Cluster Initiation

Cycle de formation 2015

16 / 06 / 2015

Alexandre Cormier
Camille Vacquié

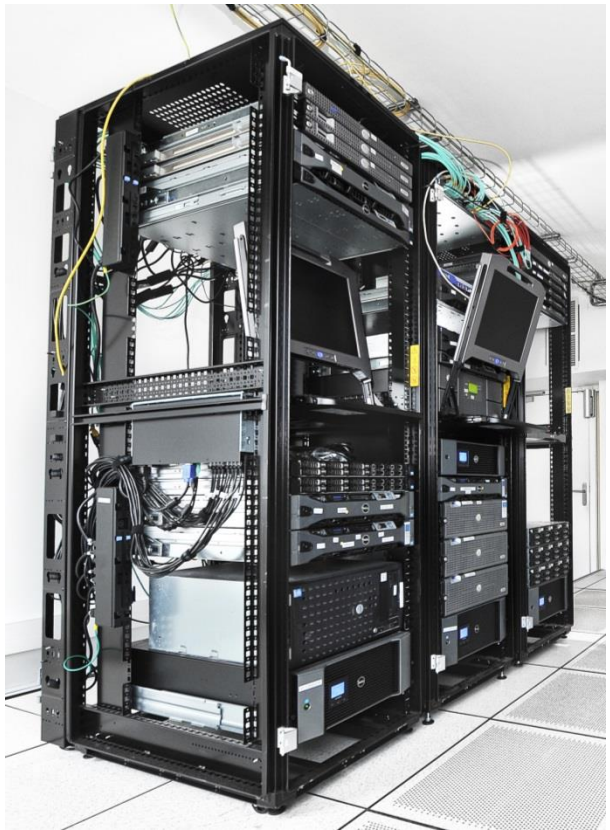
UPMC SORBONNE UNIVERSITÉS

cnrs

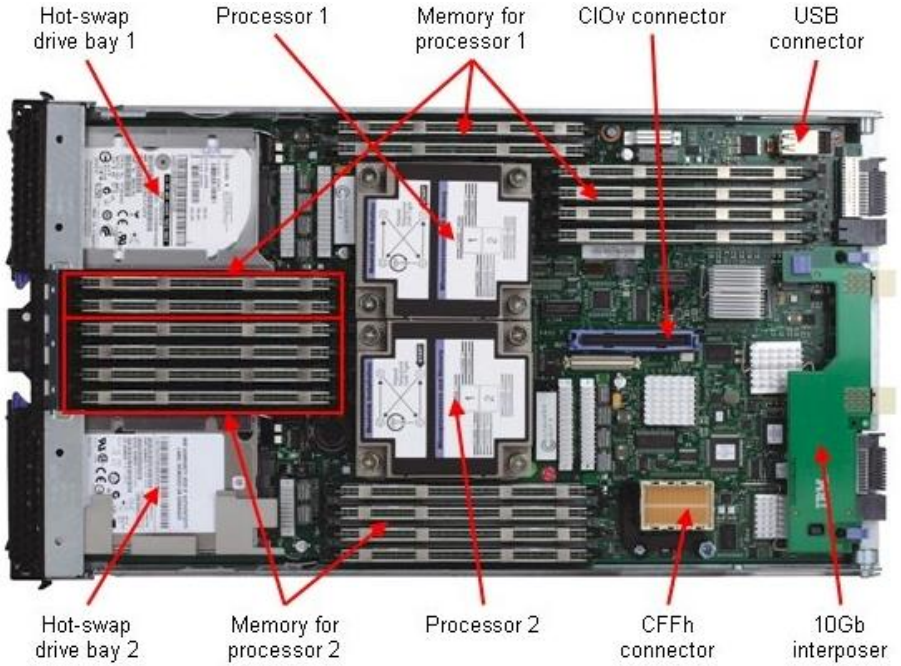
IFB
INSTITUT FRANÇAIS DE BIOINFORMATIQUE

Principle

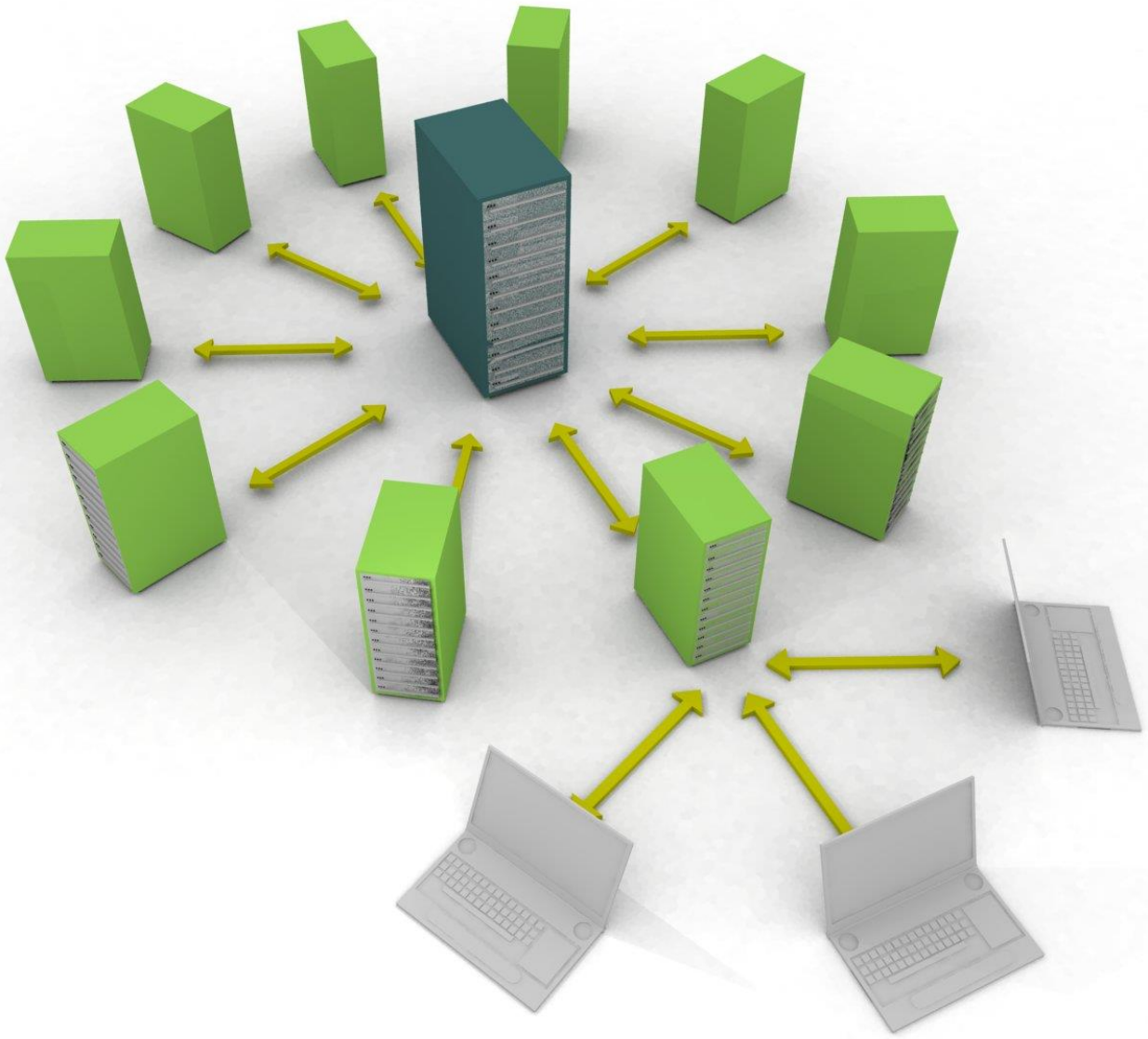
- Aggregation of computers / machines
  - Machine = node
- Distributed computing + shared access
- Transparent managment for users
- Community system → rules!



