Training day SLURM cluster

- Context
- Infrastructure
- Software usage
- SLURM directives
- For further with SLURM
- Best practices
- Support

Context

PRE-REQUISITE : LINUX

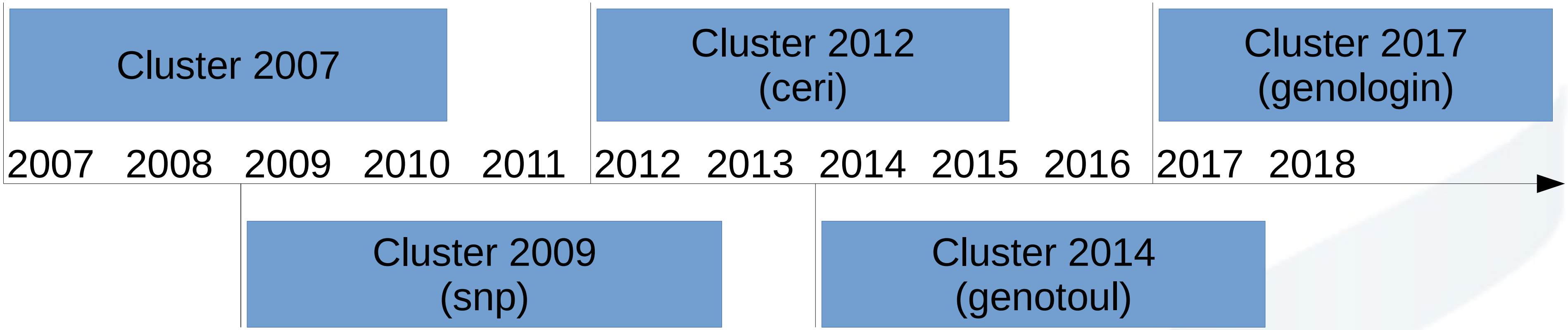
- connect to « genologin » server
- Basic command line utilization
- File System Hierarchy
- Useful tools (find, sort, cut, grep)
- Transferring & compressing files

TODAY

- How to use compute nodes cluster (submit, manage & monitor jobs)
- Objectives : Autonomy, self mastery

Context

renewal strategy



Cluster 2007

Cluster 2012
(ceri)

Cluster 2017
(genologin)

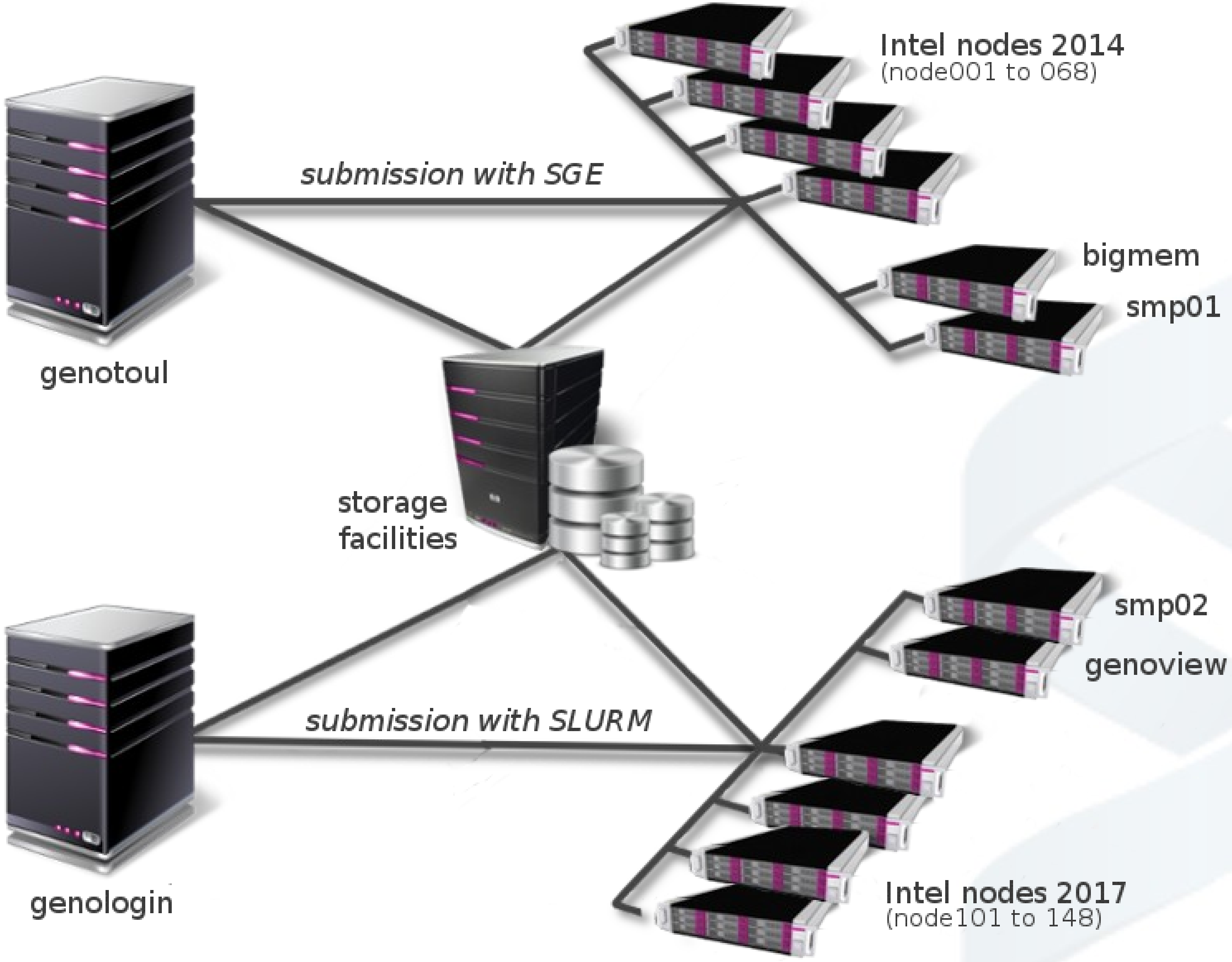
2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018

