

# Initiation HPC cluster





# Présentation i-Trop



 Yves Vigouroux	 Jean Benoit Morel/ Gilles Béna	 Eric Delaporte/ Maryline Bonnet	 Frédéric Simard
<b>DIRECTION</b>			



Julie ORJUELA-  
BOUNOL, IE  
Bioinformaticienne



Aurore COMTE, IE  
Bioinformaticienne



Bruno GRANOULLAC, IE  
Systèmes d'information



Ndomassi TANDO, IR  
Ingénieur systèmes  
Animateur plateau, RMQ

Alexis DEREPPER, IE  
Bioinformaticien



Jacques Dainat, IR  
Bioinformaticien



Nicolas Fernandez, IR  
Bioinformaticien

Christine TRANCHANT-  
DUBREUIL, IR  
Bioinformaticienne



# Présentation i-Trop





# Demandes/incidents/Howtos

- Formulaires de demandes  
<https://bioinfo.ird.fr/index.php/cluster-fr/>
  - Comptes
  - Installation logiciels
  - Projets
- Incidents: contacter [bioinfo@ird.fr](mailto:bioinfo@ird.fr)
- Howtos:  
<https://bioinfo.ird.fr/index.php/tutorials-fr/howtos-for-hpc-cluster-itrop/>
- Tutorials Slurm:  
<https://bioinfo.ird.fr/index.php/tutorials-fr/slurm/>
- FAQ:  
<https://bioinfo.ird.fr/index.php/faq-fr/>



# ARCHITECTURE



# Qu'est ce qu'un cluster?

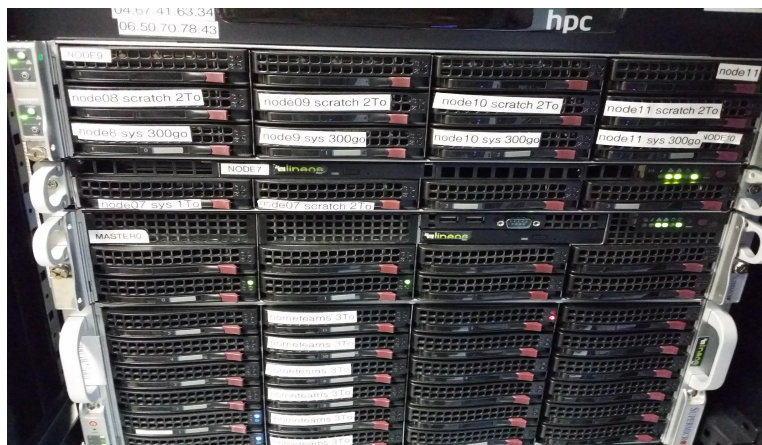
- une unité logique de plusieurs serveurs
- une unique machine puissante
- une puissance de calcul élevée
- Une plus grande capacité de stockage
- Une fiabilité supérieure
- Une plus grande disponibilité des ressources



# Qu'est ce qu'un cluster?

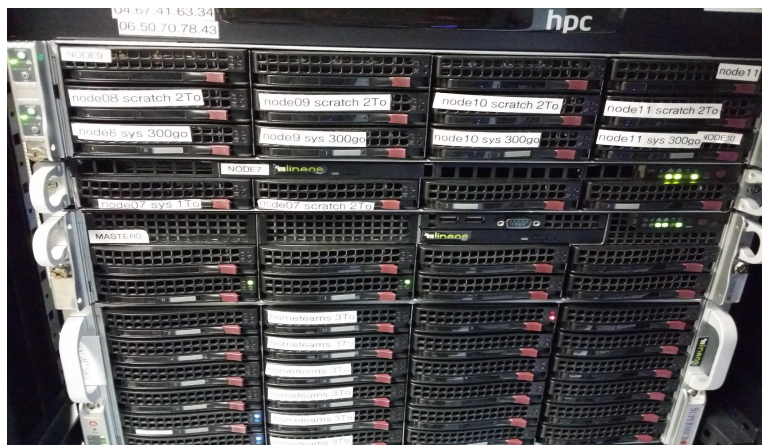
- une unité logique de plusieurs serveurs
- une unique machine puissante
- une puissance de calcul élevée
- Une plus grande capacité de stockage
- Une fiabilité supérieure
- Une plus grande disponibilité des ressources