Hardware


Hot-swap drive bay 1
Processor 1
Memory for processor 1
CIOv connector
USB connector
Hot-swap drive bay 2
Memory for processor 2
Processor 2
CFFh connector
10Gb interposer

Distributed computing



Distributed computing

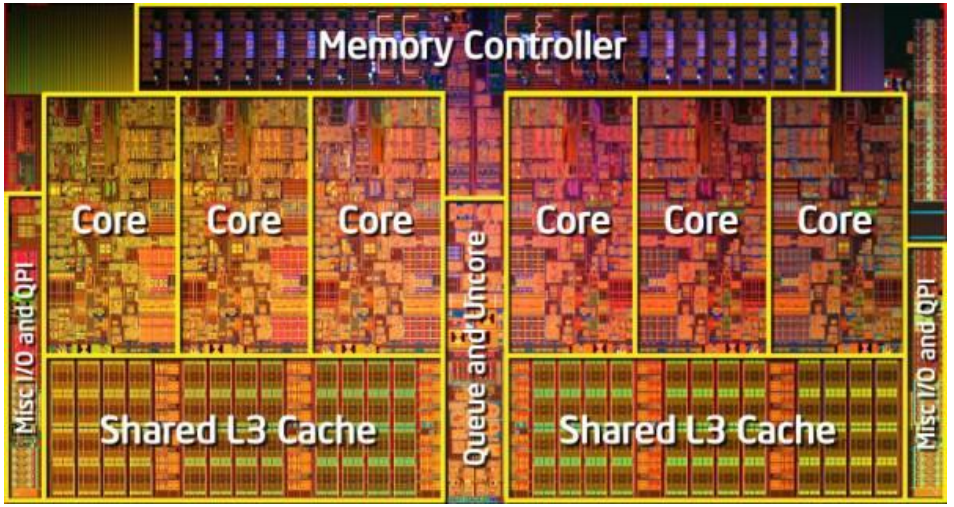
- Distribution
  - Make a job as atomic as possible
  - Simple and robust
  - Linear gain

- Generate independent tasks
  - Split the data
  - Change parameters

Multithreading

- Thread
  - Tasks running on the same machine but on several CPU or core
  - Shared memory
  - Nonlinear gain

  - Ex: Tophat, CLC, Blast,…

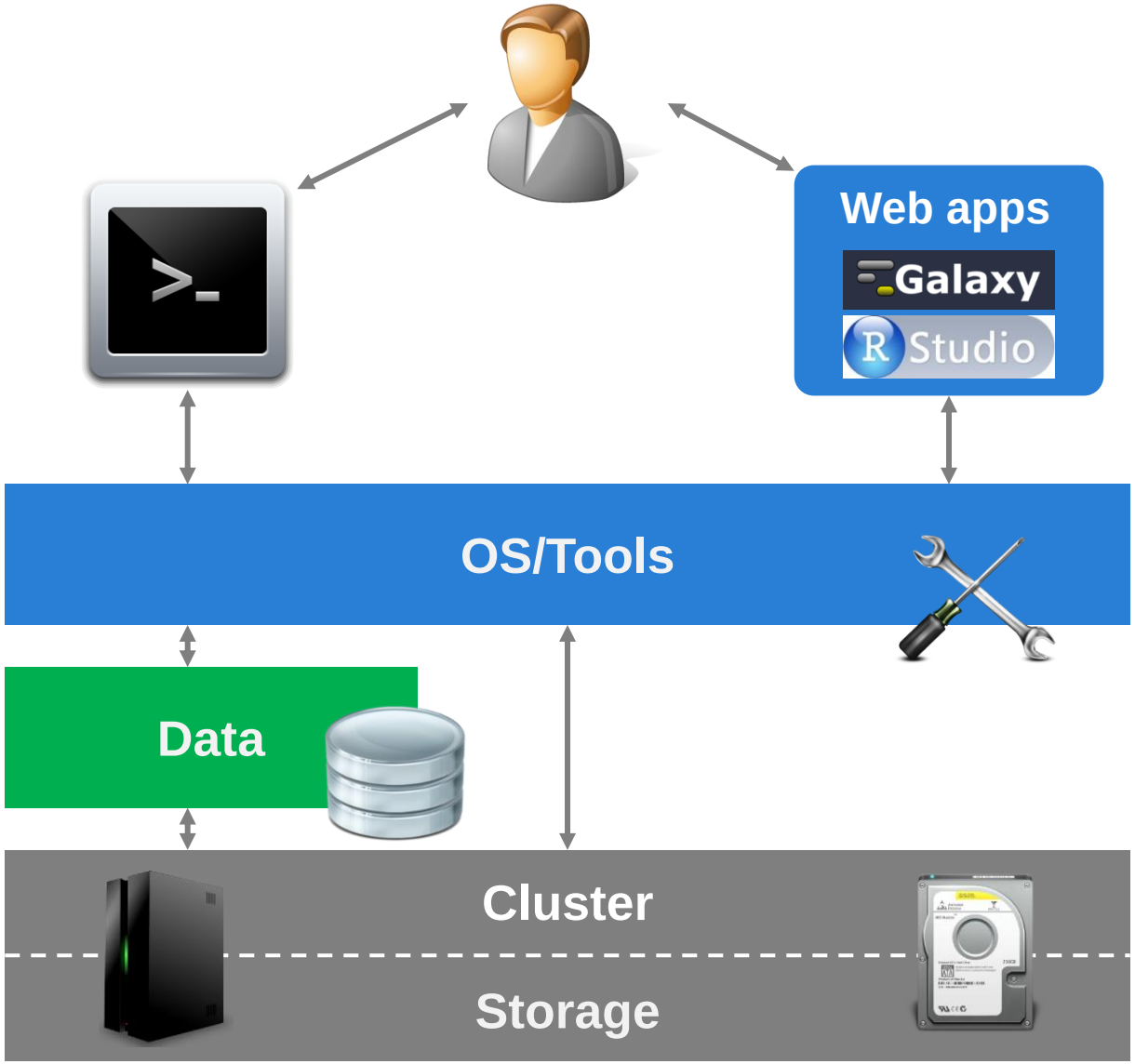


Parallel computing

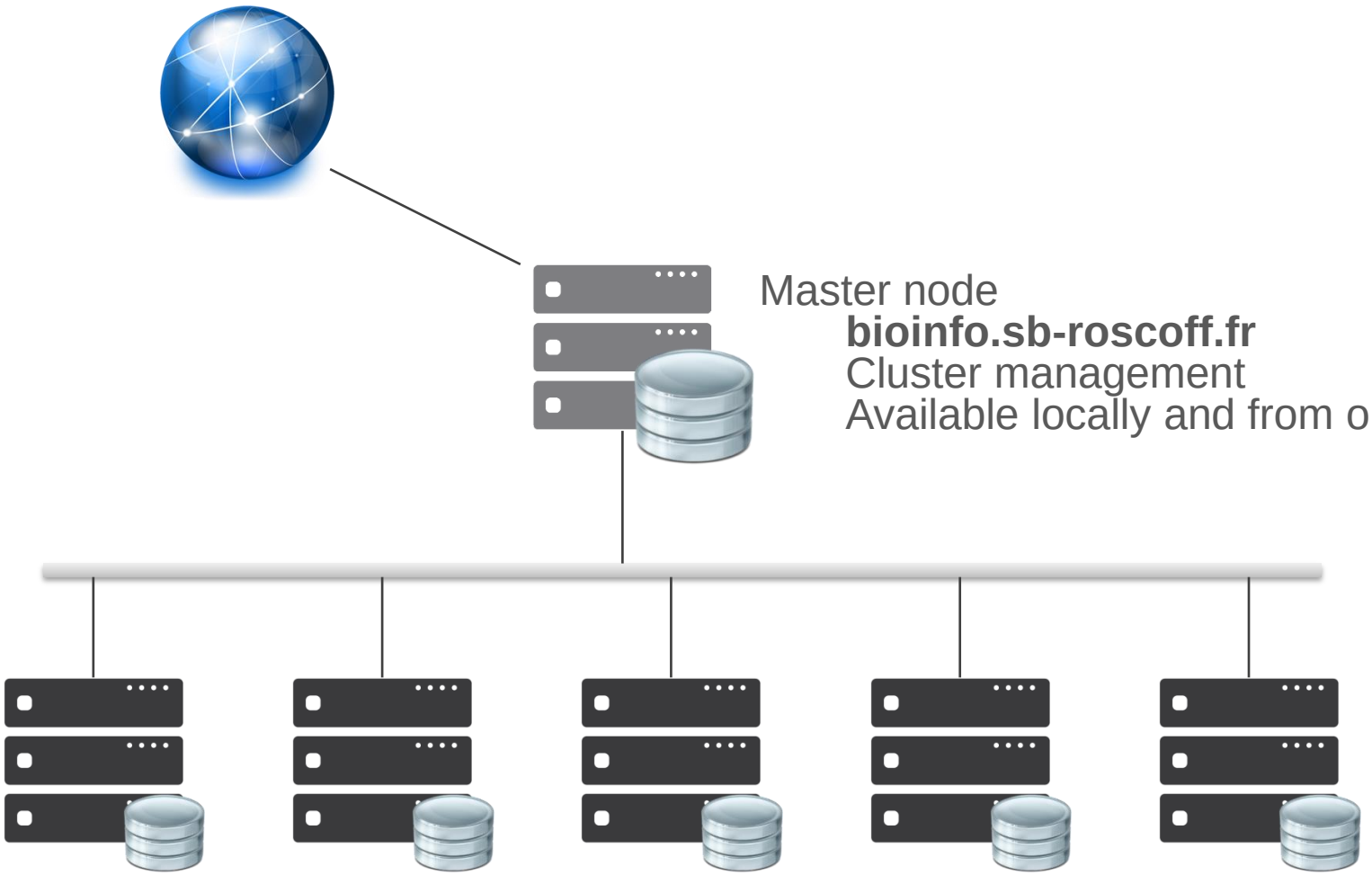
- MPI (Message Parsing Interface)
  - Tasks are running on different machines
  - Communication between tasks over the network
  - Variable gain. Nonlinear in general