Cluster 2009
(snp)

Cluster 2014
(genotoul)

- Overlapping clusters enabling to keep the service active and to renew the machines
- But this time we have changed the job scheduler (from SGE to SLURM)
- Only SLURM at the end on 2018

Infrastructure



Infrastructure
login & compute nodes

login nodes

- 2 login nodes : genologin1&2 * (32 cores, 128 GB RAM)
- Alias : genologin.toulouse.inra.fr
- Linux based on CentOs-7 distribution
- Hundreds of users simultaneous
- Secured (ssh only)
- To serve development environments
- To test his script before data analysis
- To launch jobs on the cluster nodes
- To get data results on the /save directory

Infrastructure

login & compute nodes

Compute nodes

- 1 visualization node : genoview (32 cores, 128GB, Nvidia K40)
- 68 Ivy compute nodes : [001 à 068] * (20 cores, 256G RAM)
- 48 Broadwell compute nodes : [101 à 148] * (32 cores, 256G/512G RAM)
- genosmp02 (48 cores, 1,5T RAM)
- genosmp03 (96 cores, 3T RAM)
- Low latency & high bandwidth interconnection (56GB/s)
- Interactive mode : for beginners / for remote display
- Batch access : for intensive usage (most of jobs)
- No direct ssh access to the nodes
- Workspace exactly the same as login nodes (exception read only on /save directory)

Infrastructure

Vocabulary

Cluster / Node

- Cluster : a set of compute nodes
- Node : a computer with multi-processors and huge memory

CPU / Core / Threads

- Cpu : Central Processing Unit (socket)
- Core : multi-core in a CPU
- Threads : nb of parallel execution into a cpu/core (multi-threading)

Infrastructure

User accounts

- Access to the platform: via a command line SSH connection (putty or MobaXterm for Windows)

frontal/login servers: genologin1 & 2

alias for the connection: genologin.toulouse.inra.fr

- Example

```
ssh <login>@genologin.toulouse.inra.fr
                  => genologin1 or genologin2
```

Infrastructure

Disk spaces

- All of directories are the same between genologin servers & cluster nodes
- You don’t have to copy anything between cluster nodes & genologin
- Examples :
/home, /save, /work : user directories
/usr/local/bioinfo/src : Bioinformatics software
/bank : international genomics databanks

Infrastructure

User quotas

- **2GB** for **/home** directory (configuration files only)