- **Noeud maître**  
Gère les ressources et les priorités des jobs
- **Noeuds de calcul**  
Ressources (CPU ou mémoire RAM)

CALCUL



- **Noeud maître**  
Gère les ressources et les priorités des jobs
- **Noeuds de calcul**  
Ressources (CPU ou mémoire RAM)

STOCKAGE



- **Baie SAN**  
Stockage

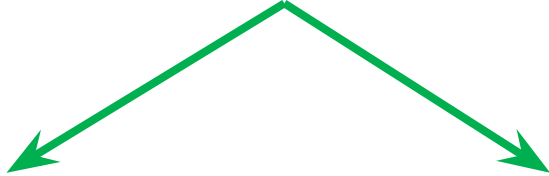


# Architecture du cluster i-Trop



**laptop client**

**ssh connection**



**bioinfo-master1.ird.fr**

**bioinfo-san.ird.fr**





# Practice

Etape 1: Connexion, sinfo

1

*Aller sur le [Practice 1](#) du github*



# Les files d'attente

Partitions	Utilisation	Caractéristiques RAM noeuds	Caractéristiques coeurs noeuds
short	Jobs courts < 1 jour	48 à 64 Go	12 coeurs
normal	Jobs courts max 14 jours	64 Go à 512 Go	12 à 112 coeurs
long	45 jours >Jobs longs > 14 jours	48 Go	12 à 24 coeurs
highmem	Jobs avec besoin de plus de mémoire (45j)	512Go à 1To	40 à 112 coeurs
supermem	Jobs avec besoin de beaucoup de mémoire	2To	256 coeurs
gpu	Besoin d'analyses sur des gpus	192Go	24 cpus et 8 coeurs GPUs



# Cas particulier : partition gpu

- Partition pour effectuer des travaux sur des processeurs GPUs: basecalling, MiniOn etc..
- Accès restreint au groupe gpu\_account
- Demande d'accès avec argumentaire à faire sur

<https://itrop.ird.fr/glipi/plugins/formcreator/front/formdisplay.php?id=15>



# Quelle partition choisir?

Règles	Partition	exemple outils	commentaire
basecalling, demultiplexing, correction	<i>gpu</i>	medaka, guppy, machine learning tools	demande d'accès
assemblages >100G RAM	<i>supermem</i>	miniasm, flye, raven, smartdenovo	génomome cible > 400 Mb (Un génomome comme le riz ne consomme pas 100Go)
genomicsbd (gatk) > 100G RAM	<i>supermem</i>	GATK genomicsDB	genomome cible de plus de 400 Mb (>10 samples)
assemblages => 35G et < 120G RAM	<i>highmem</i>	miniasm, flye, raven, smartdenovo	génomome cible entre 100 et 400 Mb
assemblages => 35G et < 100G RAM	<i>highmem</i>	miniasm, flye, raven, smartdenovo	génomome cible entre 100 et 400 Mb
structure de pops	<i>long</i>		
simulations	<i>long</i>		
metagenomic	<i>normal</i>	quiime2, frogs	
mapping	<i>normal</i>	bwa, minimap2, hisat2	besoin de bcp des coeurs pas bcp de RAM. <b>nb de coeurs tool = nb de coeurs à réserver</b>
genotypage	<i>normal</i>	GATK haplotypcaller, samtools mpileup, bcftools	besoin de bcp des coeurs pas bcp de RAM. <b>nb de coeurs tool = nb de coeurs à réserver</b>
stats	<i>normal</i>	R	
test de scripts	<i>short</i>	bash, python, R	

