Session de formation 2018

- Toutes nos formations :

https://southgreenplatform.github.io/trainings/

- Environnement de travail : **Logiciels à installer**

Initiation HPC cluster

www.southgreen.fr

https://southgreenplatform.github.io/trainings

Objectifs du module

Objectif

Acquérir les bonnes pratiques pour utiliser un cluster de calcul

Applications

- Connaître l'architecture d'un cluster
- Connaître le rôle des différentes partitions
- Utiliser SGE (qusb, qrsh, qhost, qacct, qstat, qqdel)
- Utiliser les modules environment
- Faire du scripting de base

ARCHITECTURE

Qu’est ce qu’un cluster?

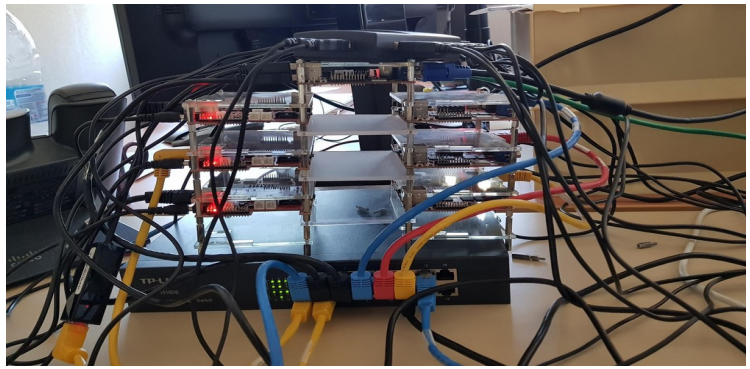
- Un cluster est une unité logique constituée de plusieurs serveurs
- Agit comme une unique machine puissante
- Permet d’obtenir une puissance de calcul élevée
- Une plus grande capacité de stockage
- Une fiabilité supérieure
- Une plus grande disponibilité des ressources

Qu’est ce qu’un cluster?

- Un cluster est une unité logique constituée de plusieurs serveurs
- Agit comme une unique machine puissante
- Permet d’obtenir une puissance de calcul élevée
- Une plus grande capacité de stockage
- Une fiabilité supérieure
- Une plus grande disponibilité des ressources

Qu’est ce qu’un cluster?

- Un cluster est une unité logique constituée de plusieurs serveurs
- Agit comme une unique machine puissante
- Permet d’obtenir une puissance de calcul élevée
- Une plus grande capacité de stockage
- Une fiabilité supérieure
- Une plus grande disponibilité des ressources



Composants d'un cluster

- Nœud maître : Ordonnanceur.