  - Ex: PhyML-MPI, ClustalW-MPI

Environment



Cluster


Master node
bioinfo.sb-roscoff.fr
Cluster management
Available locally and from outside

Cluster

- 16 nodes 8 core 2.4 Ghz / 32 Go RAM
  - n60 - n75
  - Group @blade

- 4 nodes 48 core 2.2 Ghz / 256 Go RAM
  - n76 - n79
  - Group @bignode

- 16 nodes 16 core - 32 threads / 128 Go RAM
  - n80 – n95
  - Group @intel22

- 1 node 40 core / 1To RAM
  - n99

Data

- Personal data

- Shared data:
  - By team / group
  - By UMR
  - For a community
  - Public data

- Databank
  - Genbank, Uniprot, InterPro banks, etc.
  - Format : Blast, FASTA, EMBL, etc.
  - Private & Public

Workspace

Projet

→ per person
→ by team
→ by subject

Home

→ only for connexion (Environment variable)

DB

→ Databank (Blast, Genbank, Interpro…)

Galaxy

→ import
→ export

Workspace

Projet

Partial backup

Home

Partial backup

DB

No backup

Galaxy

No backup

Workspace

Scratch

20 To

Space for all the primary analysis - generated huge amount of temporary/useless files

- **Mutualised storage** between all users
- Data are **not backed up**
- All files older than **30 days** are automatically **deleted**

Environment

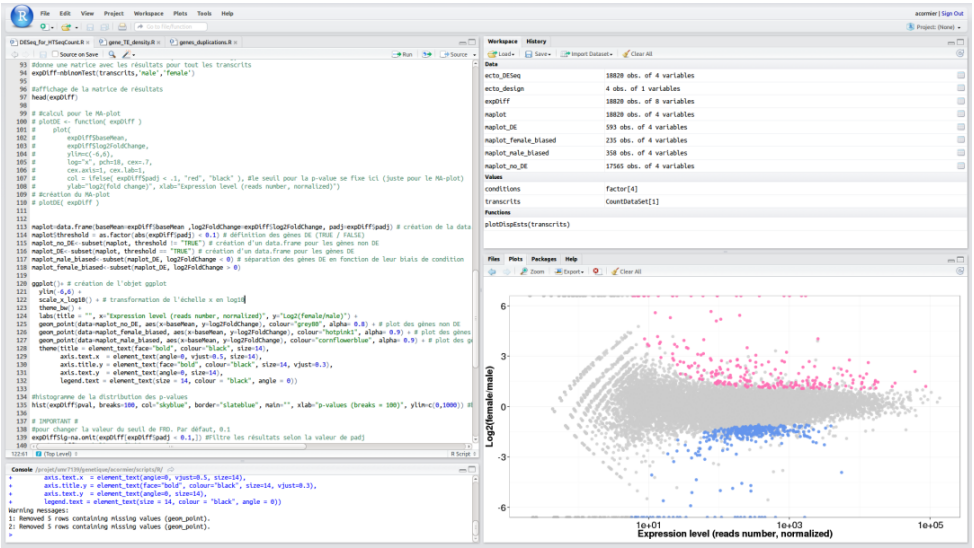
- Command-line
  - Knowledge in Unix / Bash
  - Integrated in computer resources and storage

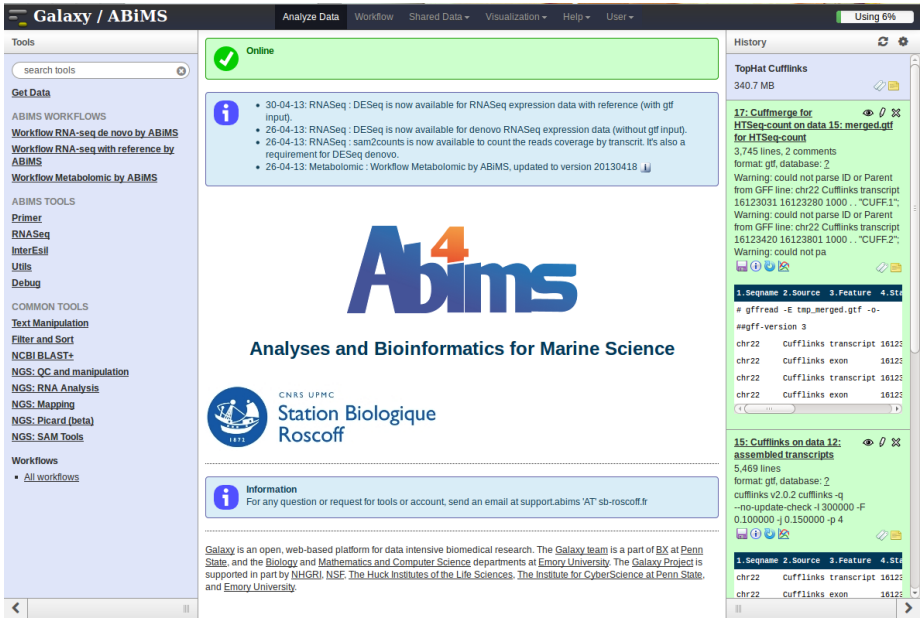


```
acormier@n0:/tmp
drwx------   2 root            root           16384 avr 15  2009 lost+found
srwxr-xr-x   1 caron           sib                0 oct 30  2012 mapping-caron
srwxr-xr-x   1 corre           sib                0 avr 18 14:08 mapping-corre
srwxr-xr-x   1 ewcorre         lbm                0 jun 25 15:37 mapping-ewcorre
srwxr-xr-x   1 jkervellec      sib                0 fév 22 13:52 mapping-jkervellec
srwxr-xr-x   1 jmaroumougom    sib                0 nov  3  2011 mapping-jmaroumougom
srwxr-xr-x   1 ndebs           lbm                0 jun 28 15:11 mapping-ndebs
srwxr-xr-x   1 root            root               0 avr 17  2009 mapping-root
srwxr-xr-x   1             5000 root               0 avr 28  2009 mapping-toto
drwx------   2 llegrand        inra            4096 jui  5 10:28 mozilla-media-cache
drwxr-xr-x 258 hfcollector     application    20480 jui  6 04:48 ODV_hfcollector
drwx------   3 nhenry          eppo            4096 jui  2 13:43 openmpi-sessions-nhenry@n0.sb-roscoff.fr_0
drwx------   2 acormier        genetique       4096 mai  3 20:57 orbit-acormier
drwx------   2 cock            genetique       4096 avr  7 13:18 orbit-cock
drwx------   2 corre           sib             4096 jun 24 09:58 orbit-corre
drwx------   2 ewcorre         lbm             4096 jui  3 13:50 orbit-ewcorre
drwx------   2 gdm             gdm             4096 avr  5 18:30 orbit-gdm
drwx------   2 hfcollector     application     4096 jui  4 10:16 orbit-hfcollector
drwx------   2 lecorguille     sib             4096 jun 20 02:48 orbit-lecorguille
drwx------   2 llegrand        inra            4096 jui  5 10:35 orbit-llegrand
drwx------   2 mhoebeke        sib             4096 jun 25 13:30 orbit-mhoebeke
drwx------   2 ndebs           lbm             4096 jun 28 16:17 orbit-ndebs
drwx------   2 stage02         stage           4096 mai 13 10:40 orbit-stage02
drwx------   2 wcarre          sib             4096 jun 20 09:53 orbit-wcarre
srwxr-xr-x   1 ewcorre         lbm                0 nov 10  2011 OSL_PIPE_6108_SingleOfficeIPC_eebd8121e860c31ca9a23ed86a44ce
drwxr-xr-x   4 root            root            4096 sep 30  2009 perl5
drwxr-xr-x   2 acormier        genetique       4096 jun 25 15:58 perl_formation
drwxr-xr-x   2 mhoebeke        sib             4096 jun 25 11:45 phyloclusters
srwxr-xr-x   1 root            root               0 avr 11  2012 sfcbLocalSocket
drwxr-xr-x   5 root            root            4096 fév 20  2010 sge
-r--------   1 root            root            3066 mai 23  2011 shadow
-rw-r--r--   1 root            root           10978 jui  5 23:00 stat_sge.txt
drwxr-xr-x   3 root            root            4096 jun 26  2012 toto
[acormier@n0 tmp]$
```