- **250GB (*2)** for **/save** directory (permanent data, with replication)

- **1TB** for **/work** directory (temporary compute disk space)
Be careful : /work directory might be purged (120 days without access)

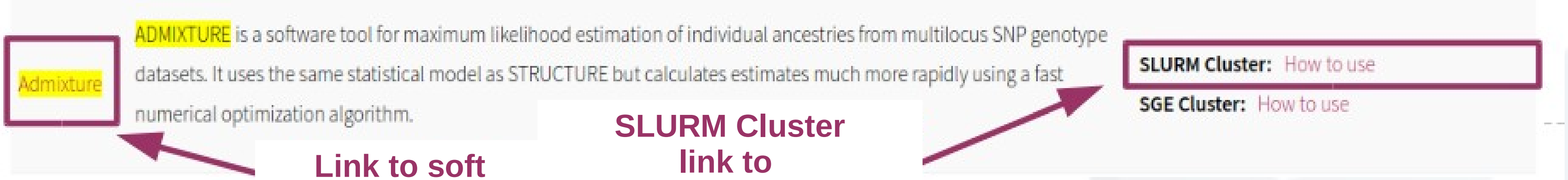
- **100,000H** annual **calculation time** (500H for private user)
You could have more time on demand (resource request)

Training day
SLURM cluster

- Context
- Infrastructure
- Software usage
- SLURM directives
- For further with SLURM
- Best practices
- Support

Software usage
Help

With Admixture on our website Software page



Admixture

ADMIXTURE is a software tool for maximum likelihood estimation of individual ancestries from multilocus SNP genotype datasets. It uses the same statistical model as STRUCTURE but calculates estimates much more rapidly using a fast numerical optimization algorithm.

SLURM Cluster: How to use

SGE Cluster: How to use

Link to soft website

SLURM Cluster link to « How_to_use » file

With Bowtie in command line

```
$ ls  /usr/local/bioinfo/src/bowtie/
```

```
bowtie-1.2.1.1  bowtie-1.2.1.1-linux-x86_64.zip  bowtie2-2.2.9  bowtie2-2.3.3.1
bowtie2-2.3.3.1-linux-x86_64.zip  example_on_cluster  How_to_use_SLURM_bowtie
```

```
$ ls  /usr/local/bioinfo/src/bowtie/example_on_cluster/
```

```
errot.txt  example  lambda_virus.1.bt2  lambda_virus.2.bt2  lambda_virus.3.bt2
lambda_virus.4.bt2  lambda_virus.rev.1.bt2  lambda_virus.rev.2.bt2  output.txt
test_v2-2.2.9.sh
```

Software usage
Command line

Installation paths

- Bioinfo -> /usr/local/bioinfo/src/
- Compilers → /tools/compilers
- Libraries → /tools/librairies
- Others system tools → /tools/others_tools
- Languages (Python, R , Java..) → /tools
- Useful scripts → /tools/bin (sarray, squota_cpu, saccount_info…). In user’s default PATH.

Software usage

Run a soft

Run a software

To run a software you need to load the corresponding module.

module load <module_name>

To run a software with others software dependencies, you need to load all required modules.

Software usage

Module command

The basic command to use is module:

- **module avail <category>** : list available software module
- **search_module <soft_name>**: display available versions for a specific application (case insensitive)
- **module load module_name** : add a module to your environment
- **module unload module_name** : unload remove a module
- **module list** : check modules already loaded
- **module purge** : remove all modules

Software usage

Other help

- **module help module_name** : find the How_to_use_SLURM_<soft_name>" file path
- **module show module_name** : show what changes a module will make to your environment
- http://vm-genobiotoul.toulouse.inra.fr/How_to_Softs/
Browse all "How_to_use_SLURM_<soft_name>" files (in your web browser)
- http://bioinfo.genotoul.fr/index.php/faq/ : Updated FAQ

Software usage Practical work

How to use Bismark_v0.19.0 ?

- Read the How to use first
- Load pre-requisite environment if needed
- Load Bismark environment
- Test Bismark help command line

How to use Python-2.7.2 ?