Gère les ressources et les priorités des jobs

- Nœuds de calcul : Ressources (CPU ou mémoire RAM) utilisées par le master

CALCUL

Composants d'un cluster

- Nœud maître :
  Ordonnanceur.
  Gère les ressources et les priorités des jobs

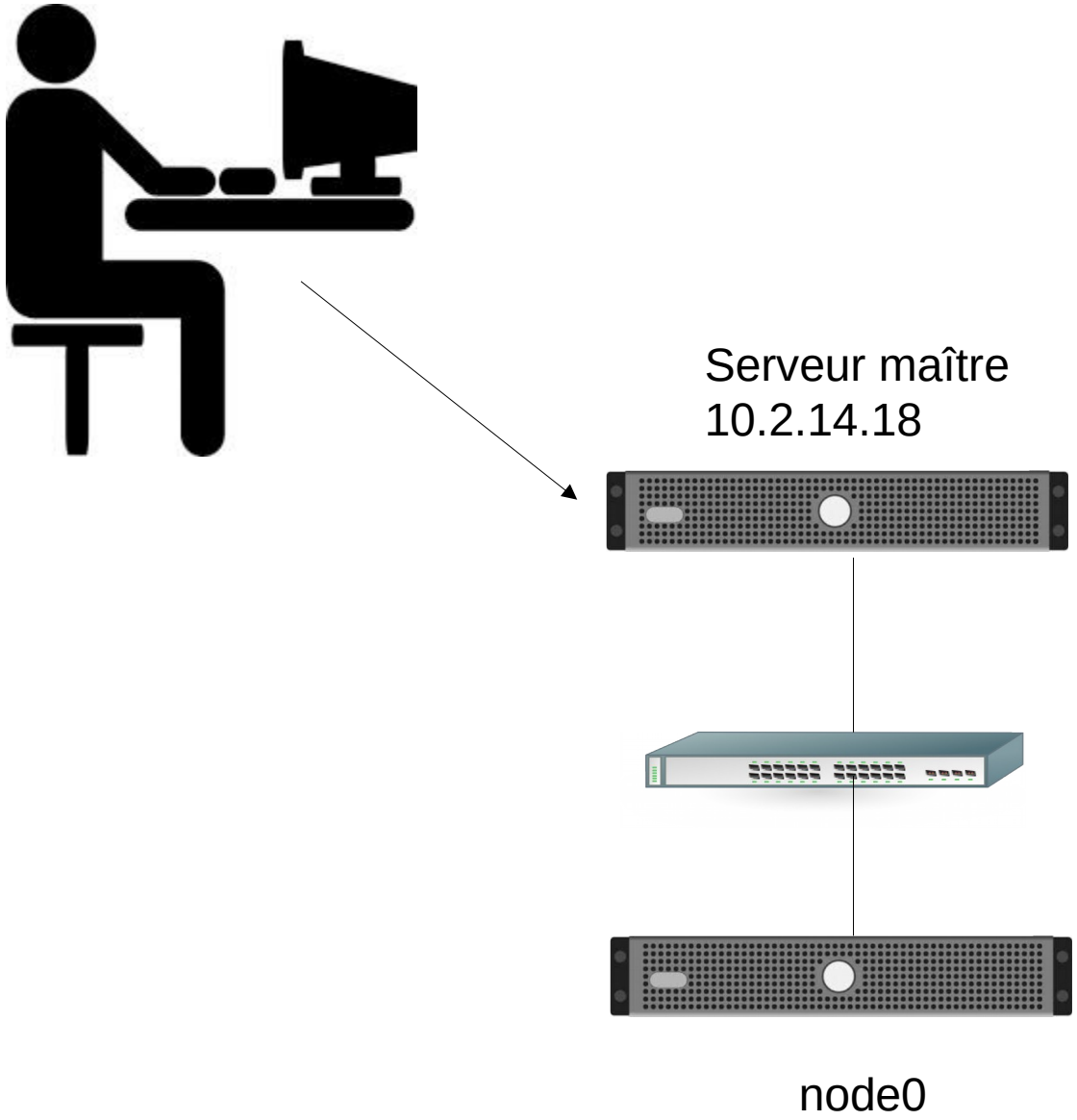
- Nœuds de calcul :
  Ressources (CPU ou mémoire RAM) utilisées par le master

- Serveur(s) NAS :
  Stockent les données utilisateurs et les résultats d'analyses

CALCUL

STOCKAGE

Composants du cluster ceraas



Partitions sur le cluster ceraas

/home : Votre répertoire personnel
Hébergée sur : le master
Partagée sur toutes les machines

/data : Données projet
Hébergée sur : le master
Partagée sur toutes les machines

/usr/local/softs/bioinfo : logiciels bioinformatiques
hébergée sur : le master
Partagée sur toutes les machines

/scratch : Répertoire temporaire de travail
Hébergée sur : le noeud de calcul
Pas partagée mais uniquement en local
Données conservées **3 semaines**

SUN GRID ENGINE (SGE)

L'ordonnanceur sge

- SGE (SUN Grid Engine) est un gestionnaire de ressources de calcul sous linux, capable de gérer de deux à des milliers de serveurs et des centaines de clusters de plusieurs nœuds à la fois.

- Un outil opensource
- 3 fonctions principales :

  - Alloue les ressources (CPU,RAM) aux utilisateurs pour qu'ils puissent lancer leurs analyses
  - Fournit un cadre pour lancer,exécuter et monitorer les jobs sur l'ensemble des nœuds alloués
  - Gère la priorité des jobs en file d'attente

Commandes sge: allocation de ressources

Actions	Commandes
●Réserver un coeur sur un nœud de manière interactive	$ **qrsh**
●Réserver un coeur sur un noeud en particulier	$ **qrsh -l hostname=nodeX** Avec X le numéro du noeud
●Réserver X coeur sur un noeud	$~ **qrsh -pe ompi X** Avec X : le nombre de processeurs de 0 à 12

Commandes sge: Options de qsub

Actions	Commandes
●Lancer un script en mode batch	$**qsub + script.sh**
●Propager l'environnement chargé au noeud	$**qsub -V script.sh**
●Donner un nom à votre job	$~ **qsub -N job_name script.sh**
●Utiliser plusieurs processeurs	$~ **qsub -pe ompi X script.sh** **Avec X le nombre de coeurs à utiliser**
●Demander un certain montant de RAM	**$~ qsub -l mem_free=XG script.sh** **Avec X le montant de mémoire à réserver**
●Demander un noeud en particulier	**$~ qsub -l hostname=nodeX script.sh**
●Lancer directement une commande avec qsub	**$~ qsub -b y command**