Web applications

- Web interface
  - Galaxy
  - R-studio: dedicated to R





http://r.sb-roscoff.fr/

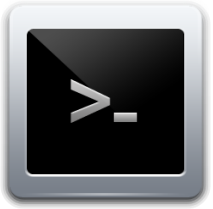
http://galaxy.sb-roscoff.fr/

Connexion and storage

Prerequisite

- Account
  → http://abims.sb-roscoff.fr/account
  → support.abims@sb-roscoff.fr
- Email
- X11 terminal
  - Windows: Putty & Xming
  - Mac OS & Linux: integrated
- Text editor
  - Vim, nano, gedit, emacs…
- SFTP client





Connection

```
$ ssh -Y acormier@bioinfo.sb-roscoff.fr # -Y → for graphic flux redirection
```

Connection


```
$ ssh -Y acormier@bioinfo.sb-roscoff.fr # -Y → for graphic flux redirection

 Last login: Thu May 30 17:17:46 2013 from 192.168.4.162

 Plateforme ABIMS (Analysis and Bioinformatics for Marine Science)

 Le cluster de calcul est désormais en production

 ************************************************************************
  IMPORTANT:     Le serveur N0 de doit pas executer de traitement
                Utiliser systematiquement les nodes de calcul SVP
 ************************************************************************

 Merci de signaler a l'alias support.abims@sb-roscoff.fr d'eventuels problemes

 ************************************************************************
  Important : Travaillez imperativement sur /projet
                - performances
                - non dependances du /home (brazil)
                - volumetrie

  Voir : http://abims.sb-roscoff.fr/faq
 ************************************************************************
```

Connection

When I'm connecting, I arrive in my:

Home

```
$ pwd #print working directory

  /home/umr8227/ga/acormier
```

Not for storage / computing

Connection

I have to go in:

Projet

To store raw data, final results and scripts

```
$ cdprojet #alias for fast moving in my project directory
$ pwd

  /projet/umr8227/ga/acormier
```

- Structuration:
  - → by team: /projet/umr8227/ga/acormier
  - → by subject: /projet/abims/ectocarpus
- Shared between all nodes
- Available from outside

Connection

I have to work in:

Scratch

For all analysis

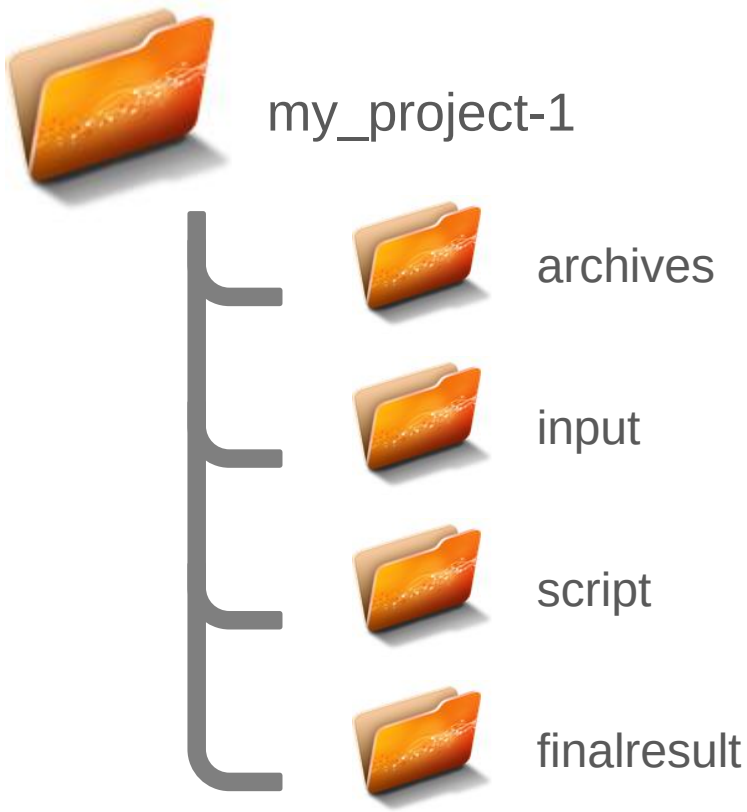
```
$ cdscratch
$ pwd

  /projet/umr8227/ga/acormier
```

- Structuration:
  → by user

- Shared between all nodes
- Available from outside

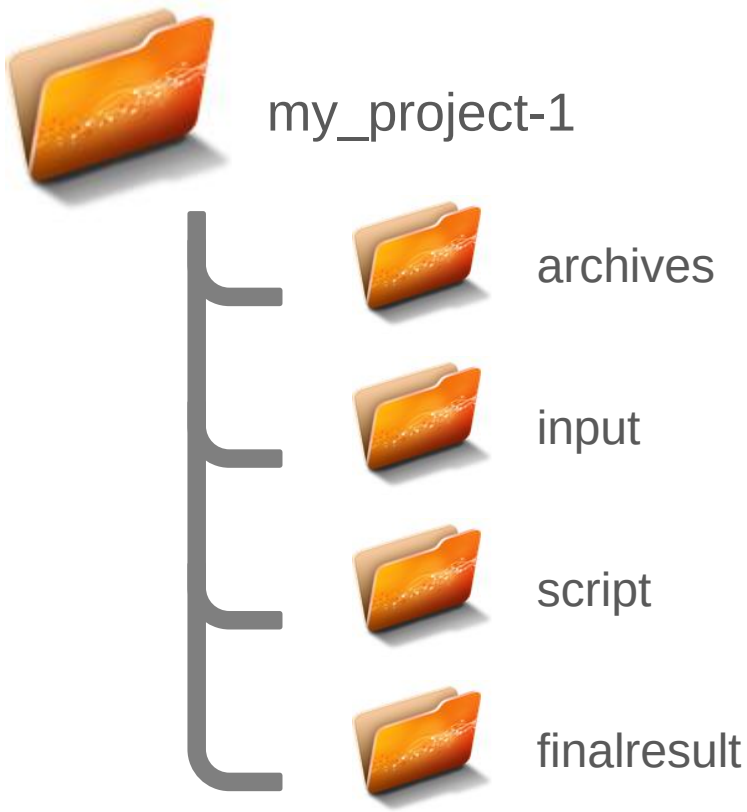
Structuration of my project directory

Each project needs to have particular folders:



Structuration of my project directory

Each project needs to have particular folders:



Backup system: by inclusion.