- Find the different versions of python installed
- Purge all of precedent modules
- Load python-2.7.2 module
- Test python help command line

Training day SLURM cluster

- Context
- Infrastructure
- Software usage
- SLURM directives
- For further with SLURM
- Best practices
- Support

SLURM
System evolution

SLURM

- Simple Linux Utility for Resource management
- Adopted by the academic community
- Supported by IT providers
- New features
- **https://slurm.schedmd.com/**

CentOS-7

- Community ENTerprise Operating System
- Supported by IBM Spectrum Scale
- Cgroups (Control Groups) compatible

Job submission

[BATCH]

- **sbatch** : submit a batch script to slurm.
- **scancel** : kill the specified job

SLURM
Interactive mode

Job submission

[INTERACTIVE]

- **srun --pty bash** : submit an interactive session with a compute node (default workq partition).
- **srun --x11 --pty bash** : submit an interactive session with X11 forwarding (default workq partition)

  For the first time, create your public key as below (onto genologin server)

  ```
  $ ssh-keygen
  $cat .ssh/id_rsa.pub >> .ssh/authorized_keys
  ```

- **runVisuSession.sh** : submit a TurboVNC / VirtualGL session with the graphical node (interq partition). Just for graphics jobs.

SLURM
Monitoring commands

Job monitoring

- **sinfo** : display nodes, partitions, reservations
- **squeue** : display jobs and state
- **sacct** : display accounting data
- **scontrol show** : get informations on jobs, nodes, partitions
- **sstat** : show status of running jobs
- **sview** : graphical user interface

SLURM

--format option

Some commands (like **sacct** and **squeue**) give the possibility to **tune output format** :

Example :

```
sacct --format=jobid%-13,user%-15,uid,jobname%-15,state%20,exitcode,Derivedexitcode,nodelist% -X –job 6969
```

```
        JobID            User     UID         JobName                State ExitCode DerivedExitCode        NodeList
------------- --------------- ------ --------------- -------------------- -------- -------------- ---------------
6969          root                 0 toto                       COMPLETED      0:0            0:0   node[101-102]
```

```
squeue --format="%10i %12u %12j %.8M %.8l %.10Q %10P %10q %10r %11v %12T %D %R" -S "T"
```

```
JOBID      USER         NAME             TIME TIME_LIM   PRIORITY PARTITION  QOS        REASON     RESERVATION STATE        NODES NODELIST(REASON)
6612       root         bash            16:09 4-00:00:00          1 workq      normal     None       (null)      RUNNING      2 node[101-102]
6542       dgorecki     TurboVNC     1-06:27:44 UNLIMITE          1 interq     normal     None       (null)      RUNNING      1 genoview
```

SLURM

Default parameters

- workq partition
- 1 thread
- 2GB RAM memory
- 100,000H annually compute time (more on demand)
- 10,000: max jobs for all users
- 2500: max jobs per user inside the queue
- 2500 : max tasks array per job

SLURM

Sample sbatch script

```
# !/bin/bash
#SBATCH --time=00:10:00 #job time limit
#SBATCH -J testjob          #job name
#SBATCH -o output.out       #output file name
#SBATCH -e error.out        #error file name
#SBATCH --mem=8G            #memory reservation
#SBATCH --cpus-per-task=4         #ncpu on the same node
#SBATCH --mail-type=BEGIN,END,FAIL (email address is LDAP account's)
#Purge any previous modules
module purge
#Load the application
module load bioinfo/ncbi-blast-2.2.29+
# My command lines I want to run on the cluster
blastn ...
```

Practical work 1

Simple execution on interactive mode

- Log in to genologin server
- Go to your “work” directory
- Create a sub-directory “cluster”
- Go to the “cluster” directory
- Download the file http://genoweb.toulouse.inra.fr/~formation/cluster/data/contigs.fasta.gz
- **Connect to compute node in interactive mode**
- Un-compress contigs.fasta.gz file
- Display the first 10 lines
- Which is the format file ?
- Which is the kind of data ?

Practical work 2

blastx submission on interactive mode

- **Stay connected to the compute node in interactive mode**
- Load the module: bioinfo/ncbi-blast-2.6.0+
- Launch a blastx against “ensembl_danio_rerio” (-evalue 10e-10)
  Your query is genomic, your database is proteic so you need a blastx program.
- Open a new terminal and check your job waiting or running with SLURM
- On wich node are you running ?
- Kill you job

Practical work 3

blastn submission on batch mode

- Go back to genologin server
- Use a text editor to create the command file “cmd.txt”
- Type inside the same command lines as Practical work 2

  Use “**blastn**” instead of “blastx”
  The first line must start with **#!/bin/sh**
- Lanch the command file with SLURM on batch mode
- Check the execution on SLURM
- When it’s over, check and look at the output files
- Is the job finished correctly ?