Commandes sge: Informations

Actions	Commandes
●Informations sur l'état des noeuds	$ **qhost**
●Voir ses jobs en cours	$~ qstat
	$~ **qstat -j <JOB_ID>**
●Informations sur les jobs lancés	With JOB_ID :the job number
	$~ **qacct -j <JOB_ID>**
●Informations sur les jobs terminés	With JOB_ID :the job number
●Informations globales sur les queues	$~ **qstat -g c**

Commandes sge: suppression

Actions	Commandes
● Suppression d’un job	$~ **qdel <JOB_ID>** avec JOB_ID : l’identifiant du job

1

TP: Lancer une analyse blast de manière interactive

Etape1: copie des données sur le cluster

Ouvrir filezilla et récupérer le fichier « HPC_thies.pdf »
dans /data/FORMATION/2018/tp-cluster/

Etape1: copie des données sur le cluster

Ouvrir filezilla et récupérer le fichier « HPC_french.pdf »
dans /data/FORMATION/2018/tp-cluster/

Renseigner les paramètres suivants :

Hostname : **nom du serveur maître**
Login : votre login
Mot de passe : votre login
Port :**22**
Naviguer dans la fenêtre de droite jusqu'à
/data/FORMATION/2018/tp-cluster/
Récupérer le fichier HPC_thies.pdf en faisant un glisser-déposer

Connexion au cluster

Lancer des scripts sur le noeud de calcul

10.2.14.18

Avec la commande qsub

Hostname : 10.2.14.18

Login : cluster account

Password : mot de passe du compte
Port : **22**

En utilisant putty ou MobaXterm et les paramètres ci-dessus

Utiliser la commande ssh avec le terminal

Etape2: réservation d'un coeur sur un noeud

Se connecter au maître via ssh

Taper :
$~ssh login@nomserveurmaitresur Apple ou Linux

Sous windows : télécharger Mobaxterm à l'URL :
https://mobaxterm.mobatek.net/download-home-edition.html
Puis se connecter à la machine maitre

Etape2: réservation d'un coeur sur un noeud

On peut réserver un nœud par lancer une analyse pendant une durée limitée en utilisant la commande qrsh
Taper la commande qstat et analyser le résultat

Etape2: réservation d'un coeur sur un noeud

On peut réserver un processeur sur un nœud pour lancer une analyse
pendant une durée limitée en utilisant la commande qrsh
Taper la commande qstat et analyser le résultat

Type :
$~qrsh
Vérifier sur quel noeud vous êtes avec la commande
$~ uname -a
$~ qstat

Etape3 : création d’un répertoire

Se déplacer dans le répertoire d'accueil des données temporaires
/scratch
Créer un répertoire pour y accueillir nos données

Etape3 : création d'un répertoire

Se déplacer dans le répertoire d'accueil des données temporaires
/scratch
Créer un répertoire pour y accueillir nos données

Taper les commandes :
$~cd /scratch
$~ mkdir login (avec le login le mot de répertoire de son choix)

La copie avec scp

Copie entre 2 serveurs distants :

scp source destination

Syntaxe si la source est distante :

scp nom_serveur:/chemin/fichier_a_copier répertoire_local

Syntaxe si la destination est distante :

scp /chemin/fichier_a_copier nomserveur:/chemin/répertoire_local

Etape4 : copie des données dans le répertoire d'analyses

Copier le répertoire /data/FORMATION/2018/tp-cluster/Blast dans /scratch/login

Etape4 : copie des données dans le répertoire d'analyses

Copier le répertoire /data/FORMATION/2018/tp-cluster/Blast dans /scratch/login

Taper les commandes :

```
$~cd /scratch/login
$~ scp -r  master:/data/FORMATION/2018/tp-cluster/Blast /scratch/login
```

Etape5 : se déplacer dans le répertoire copié

Aller dans le répertoire /scratch/login/Blast
Lister les fichiers du répertoire

Module Environment

- Permet de choisir la version du logiciel que l'on veut utiliser
- 2 types de logiciels :
- bioinfo : désigne les logiciels de bioinformatique (exemple BEAST)
  system : désigne tous les logiciels systèmes(exemple JAVA)
- Surpassent les variables d'environnement

Module Environment : les commandes

➢ 5 types de commandes :

- Voir les modules disponibles :
  - module avail
- Obtenir une info sur un module en particulier :
  - module whatis + module name
- Charger un module :
  - module load + modulename
- Lister les modules chargés :
  - module list
- Décharger un module :
  - module unload + modulename
- Décharger tous les modules :
  Module purge

Etape6 : Lancer une analyse

Charger le module version 2.7.1+
Utiliser la commande blastn pour lancer une analyse blast
qui fournira le fichier de sortie appelé blastn.out