Only these folder are saved:

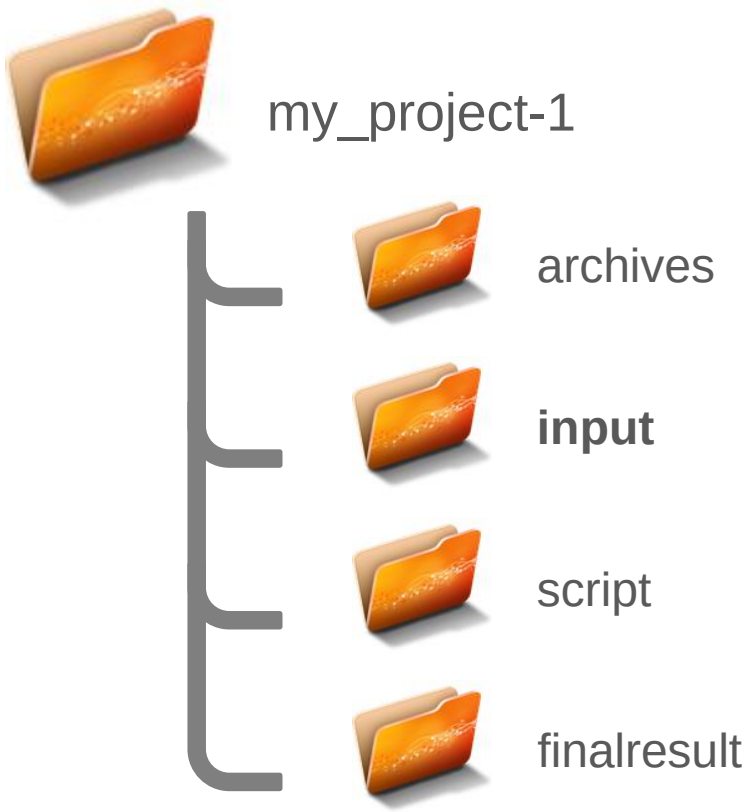
- **finalresult**
- **script**
- **archives**

Pay attention to typo! Case sensitive

Structuration of my project directory

Original data sources.
Rarely used, only for archiving.
E.g. Raw data from sequencing (Sanger, DNA-seq, RNA-seq, etc)

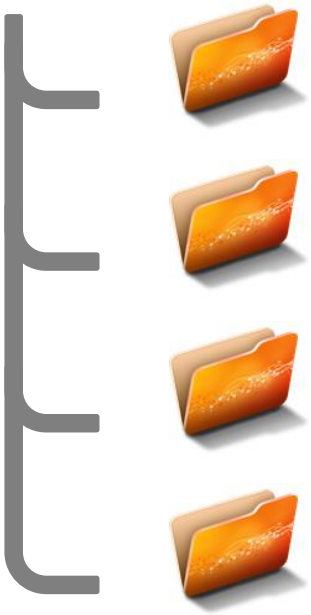
Structuration of my project directory



Data used as input file for analysis
E.g. Cleanded data from sequencing, fasta files, etc

Structuration of my project directory

my_project-1



archives

input

script

finalresult

Qsub files
R scripts and workspace
Python/perl/C/Ruby/… scripts

Structuration of my project directory

my_project-1

archives

input

script

finalresult

Results of analysis that need to be conserved.

Structuration of my scratch directory

- No particular structuration

- Don’t forget this:
  - **All files older than 30 days are automatically deleted** (based on the last modification date)

Storage limitation

Regularly, check the volume of my project to prevent saturation. The storage is not by user, but by team…

```
$ df -h . #report filesystem disk space usage
  Sys. de fich.          Tail. Occ. Disp. %Occ. Monté sur
  cfs1:/projet/umr8227/ga  1,4T  651G  658G  50% /projet/umr8227/ga

$ cdscratch
$ du -sh * #size of each file/folder -> who is the biggest?

  68G    assembly
  341G   pagit
  3,8G   remapping
  12K     cache_tmp
  17M     chr_similarity
  1008M   galaxy_dataset
  669M    metrics
  2.1M    Tes

$ du -sh assembly/*

  11G     assembly/transcriptome_V1
  9.8G    assembly/transcriptome_V2
  48G     assembly/trinity
```

Management of my project directory

Compress your data!

```
$ ll -h

  -rw-rw-r--+ 1 acormier    ga  26G mars  8 08:16 140220_SND393_B_L006_GPO-12_R1.fastq
  -rw-rw-r--+ 1 acormier    ga  26G mars  8 08:16 140220_SND393_B_L006_GPO-12_R2.fastq

$ gzip 140220_SND393_B_L006_GPO-12_R1.fastq
$ gzip 140220_SND393_B_L006_GPO-12_R2.fastq

$ ll -h

  -rw-rw-r--+ 1 acormier    ga 7,7G mars  7 12:25 140220_SND393_B_L006_GPO-12_R1.fastq.gz
  -rw-rw-r--+ 1 acormier    ga 7,9G mars  7 12:29 140220_SND393_B_L006_GPO-12_R2.fastq.gz
```

Some softwares are capable to use directly compressed data (TopHat2, Trimmomatic,…)

Sun Grid Engine

How to use the cluster?

Tools

- Applications (x700)
  - /usr/local/genome2/
    - Localisation of all software available on the cluster
    - Soon: a list of all tools
  - /usr/local/genome2/script/
    - Scripts developed by people of the SBR
    - Just send an email to support.abims@sb-roscoff.fr if you want to share your scripts

- The software components are shared:
  - One installation in a unique place
  - Request to add or update: support.abims@sb-roscoff.fr

Genomes databank

- /db/
  - Public databank:
    - NCBI
    - GenBank
    - UniProt
    - InterPro
    - Etc
  - SBR databank
    - Start with the prefix “sbr_”
    - Description of these databank is currently in progress

Job management system

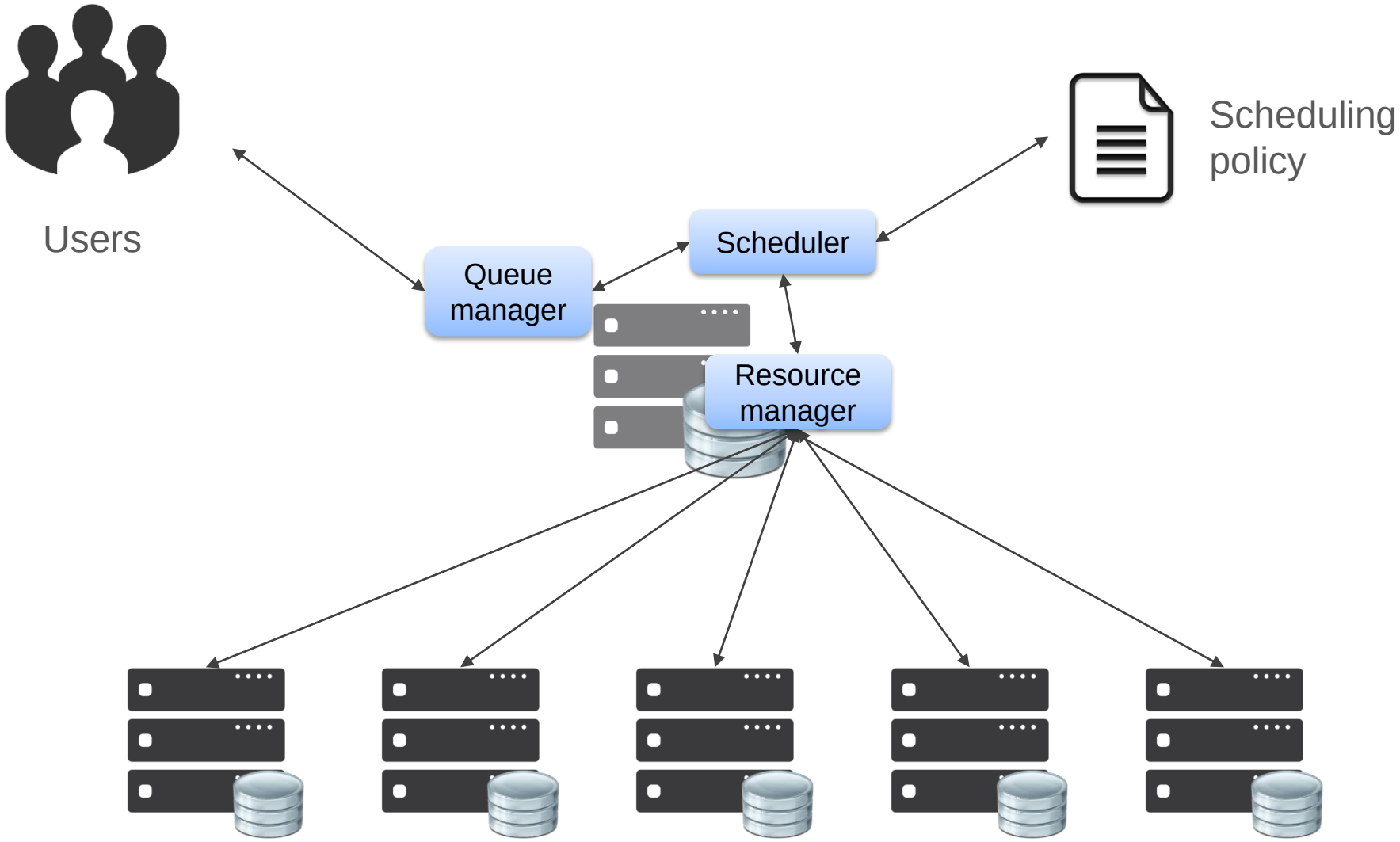
- Sun Grid Engine (SGE)
  - Scheduler in charge of the jobs management
  - User interface for submitting and controlling jobs