Training day SLURM cluster

- Context
- Infrastructure
- Software usage
- SLURM directives
- For further with SLURM
- Best practices
- Support

SLURM

Directives (1/2)

-p workq	#partition name
--time=00:10:00	#job time limit
-J testjob	#job name
-o output.out	#output file
-e error.out	#error file name
--mem=8G or --mem-per-cpu	#memory size

SLURM

Directives (2/2)

--cpus-per-task=4	#ncpu on the same node
--mail-type=[events]	#event notification
--mail-user=[address]	#default LDAP account's
--export=[ALL|NONE|variables]	#copy environment
--workdir=[dir_name]	#working directory
--wrap=”command”	#With sbatch to submit directly one command”

SLURM

Partitions

- Each job is submitted to a specific partition (the default one is the workq).
- Each partition has a different priority considering the maximum time of execution allowed.

Partitions (queues)	Access	Priority	Max time	Max threads
workq	everyone	100	4 days (96h)	3072
unlimitq	everyone	1	180 days	500
interq (runVisusession.sh)	on demand		1 day (24h)	32
smpq	on demand		180 days	96
wflowq	specific software	200	180 days	3072

SLURM
Ressources

- It depends on your genotoul linux group : contributors / INRA or REGION / others.
- There are limitations on user + group of users
- It is the same thing for the RAM memory (1 thread <=> 6GB RAM)

Partition / max threads	workq (group)	workq (user)	unlimitq (all)	unlimitq (user)
contributors	6218	1448	500	128
Inra or region	4663	1086	500	96
others	1554	362	500	32

SLURM

Job dependencies

sbatch -d | --dependency=<dependency_list>

Defer the start of this job until the specified dependencies have been satisfied completed.

<dependency_list> is on the form <type :jobID[:jobID][,type :jobID[:jobID]]>

Example :

sbatch --dependency=afterok:6265 HELLO.job

Type	Correspondance
after	this job can begin execution after the specified jobs have begun execution
afterany	this job can begin execution after the specified jobs have terminated
afterok	This job can begin execution after the specified jobs have successfully executed (ran to completion with an exit code of zero)
afternotok	This job can begin execution after the specified jobs have terminated in some failed state (non-zero exit code, node failure, timed out, etc)

SLURM
Job arrays

sbatch -a | array=<indexes>

Submit a job array, multiple jobs to be executed with identical parameters.

Multiple valued may be specified using a comma separated list and/or a range of values with a « - » separator.

Example :

--array=1-10

--array=0,6,16-32

--array=0-16:4 #a step of 4

--array=1-10%2 #a maximum of 2 simultaneously running task

Variable	Correspondance
SLURM_ARRAY_TASK_ID	Job array ID (index) number
SLURM_ARRAY_JOB_ID	Job array's master job ID number
SLURM_ARRAY_TASK_MAX	Job array's maximum ID (index) number
SLURM_ARRAY_TASK_MIN	Job array's minimum ID (index) number
SLURM_ARRAY_TASK_COUNT	total number of tasks in a job array

SLURM

Useful scripts

These useful scripts are already in your default path or /tools/bin

- saccount_info <login>: account expiration date and last password change date, primary and secondary Linux group, status of your Linux primary group in Slurm (contributors, inraregion or others), groups' members, some Slurm limitations of your account : cpu and memory limit, CPU Time ...
- **sq_long** or **sq_debug**: squeue long format
- **sa_debug**: sacct long format
- **squota_cpu**: see your CPU time limit.
- **seff <jobid>**: check the efficiency of a COMPLETED job (cpu, memory)
- **sarray <file.txt**> : each line in file.txt will be run in parallel

Practical work

Array of jobs (1/2)

- Split the fasta file in 10 fasta files into a new directory called out_split :

module load bioinfo/exonerate-2.2.0; fastasplit -f contigs.fasta -c 10 -o out_split