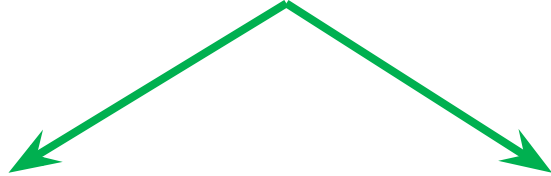


# Réservation de ressources sur le cluster



**laptop client**

**ssh connection**



**bioinfo-san.ird.fr**



**bioinfo-master1.ird.fr**



**srun or sbatch**



**node servers**





# Practice

Etape 2:srun, partition

2

*Aller sur le [Practice2](#) du github*



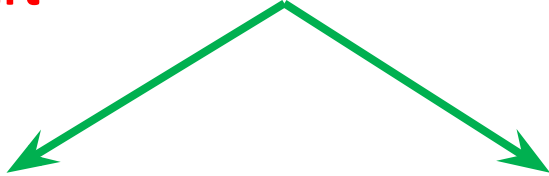
# Stockage sur le cluster i-Trop



laptop client

data transfert

ssh connection



bioinfo-san.ird.fr

bioinfo-master1.ird.fr

node servers



srun or sbatch

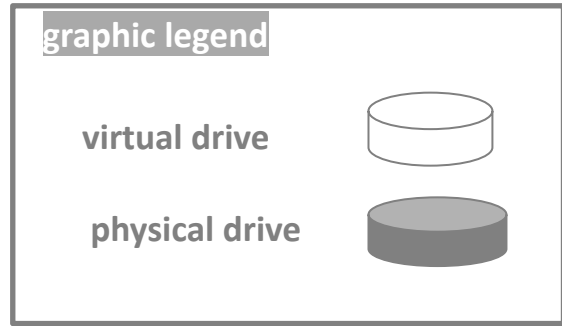




# Stockage sur le cluster i-Trop

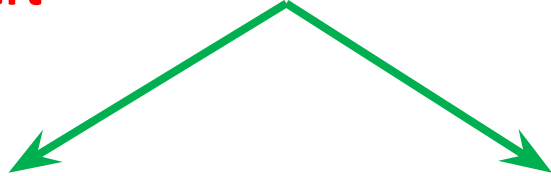


laptop client



data transfert

ssh connection



bioinfo-san.ird.fr

bioinfo-master1.ird.fr

node servers

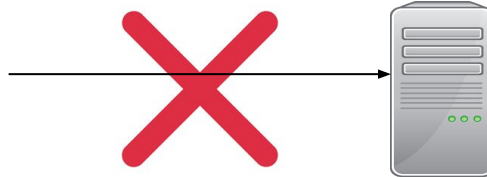


srun or sbatch





Ordinateur  
personnel



**Transfert direct  
via filezilla  
interdit**



**bioinfo-master1.ird.fr**



# Le transfert entre son PC et bioinfo-san.ird.fr

- Copie entre 2 serveurs distants :

```
rsync -ravz --progress source destination
```

- Syntaxe pour transférer de son PC vers bioinfo-san.ird.fr :

```
rsync -ravz --progress /chemin/rep_a_copier login@bioinfo-san.ird.fr:/chemin/répertoire_distant
```

- Syntaxe pour récupérer un répertoire sur son PC depuis bioinfo-san.ird.fr :

```
rsync -ravz --progress login@bioinfo-san.ird.fr:/chemin/rep_a_copier répertoire_local
```