- Task scheduling
  - Resources allocation
  - Nodes load
  - Priority
- Management policy and resource sharing
  - CPU / Memory
  - Execution time
- Reporting and errors
  - History
  - Usage statistics

Jobs

- Job == user application
- There are several types of jobs:
  - Batch (script)
  - Interactive
- Serial vs parallel
  - Serial: only need 1 processor
  - Parallel: require more than 1 processor

Job management system



Concepts

- Slots
  - Number of jobs allowed on one node
- Job
  - Task unit
- Queue
  - Type of resources (node groupe, execution time…)
- Priority
  - **Fair Share :** calculated on 1 week → sliding window

Concepts

- Queues
  - **short.q** : priority +++ → 12h → 50% of resources
  - **long.q** : priority ++ → 10j → 50% of resources
  - **infinite.q** : priority + → infini → 25% of resources
  - **bigmem.q** : dedicated to jobs that use high quantities of memory
  - **qlogin.q** : 48h – interactive job

Max load: 1,25

→By default, no queue!

Nodes groups

- @intel22
  - High memory
  - High parallelisation
- @bignode
  - High memory
  - Very high parallelisation
- @blade
  - Low memory
  - Low parallelisation

Queues

```
$ qstat -g c #displays the available queues

CLUSTER QUEUE                   CQLOAD   USED    RES  AVAIL  TOTAL aoACDS   cdsuE
--------------------------------------------------------------------------------
short.q                           0.32    140      0    184    324      0       0
long.q                            0.32     93      0    231    324      0       0
infinite.q                        0.32      0      0     66     66      0       0
bigmem.q                          0.60     20      0     20     40      0       0
qlogin.q                          0.30      7      0     23     30      0       0
clc.q                             0.28     10      0     38     48      0       0
formation.q                       0.61      3      0     57     60      0       0
galaxy.q                          0.72      0      0    140    140      0       0
galaxy1.q                         0.33      0      0     72     72      0       0
```

The queues freely available:

- short.q
- long.q
- infinite.q
- qlogin.q → for interactive jobs

On request:

- bigmem.q : for jobs that require a lot of RAM
- clc.q : for CLC Assembly Cell

Monitoring

```
$ qhost #liste of all nodes

HOSTNAME                ARCH         NCPU  LOAD  MEMTOT  MEMUSE  SWAPTO  SWAPUS
----------------------------------------------------------------------------------
n0                      lx24-amd64      8  0.10    7.8G  841.7M    4.0G   81.6M
n60                     lx24-amd64      8  0.03   31.5G    2.3G    1.0G  656.0K
n61                     lx24-amd64      8  0.03   31.5G  350.3M    1.0G  180.0K
n62                     lx24-amd64      8  1.32   31.5G  208.5M    1.0G   80.5M
n63                     lx24-amd64      8  0.03   31.5G    1.8G    1.0G   72.2M
n64                     lx24-amd64      8  1.00   31.5G  335.0M    1.0G   82.4M
n76                     lx24-amd64     48 13.59  252.0G   22.4G    2.0G   28.5M
n77                     lx24-amd64     48 11.12  252.0G   21.3G    2.0G  240.0K
n78                     lx24-amd64     48  5.02  252.0G   22.4G    2.0G   58.8M
n79                     lx24-amd64     48 37.07  252.0G   24.6G    2.0G     0.0
n80                     lx24-amd64     32 22.14  126.0G    2.9G 1024.0M   11.0M
n81                     lx24-amd64     32 32.02  126.0G    2.8G 1024.0M     0.0
n82                     lx24-amd64     32 32.03  126.0G    2.8G 1024.0M     0.0
n83                     lx24-amd64     32 32.02  126.0G    2.8G 1024.0M     0.0
n84                     lx24-amd64     32 32.02  126.0G    3.1G 1024.0M     0.0
n99                     lx24-amd64     40 24.00 1009.7G  238.6G    4.0G  107.6M
```

Allows to check load level on each node:

- Load level on the cluster

Monitoring

- qhost: available nodes and some indicators
- qhost –j: list of jobs on each node
- qhost –q: list of queues/slots on each node
- man qhost: help

```
$ qstat #shows all jobs

  job-ID  prior   name       user         state submit/start at     queue       slots ja-task-ID
  ------------------------------------------------------------------------------------------------
  1236477 0.50000 QLOGIN     wcarre       r     04/30/2012 15:10:26 long.q@n77      1
  1236479 0.50000 QLOGIN     wcarre       r     04/30/2012 15:11:12 long.q@n77      1
  1268627 1.00000 QLOGIN     balzano      r     05/03/2012 09:39:24 long.q@n74      1
  1302170 0.06667 alpha0.sh  mrescan      r     05/06/2012 12:05:48 long.q@n64      1
  1302173 0.06667 alpha3.sh  mrescan      r     05/06/2012 12:06:18 long.q@n72      1
  1302174 0.06667 alpha4.sh  mrescan      r     05/06/2012 12:06:33 long.q@n70      1
  1302175 0.06667 alpha5.sh  mrescan      r     05/06/2012 12:06:43 long.q@n63      1
  1302261 0.06667 alpha6.sh  mrescan      r     05/06/2012 12:07:03 infinite.q@n60 1
  1314908 0.21765 evol.sh    ablanckaert  qw    05/07/2012 10:04:39                 20
  2216309 0.00045 sge_blastn gfarrant     qw    05/23/2013 14:43:03                 1 69-2379:1
```

- Prior: priority level
- State
  - r: running
  - qw: pending
  - Eqw: in error
- Slots: cores used
- Ja-task-ID: job array

Jobs type

- Interactive mode: qlogin
  - Short job and/or development
  - Prerequisite: none
  - Note: disconnect you at the end of the session

- Batch mode: qsub
  - Heavy jobs
  - Prerequisite: text editor
  - One script per job

n0 (master node) never should be used for computing!

Interactive mode: qlogin

- Connexion on:
  - A queue: qlogin -q qlogin.q
  - A node : qlogin -q qlogin.q@n72
  - A group : qlogin -q qlogin.q@@blade

```
$ qlogin -q qlogin.q

  Your job 2217414 ("QLOGIN") has been submitted
  waiting for interactive job to be scheduled ...
  Your interactive job 2217414 has been successfully scheduled.
  Establishing /opt/sge/qlogin.sh session to host n78 ...
  Last login: Mon Apr 15 10:22:01 2013 from n0.sb-roscoff.fr

@n78$ cdprojet
.
<my test>
.
@n78$ exit

  Connection to n78 closed.
  /opt/sge/qlogin.sh exited with exit code 0

$
```

Batch mode: qsub

- Progress:
  - Script edition
  - Choose the right queue
  - Submitting → Execution → Results

- Edition
  - In command line: vi, vim, nano…
  - In graphic mode: gedit, kate…

How to?