- Check the number of files into **out_split** dir.
- Check if all the sum of all splitted sequences files matches with the number of sequences in “**contig.fasta**” file
- Create a command file with one blast command per fasta file.
  blast each fasta file against **ensembl_danio_rerio** genomic bank

  See the FAQ : http://bioinfo.genotoul.fr/index.php/faq/bioinfo_tips_faq/
  -> How to generate an sarray command file with bash for single fastq file

Practical work

Array of jobs (2/2)

- Test the first line to check until there is no syntax error
- **Kill** the process using « **ctrl+c** »
- Launch the **job array** on SLURM ; check how many jobs are running ?
- After execution check trace files « slurm-<jobid>_*.out
- Use “seff” command to check how many ressources are used
- **Concat** all blast results in one file

Practical work
multithread job

- Launch the **blastx** command line with SLURM (batch mode) with **8 threads** on the same node
  Use **all the contigs** (contigs.fasta) file against **ensembl_danio_rerio** genomic bank
  Be careful to reserve **8 cpu per task** (SLURM directive)
- Check the execution on the cluster in details
- **Re-use the jobarray script** to lanch it with **8 threads** instead of one
  Be careful to reserve 8 cpu per task (SLURM directive)
- Compare the different ways to lanch a blast ; which is the better ? (fastest)

Training day
SLURM cluster

- Context
- Infrastructure
- Software usage
- SLURM directives
- For further with SLURM
- Best practices
- Support

Best pratices
Security

One user = one account

You are responsible of the damage caused by your login.

Default permissions directories

- home: drwxr-x—x : **R**ead, **W**rite, e**X**ecution for the owner, **R**ead and e**X**ecution for your group members, e**X**ecution for all.

- save and work: drwxr-x--- : **R**ead, **W**rite, e**X**ecution for the owner, **R**ead and **E**xecution for your group members, no permissions for all.

To change permissions: **chmod** command

Best pratices

Resources

Cluster is a shared resource, so ... think about the others

- try to adapt requested resources to your needs.

- DO NOT run treatments on frontal servers:

Why ?

- overloading frontal servers slow down everyone.
- overloading frontal servers can crash frontal servers and block everyone.
- more time for the administrators to answer support requests.

Check your process on frontal servers : **$ pstree -u <login>**

Any treatment launched on the servers "genologin" will be immediately killed by the system administrators

Best pratices

memory

Try to adjust requested memory reservation to your needs.

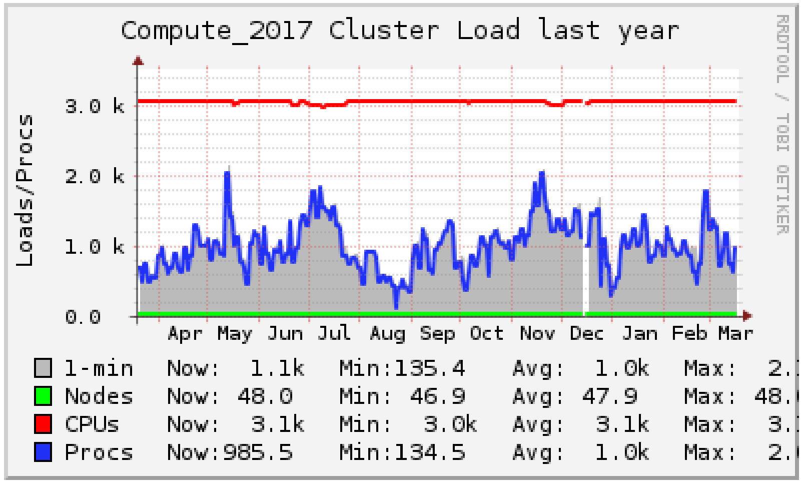
- If you overbook the memory reservation then you will stay more time in queue
- If you overbook the memory reservation then the memory will not be available for others
- To know how the job needs memory, you may use “seff” command on a completed job