# Practice

Etape3: transfert de fichiers

3

*Aller sur le [Practice3](#) du github*



# Stockage sur le cluster i-Trop



laptop client

data transfert

ssh connection

bioinfo-san.ird.fr

bioinfo-master1.ird.fr

node servers



srun or sbatch



/users

/users

/users

/projects

/projects

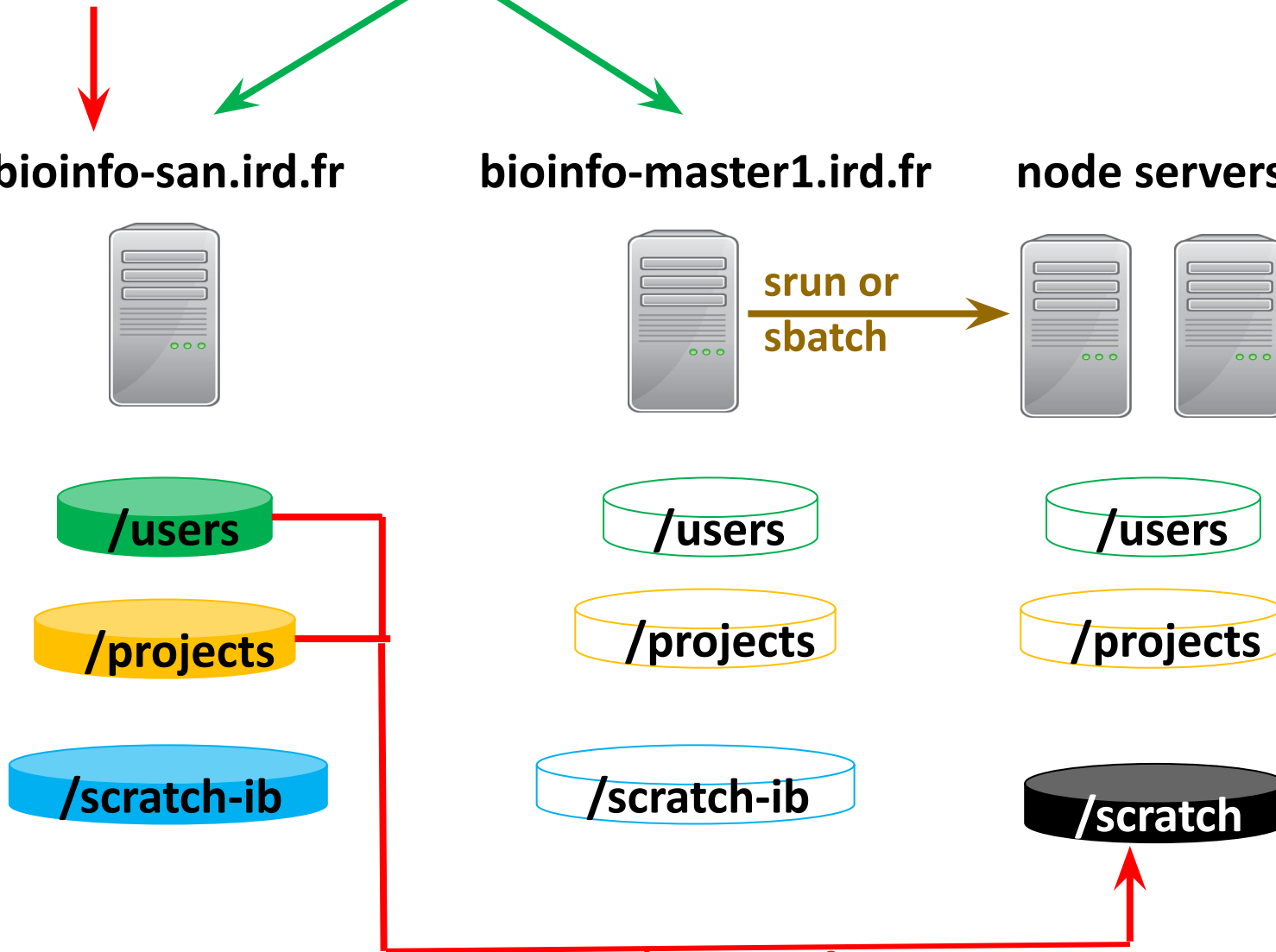
/projects

/scratch-ib

/scratch-ib

/scratch

data transfert





# Stockage sur le cluster i-Trop



laptop client

data transfert

ssh connection

bioinfo-san.ird.fr

bioinfo-master1.ird.fr

node servers



srun or sbatch

/users

/users

/users

/projects

/projects

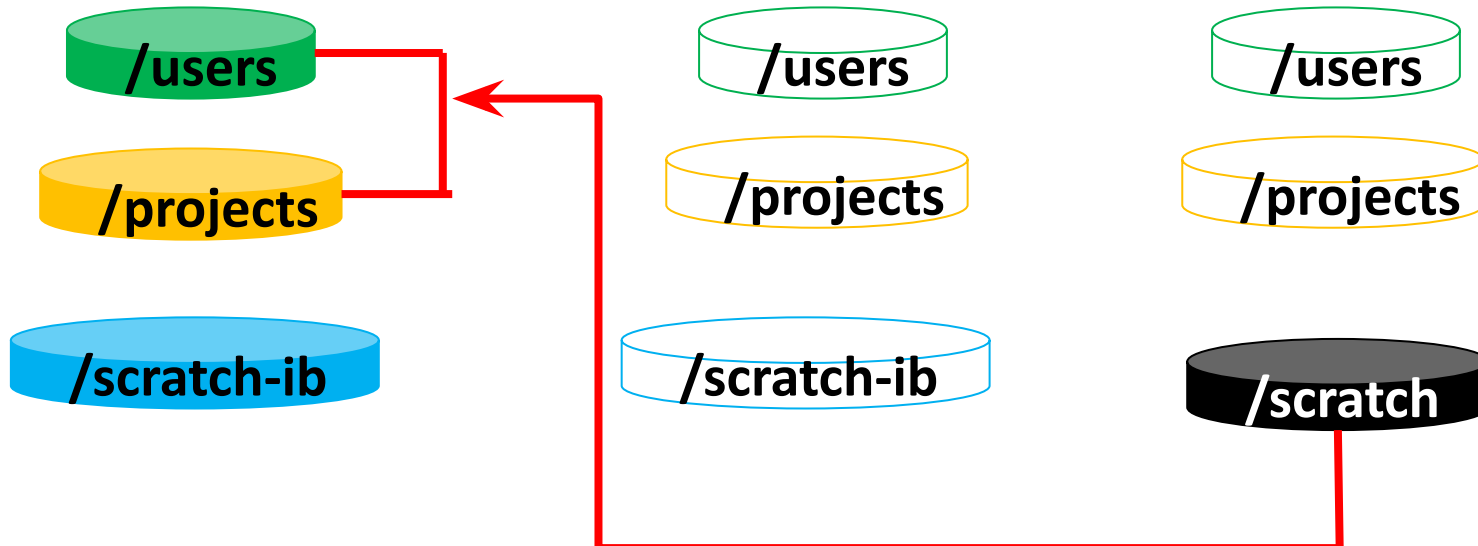
/projects

/scratch-ib

/scratch-ib

/scratch

data transfert





# Le transfert entre le san et les noeuds de calcul

- Copie entre 2 serveurs distants :

```
rsync -ravz --progress source destination
```

- Syntaxe pour récupérer un répertoire du san depuis le noeud

```
rsync -ravz --progress san:/chemin/rep_a_copier repertoire_local
```

- Syntaxe pour transférer un résultat du noeud vers le san :

```
rsync -ravz --progress /chemin/rep_a_copier san:/chemin/repertoire_distant
```



# Practice

Etape4: transfert de données  
vers noeuds

4

*Aller sur le [Practice4](#) du github*



# 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

➤ 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`



# Practice

Etape5 et 6 : module  
environnement et lancement  
analyse

5

*Aller sur le [Practice5](#) du github*



# Le transfert entre le san et les noeuds de calcul

- Copie entre 2 serveurs distants :

```
rsync -ravz --progress source destination
```

- Syntaxe pour récupérer un répertoire du san depuis le noeud

```
rsync -ravz --progress san:/chemin/rep_a_copier repertoire_local
```

- Syntaxe pour transférer un résultat du noeud vers le san :

```
rsync -ravz --progress /chemin/rep_a_copier san:/chemin/repertoire_distant
```