Etape6 : Lancer une analyse

Charger le module version 2.7.1+
Utiliser la commande blastn pour lancer une analyse blast
qui fournira le fichier de sortie appelé blastn.out

Taper :
$~ module load bioinfo/blast/2.7.1+
$~ blastn -db All-EST-coffea.fasta -query sequence-NMT.fasta -out blastn.out

Etape7: analyse du fichier de résultat

Editer le fichier blastn.out avec l'utilitaire nano

Etape7 : Analyse du fichier de résultat

Editer le fichier blastn.out avec l'utilitaire nano

Taper :
$~ nano blastn.out

Etape8 : Copie du résultat vers son /home

Copier le fichier blastn.out vers son répertoire home utilisateur
Vérifier que le fichier est bien copié

Etape8 : Copie du résultat vers son /home

Copier le fichier blastn.out vers son répertoire home utilisateur
Vérifier que le fichier est bien copié

Taper :
$~scp blastn.out 10.2.14.18:/home/login
$~ ls -ali /home/login

Etape9 : Suppression du répertoire dans /scratch

Se déplacer dans le répertoire
Supprimer le répertoire de travail

Taper:
$~cd /scratch
$~ rm -rf *login*

2

TP: Lancer un bwa de manière interactive

TP: Lancer une analyse bwa

- Suivre les étapes du TP précédent et les adapter à celui-ci
- Le répertoire à copier est: /data/FORMATION/2018/tpcluster/bwa
- La version de bwa à utiliser est la 0.7.12
- Les commandes à lancer sont:

bwa index referenceIrigin.fasta

bwa mem referenceIrigin.fasta irigin1_1.fastq irigin1_2.fastq >mapping.sam

- Récupérer le fichier mapping.sam et le mettre dans son /home/login

3

TP: Lancer une analyse à l’aide d’un script

Lancer un job en mode batch

- C’est le fait d’exécuter un script bash via sge
- On utilise la commande:

```
$~ qsub script.sh
```

Avec script.sh : le nom du script

Syntaxe des scripts bash

Dans la première partie du script on renseigne les options d'exécution de sge avec le mot clé #$ (partie

```
#!/bin/sh


############      SGE CONFIGURATION      ####################
# Ecrit les erreur dans le fichier de sortie standard
#$ -j y

# Shell que l'on veut utiliser
#$ -S /bin/bash



# Email pour suivre l'execution
#$ -M prenom.nom@ird.fr          ######### Mettre son adresse mail

# Type de message que l'on reçoit par mail
#    -  (b) un message au demarrage
#    -  (e) a la fin
#    -  (a)  en cas d'abandon
#$ -m bea

# Queue que l'on veut utiliser
#$ -q all.q

# Nom du job
#$ -N Nom_a_choisir
##############################################################
```

Syntaxe des scripts bash

Dans la 2e partie du script on renseigne les actions à effectuer

```
path_to_dir="/data/FORMATION/2018/tp-cluster/rep_a_choisir";
path_to_tmp="/scratch/nom_rep_a_choisir-$JOB_ID"

###### Creation du repertoire temporaire sur noeud et chargement du module blast
module load bioinfo/blastn/2.7.1+
mkdir $path_to_tmp
scp -rp master:$path_to_dir/* $path_to_tmp
echo "tranfert donnees master -> noeud";
cd $path_to_tmp

###### Execution du programme
cmd="blastn -db All-EST-coffea.fasta -query  sequence-NMT.fasta -num_threads $NSLOTS  -out blastn1-
$JOB_ID.out";
echo "Commande executee : $cmd";
$cmd;

##### Transfert des données du noeud vers master
scp -r $path_to_tmp/ master:$path_to_dir/
echo "Transfert donnees node -> master";

#### Suppression du repertoire tmp noeud
rm -rf $path_to_tmp
echo "Suppression des donnees sur le noeud";
```

Création du script blastn

- Reprendre le TP 1 et le mettre sous forme de script en s'aidant du script exemple précédent
- Rendre le script exécutable avec la commande

$~ **chmod 755 script.sh**

- Lancer le script avec la commande qsub

$~ **qsub script.sh**

Utiliser la commande dos2unix quand le script a été écrit sous windows

Création du script bwa

- Reprendre le TP 2 et le mettre sous forme de script en s'aidant du script exemple précédent
- Rendre le script exécutable avec la commande

$~ **chmod 755 script.sh**

- Lancer le script avec la commande qsub

$~ **qsub script.sh**

- Observer le déroulement avec la commande watch qstat

Utiliser la commande dos2unix quand le script a été écrit sous windows