1. Prepare script of executable commands

2. Submit to batch system (returns a job ID)

3. Use the job ID for job control (query status, cancel, ...)

4. Check the job status (no execution error)

1. Prepare script of executable commands

It's always the same structure

Header

```
#!/bin/bash
#$ -S /bin/bash
#$ -M acormier@sb-roscoff.fr
#$ -m bea
#$ -V
#$ -cwd
```

cmd lines

```
blastall -p blastp -d nr -i query_1.fa -o blastout_1.txt
blastall -p blastp -d nr -i query_2.fa -o blastout_2.txt
```

Essential for qsub:

- The header:
  - Shell path
  - -S : path to shell (for SGE)
  - -m b|e|a|s|n|...: send mail at beginning|end|...of the job
  - -M: E-mail address for notification
- The command line(s)

1. Prepare script of executable commands

```
#!/bin/bash
#$ -S /bin/bash
#$ -M acormier@sb-roscoff.fr
#$ -m bea
#$ -V
#$ -cwd
#$ -q short.q
#$ -pe thread 2

blastall -p blastp -d nr -i query_1.fa -o blastout_1.txt
blastall -p blastp -d nr -i query_2.fa -o blastout_2.txt
```

Optionnal for qsub:

- The header:
  - -q queue
  - -p priority
  - -P name of project
  - …

2. Submit to batch system (returns a job ID)

```
$ qsub -q short.q blast.sh

  Your job 2217418 ("blast.sh") has been submitted

$ qsub -q short.q@n60 blast.sh

  Your job 2217419 ("blast.sh") has been submitted

$ qsub -q short.q@@blade blast.sh

$ qsub -q short.q -pe thread 2 blast.sh

$ qsub -q short.q@@bignode -pe thread 20 blast.sh
```

```
Job 1236477 (tophat.sh) Started
 User       = acormier
 Queue      = long.q
 Host       = n77
 Start Time = 05/25/2013 13:30:39
```

- -q : select a queue
- @ : select a node
- @@ : select a group
- -pe : parallelisation (thread / MPI)

```
$ qsub blast.sh

  Your job 2217418 ("blast.sh") has been submitted
```

If options are defined in the script

2. Submit to batch system (returns a job ID)

- The choice of the queue is subject to several criteria …
  - Job duration:
    - <12h → short.q
    - 12h > job > 10j → long.q
    - > 10j → infinite.q
  - Thread number:
    - < 6 → all nodes
    - > 6 → @bignode, @intel22
  - RAM
    - < 4Go → @blade
    - > 4Go → @bignode, @intel22
    - For huge amount of RAM → bigmem.q
  - Tools:
    - CLC Assembly Cell → clc.q

2. Submit to batch system (returns a job ID)

- ... But also rules, because you are not alone on the cluster!
  - **Work in the scratch directory**
  - Choose the more adapted queue (by default, used long.q)
  - Disconnect you from your qlogin

3. Use the job ID for job control (status,...)

```
$ qstat #shows all jobs

  job-ID  prior   name       user         state submit/start at     queue       slots ja-task-ID
  -----------------------------------------------------------------------------------------------
  1236477 0.50000 QLOGIN     wcarre       r     04/30/2012 15:10:26 long.q@n77      1
  1236479 0.50000 QLOGIN     wcarre       r     04/30/2012 15:11:12 long.q@n77      1
  1268627 1.00000 QLOGIN     balzano      r     05/03/2012 09:39:24 long.q@n74      1
  1302174 0.06667 alpha4.sh  mrescan      r     05/06/2012 12:06:33 long.q@n70      1
  1302175 0.06667 alpha5.sh  mrescan      r     05/06/2012 12:06:43 long.q@n63      1
  1302261 0.06667 alpha6.sh  mrescan      r     05/06/2012 12:07:03 infinite.q@n60 1
  1314908 0.21765 evol.sh    ablanckaert  qw    05/07/2012 10:04:39                 20
  2216309 0.00045 sge_blastn gfarrant     qw    05/23/2013 14:43:03                 1 69-2379:1
```

- Prior: priority level
- State
  - r: running
  - qw: pending
  - Eqw: in error
- Slots: cores used

```
$ \qstat #shows my jobs

  job-ID  prior   name       user         state submit/start at     queue       slots ja-task-ID
  -----------------------------------------------------------------------------------------------
  1236477 1.00000 tophat.sh  acormier       r     05/25/2013 15:10:26 long.q@n77      8
```

3. Use the job ID for job control (status,...)

- qstat: shows all jobs (running, pending, error)
- qstat –s r: shows only running jobs
- qstat –s p: shows only pending jobs
- \qstat: shows only **my** jobs
- qstat -g c: list available queues
- qstat -j <job id>: informations about the job
- man qstat: help

3. Use the job ID for job control (status,...)

```
$ qdel 2217417 #deletion by the job-ID

  acormier has registered the job 2217417 for deletion

$ qdel -f 2217418

  acormier has registered the job 2217418 for deletion

$ qdel -u acormier #deletion by user name

  acormier has registered the job 2217419 for deletion
  acormier has registered the job 2217420 for deletion
  acormier has registered the job 2217421 for deletion
  acormier has registered the job 2217422 for deletion
```

Job 2217424 (clc_mapping_info.sh) was killed by
acormier@n0.sb-roscoff.fr

Job 2217424 (clc_mapping_info.sh) Aborted
Exit Status = 137
Signal = KILL
User = acormier
Queue = clc.q@n76
Host = n76.sb-roscoff.fr
Start Time = 05/30/2013 21:24:06
End Time = 05/30/2013 21:24:08
CPU = 00:00:01
Max vmem = 75.023M
failed assumedly after job because:
job 2217424.1 died through signal KILL (9)

4. Check the job status (no execution error)

```
Job 1236477 (tophat.sh) Complete
 User           = acormier
 Queue          = long.q@n77
 Host           = n77
 Start Time     = 05/25/2013 13:30:39
 End Time       = 05/25/2013 17:39:00
 User Time      = 04:01:31
 System Time    = 00:06:02
 Wallclock Time = 04:08:21
 CPU            = 04:07:33
 Max vmem       = 10.976G
 Exit Status    = 0
```

<my_script>**.e**<job-ID> : error file and/or progress bar

<my_script>**.o**<job-ID> : results, except if the program provides an option to output file.

And in mutlithreading mode:

<my_script>**.pe**<job-ID>

<my_script>**.po**<job-ID>

Job-array

- Job-array: the idea is to split a job in a large number of sub-job
  - Very high load on the cluster
  - Problem with slot reservation
  - To prevent this problem:

Job-array should be run only on short.q!

Exercises

1. Simple script
2. Multithread script
3. Job-array

1. My first script

- With graphic interface

```
$ cd script/ #qsub are stored in the "script/" folder

$ gedit mon_script.sh #if the file does not exist, it's created, else it's edited
```

- Efficient way: in terminal

```
$ vim my_script.sh #if the file does not exist, it's created, else it's edited
```

```
~
~
~
"my_script.sh" [new file]
```

1. My first script


```
#!/bin/bash
# Comments that start with '#$' are
# interpreted by SGE as directives

# Shell to use for the execution
#$ -S /bin/bash

# Notified user
#$ -M alexandre.cormier@sb-roscoff.fr

# Export all environment variables
#$ -V

# Notifying at the (b)egin, at the end , when (a)bort and
# when (s)uspend a job
#$ -m bea

# Launched the command in the current working directory
#$ -cwd
```

1. My first script


```
#!/bin/bash
#$ -S /bin/bash
#$ -M alexandre.cormier@sb-roscoff.fr
#$ -V
#$ -m bea
#$ -cwd

echo ""
date
echo ""
echo "Job running on the node: "
hostname
echo ""
ls -lart
echo ""
echo "Job done - Check your email"
echo ""
```

1. My first script

- Launch my_script.sh on long.q

- Test options, and checking
  - Results in a other terminal
  - Cluster state
  - Jobs running

1. My first script

```
$ qsub -q long.q mon_script.sh
```

```
$ ll

  -rw-r--r-- 1 acormier ga       474 mai 31 09:53 my_script.sh
  -rw-r--r-- 1 acormier ga         0 mai 31 09:53 my_script.sh.e2217433
  -rw-r--r-- 1 acormier ga      1538 mai 31 09:53 my_script.sh.o2217433
```

We get two new files:

- <my_script>**.e**<job-ID> : error file and/or progress bar
- <my_script>**.o**<job-ID> : results, except if the program provides an option to output file.