# Practice

## Etape7: Récupérer les résultats

7

*Aller sur le [Practice7](#) du github*



# Supprimer les résultats des scratches

- Scratch = espaces temporaires
- Vérifier la copie des résultats avant
- Utiliser la commande rm

```
cd /scratch  
rm -rf nom_rep
```



# Practice

Etape8: suppression des données

8

*Aller sur le [Practice8](#) du github*



# Scripts pour visualiser/supprimer données temporaires

- Emplacement des scripts: `/opt/scripts/scratch-scripts`
- Visualiser ses données sur les scratches: `scratch_use.sh`

```
sh /opt/scripts/scratch-scripts/scratch_use.sh
```

- Supprimer ses données sur les scratches: `clean_scratch.sh`

```
sh /opt/scripts/scratch-scripts/clean_scratch.sh
```



# Utilisation scratch-ib

- Un seul disque dur centralisé sur san et les noeuds **infiniband**
- Transferts directement vers /scratch-ib dans bioinfo-san.ird.fr
- Nécessite d'avoir un noeud infiniband avec l'option:
  - -constraint=infiniband
- Plus d'informations ici:

<https://bioinfo.ird.fr/index.php/how-to-utiliser-le-scratch-mutualise/>



# Architecture on i-Trop cluster



laptop client

ssh connection

bioinfo-san.ird.fr



bioinfo-master1.ird.fr



node servers



Infiniband  
node servers



srun or  
sbatch

--constraint=infiniband



# Stockage sur le cluster i-Trop



laptop client

data transfert

ssh connection

bioinfo-san.ird.fr

bioinfo-master1.ird.fr

node servers

Infiniband node servers



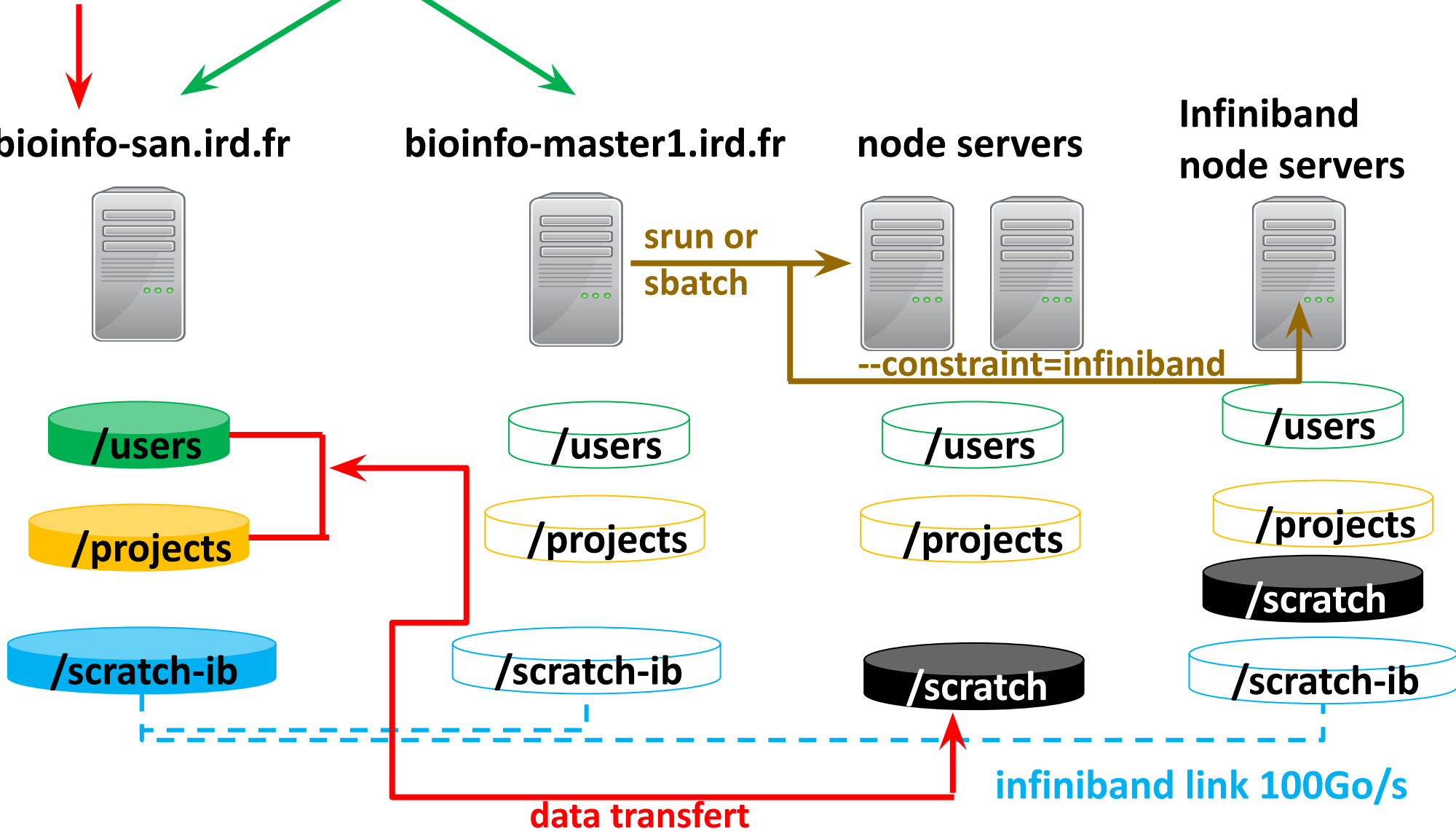
srun or sbatch

--constraint=infiniband



infiniband link 100Go/s

data transfert





# LANCER UN JOB



# Avantages

- Le scheduler choisit les ressources automatiquement
- Lancer des jobs utilisant jusqu'à **100 coeurs**
- Possibilité de paramétrer ce choix
- Jobs lancés en arrière plan
  - possibilité d'éteindre son ordinateur
  - récupération des résultats automatique



# Lancer un job en mode batch

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