Support

Cluster monitoring

Ganglia → https://monitoring.bioinfo.genotoul.fr

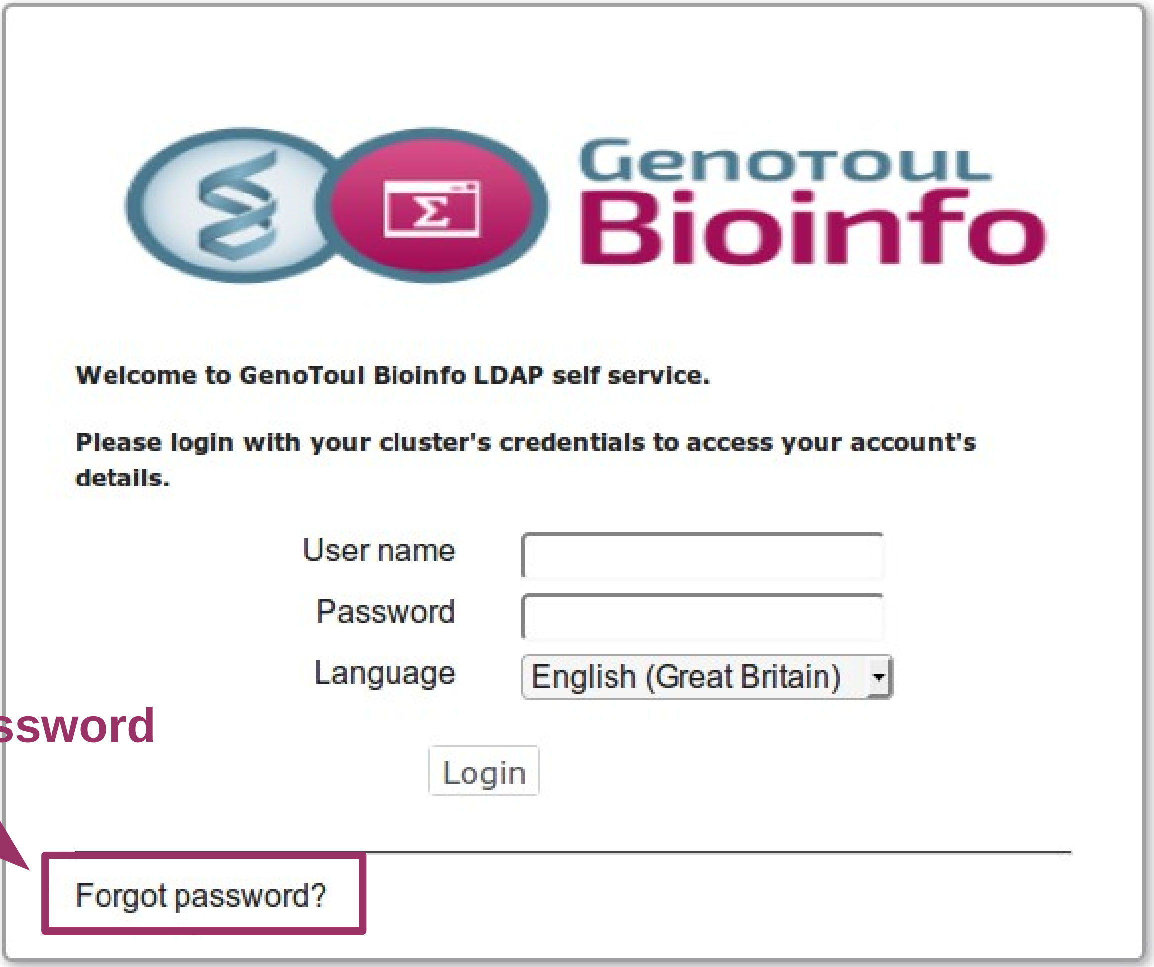
(or our website : Resources/Monitoring)



Support

Account information and password change

Self Service → https://selfservice.bioinfo.genotoul.fr



Welcome to GenoToul Bioinfo LDAP self service.

Please login with your cluster's credentials to access your account's details.

Change your password (every year)

Support

- **Bioinfo genotoul website :**

http://bioinfo.genotoul.fr/

- **Bioinfo Genotoul Chart**

http://bioinfo.genotoul.fr/wp-content/uploads/ChartPFBioinfoGenoToul.pdf

- **FAQ**

http://bioinfo.genotoul.fr/index.php/faq/

- **Support**

Mail: support.bioinfo.genotoul@inrae.fr

Fill form (best for us): http://bioinfo.genotoul.fr/index.php/ask-for/support/

End of Presentation

Thanks for your attention !