```
$ vit_soft -s blast
ncbi-blast-2.7.1+-1.el7.x86_64
ncbi-blast-2.6.0+-1.el7.x86_64
ncbi-blast-2.3.0+-1.el7.x86_64
ncbi-blast-2.2.31+-1.el7.x86_64
blast-2.2.26-1.el7.x86_64

$ vit_soft -s Blas
ncbi-blast-2.7.1+-1.el7.x86_64
ncbi-blast-2.6.0+-1.el7.x86_64
ncbi-blast-2.3.0+-1.el7.x86_64
ncbi-blast-2.2.31+-1.el7.x86_64
blast-2.2.26-1.el7.x86_64
```

Case insensitive and partial match

vit_soft - examples

```
$ vit_soft -m blast
module add Blast/blast/2.2.26

$ vit_soft -m ncbi-blast
module add Blast/ncbi-blast/2.7.1+
module add Blast/ncbi-blast/2.6.0+
module add Blast/ncbi-blast/2.3.0+
module add Blast/ncbi-blast/2.2.31+
```

Exact search only

vit_soft - examples

```
$ vit_soft -s blastp
No package found!

$ vit_soft -m blastp
package blastp is not found
```

vit_soft - examples

```
$ vit_soft -b blastp
RMBlast-2.6.0+-1.el7.x86_64
ncbi-blast-2.7.1+-1.el7.x86_64
ncbi-blast-2.6.0+-1.el7.x86_64
ncbi-blast-2.3.0+-1.el7.x86_64
ncbi-blast-2.2.31+-1.el7.x86_64
```

Exercises 1 to 3…

https://edu.sib.swiss/mod/page/view.php?id=4085