1. Blast

- Create a qsub script to launch a blastn with blastall
  - Blastall –help
  - Input file: insulin.fasta
- Parameters
    - -p blastn
    - -m 8
    - -e 1e-6
    - -v 5
    - -b 5
    - -d /db/blast/all/nt
- Tips
  - keep in mind that your project directory is structured (input, scratch, script…)

1. Blast script

```
#!/bin/bash
#$ -S /bin/bash
#$ -M alexandre.cormier@sb-roscoff.fr
#$ -V
#$ -m bea
#$ -cwd

INPUT="/projet/umr8227/ga/acormier/input/insulin.fasta"
OUTPUT="/projet/umr8227/ga/acormier/tmp/blast/insulin.blast"
DATABASE="/db/blast/all/nt"

blastall -p blastn -o $OUTPUT -i $INPUT -d $DATABASE -m 8 -e 1e-6 -v 5 -b 5
```

```
$ qsub -q short.q qsub_blastn.sh

Your job 2217418 ("qsub_blastn.sh") has been submitted
```

2. Multithread script

- Need to add 2 arguments:
  - 1 argument in the script defined by the software
  - 1 argument to submit during the qsub or in the script

- Redo the blast un multithread mod

```
-Q  Query Genetic code to use [Integer]
  default = 1
-D  DB Genetic code (for tblast[nx] only) [Integer]
  default = 1
-a  Number of processors to use [Integer]
  default = 1
-O  SeqAlign file [File Out]  Optional
-J  Believe the query defline [T/F]
  default = F
-M  Matrix [String]
  default = BLOSUM62
```

2. Multithread script

```
#!/bin/bash
#$ -S /bin/bash
#$ -M alexandre.cormier@sb-roscoff.fr
#$ -V
#$ -m bea
#$ -cwd

INPUT="/projet/umr8227/ga/acormier/input/insulin.fasta"
OUTPUT="/projet/umr8227/ga/acormier/tmp/blast/insulin_results.txt"
DATABASE="/db/blast/all/nt"

blastall -p blastn -o $OUTPUT -i $INPUT -d $DATABASE -m 8 -e 1e-6 -v 5 -b 5 -a 2
```

```
$ qsub -q short.q -pe thread 2 qsub_blastn.sh

Your job 2217418 ("qsub_blastn.sh") has been submitted
```

The CPU/thread/core value in the script and in the qsub must be identical!

```
#!/bin/bash
#$ -S /bin/bash
#$ -M alexandre.cormier@sb-roscoff.fr
#$ -V
#$ -m bea
#$ -cwd
#$ -q short.q
#$ -pe thread 2
```

2. Multithread script

- Example of argument to define the thread value:
  - TopHat: -p / --num-threads
  - Bowtie2: -p / --threads
  - Trinity: --CPU
  - CLC Assembly Cell: --cpus

- Multithreading is not possible with all software

4. Job-array

<u>Problem:</u> a large number of jobs to run and they are largely identical in terms of the command to run.

For example, you may have 1000 data sets, and you want to run a single program on them.

<u>Naive solution:</u> generate 1000 shell scripts, and submit them to the cluster.

<u>Best solution:</u> on SGE systems – array jobs. The advantages are:

- You only have to write one shell script

One Script to rule them all, One Script to find them,
One Script to bring them all and in the darkness bind them

4. Job-array: Example

```
$ ls

sctg_129-134.EctsiV2_prot_27052015.fasta
sctg_338-352.EctsiV2_prot_27052015.fasta
sctg_7-8.EctsiV2_prot_27052015.fasta
...
```

n fasta files (proteins), with different names but the same extension (.fasta) and I need to make the same analysis == InterproScan

1. Create the main structure

```
#!/bin/bash
#$ -S /bin/bash
#$ -M acormier@sb-roscoff.fr
#$ -V
#$ -m bea
#$ -cwd


OUT="/scratch/umr8227/ga/acormier/iprscan/ectsi_v2.prot/"

/opt/6.x/interproscan/interproscan.sh -i $INPUT -t p -d $OUT -f xml -iprlookup -goterms
```

4. Job-array: Example

2. Add job-array options

```
#!/bin/bash
#$ -S /bin/bash
#$ -M acormier@sb-roscoff.fr
#$ -V
#$ -m bea
#$ -cwd

INPUT=`ls ectsi_v2.prot/*.fasta | awk "NR==$SGE_TASK_ID"`

echo -e `date '+%y%m%d-%H:%M'`"\t"$SGE_TASK_ID"\t"$INPUT >> qsub_array_files.tab

OUT="/scratch/umr8227/ga/acormier/iprscan/ectsi_v2.prot/"

/opt/6.x/interproscan/interproscan.sh -i $INPUT -t p -d $OUT -f xml -iprlookup -goterms
```

4. Job-array: Example

2. Add job-array options

```
#!/bin/bash
#$ -S /bin/bash
#$ -M acormier@sb-roscoff.fr
#$ -V
#$ -m bea
#$ -cwd

INPUT=`ls ectsi_v2.prot/*.fasta | awk "NR==$SGE_TASK_ID"`

echo -e `date '+%y%m%d-%H:%M'`"\t"$SGE_TASK_ID"\t"$INPUT >> qsub_array_files.tab

OUT="/scratch/umr8227/ga/acormier/iprscan/ectsi_v2.prot/"

/opt/6.x/interproscan/interproscan.sh -i $INPUT -t p -d $OUT -f xml -iprlookup -goterms
```

INPUT=`ls ectsi_v2.prot/*.fasta | awk "NR==$SGE_TASK_ID"`

Give a new file at each iteration of the variable $SGE_TASK_ID
When $SGE_TASK_ID==1; INPUT==sctg_129-134.EctsiV2_prot_27052015.fasta

echo -e `date '+%y%m%d-%H:%M'`"\t"$SGE_TASK_ID"\t"$INPUT >>
qsub_array_files.tab

Give a file to make the link between the $SGE_TASK_ID and the filename

4. Job-array: Example

3. Launch the job-array

```
$ qsub -t 1-`ls *.fasta | wc -l` -q long.q -sync no -tc 10 easy_array.qsub
```

It will define the $SGE_TASK range
With 1000 files:
-t 1-1000

Limits the number of jobs running at the same time (optional)

4. Job-array: Atomicblastplus.py

Standard blast: weak performances with big dataset against huge databases (nr, nt,…)

Solution: splitting your set of sequences in order to create a job-array:

- A way to parallelized blast

```
$ atomicblastplus -p blastn -i input.fasta -o myproject/tmp/blast/input_vs_nr -d
/db/blast/all/nt -e 1e-4 -n 100 --verbose

!!! This is Atomic Blast !!!

PROGRAM:
blastn: 2.2.28+
Package: blast 2.2.28, build Mar 12 2013 16:52:31

QUERY: insulin.fasta
DB: /db/blast/all/nt
OUTDIR: test

INFO: The query was splitted into 1 subfiles

INFO: SGE qsub script was written to test/qsub.insulin.atomic_blastn_vs_nt.sh

INFO: Running job-array on SGE...

CMD: qsub -q short.q -t 1-1 -tc 100 -sync yes -N at_blastn_insulin.atomic_blastn_vs_nt
test/qsub.insulin.atomic_blastn_vs_nt.sh
```

```
$ atomicblastplus -p blastn -i input.fasta -o myproject/tmp/blast/input_vs_nr -d
/db/blast/all/nt -e 1e-4 -n 10 --dont_wait --verbose
```

End