```
$~ sbatch 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 slurm avec le mot clé #SBATCH (partie en vert)

```
#!/bin/bash

##### Configuration SLURM#####
## On définit le nom du job
#SBATCH --job-name=test
## On définit le nom du fichier de sortie
#SBATCH --output=res.txt
## On définit le nombre de tâches
#SBATCH --ntasks=1
## On définit le temps limite d'exécution
#SBATCH --time=10:00
#####
```



# Syntaxe des scripts bash

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

```
#####Partie exécution des commandes #####
```

```
nom_variable1="valeur_variable1"  
nom_variable2="valeur_variable2"
```

```
sleep 30  
hostname
```



# Practice

Lancer un script avec sbatch

9

*Aller sur le [Practice9](#) du github*



# Enquête de satisfaction

La réponse à l'enquête suivante est **obligatoire** pour avoir **votre compte prolongé**:

<https://itrop-survey.ird.fr/index.php/537269?lang=fr>



# Citations

Si vous utilisez les ressources du plateau i-Trop.

Merci de nous citer avec:

“The authors acknowledge the ISO 9001 certified IRD i-Trop HPC (South Green Platform) at IRD montpellier for providing HPC resources that have contributed to the research results reported within this paper.

URL: <https://bioinfo.ird.fr/>- <http://www.southgreen.fr>  
”



# Projets

- Pensez à inclure un budget ressources de calcul dans vos réponses à projets
- Besoin en disques dur, renouvellement de machines etc...
- Devis disponibles
- Contactez [bioinfo@ird.fr](mailto:bioinfo@ird.fr) : aide, définition de besoins, devis...



# Merci pour votre attention !



Le matériel pédagogique utilisé pour ces enseignements est mis à disposition selon les termes de la licence Creative Commons Attribution - Pas d'Utilisation Commerciale - Partage dans les Mêmes Conditions (BY-NC-SA) 4.0 International:

<http://creativecommons.org/licenses/by-nc-sa/4.0/>