

Johannes/ALMA

R. Moreno (Lesia)

Généralités

- Information générale sur la machine johannes : <http://dio.obspm.fr/Calcul/posttraitement/>
-
- 1) partition générales
- | partition | temps max | nb coeurs max | mémoire max |
|-----------|-----------|---------------|-------------|
| alma | 15 jours | 12 coeurs | 289 Go |
| orage | 15 jours | 12 coeurs | 289 Go |
| wibar | 15 jours | 12 coeurs | 367 Go |
| pub | 15 jours | 12 coeurs | 524 Go |
-
- | partition | temps max | nb coeurs max | mémoire max |
|-----------|-----------|---------------|-------------|
| low | 15 jours | 36 coeurs | 131 Go |
-
-
- Cette partition de moindre priorité (low) permet d'utiliser plus de coeurs pendant que les partitions dédiées aux projets sont partiellement utilisées. Les jobs soumis sur cette partition peuvent être suspendus pour permettre aux jobs soumis sur les partitions projets de démarrer immédiatement.
-
- Attention : en soumettant un job, il faut tenir compte de la prochaine réservation (partition=all voir ci-après) car un job ne peut démarrer que s'il se termine avant le début de la réservation.

Partition=ALL

- Attention : en soumettant un job, il faut tenir compte de la prochaine réservation (partition=all voir ci-après) car un job ne peut démarrer que s'il se termine avant le début de la réservation.
-
- Pour vérifier les périodes de réservation utilisé la commande :
 - scontrol show reservation
 -
- 2) partition ALL
-
- la partition=all permet à 1 job d'utilisé jusqu'à 83% de la machine johannes
- la partition=urg permet à plusieurs job d'utilisé les 17% restant pendant cette période de reservation=all.
-
-

partition	temps max	nb coeurs max	mémoire max
all	2.5 jours	40 coeurs	1224 Go
urg	2.5 jours	8 coeurs	245 Go

Commandes des jobs SLURM (I)

- 1- Commandes d'information
-
- `squeue` : suivi d'un job
- `scancel jobid` : destruction d'un job de numéro jobid
- `sprio --long` : pending job priority
- `scontrol show job` : liste détaillé des jobs
- `scontrol show reservation` : liste des périodes de réservation des week-ends
- `scontrol show partition` : liste détaillé des partitions
- `sacct` : récapitulatif des jobs sur une période
-
- e.g.
- `sacct -S 2019-01-01 --format=jobid%5 --format=JobName%10 --format=user%9 --format=partition%5 --format=state%10 --format=AllocCPUS%3 --format=ReqMem%10 --format=CPUTime%12 --format=TotalCPU%12 --format=CPUTimeRAW%9 --format=Priority%5 --format=MaxVMSize%11 --format=end%19 -a -u moreno`

Commandes des jobs SLURM (I)

- 2-Soumission de jobs de type batch (sbatch)
-
- Un modèle simple de script exemple.slurm :
- `#!/bin/bash`
- `#SBATCH --job-name=MonProg {Nom du programme}`
- `#SBATCH --partition=pub {La file d'attente}`
- `#SBATCH --time=1-23:00:00 {Le temps sous la forme J-HH:MM:SS}`
- `#SBATCH -n 12 {Le nombre de coeurs}`
- `#SBATCH --mem=1gb {La mémoire demandée}`
- `./nom_de_l_executable {Le programme a lancer}`
-
- Le script est à soumettre avec la commande :
- `$ sbatch exemple.slurm`
-
-

Commandes des jobs SLURM (III)

- 3-Soumission de jobs interactifs (srun)
-
- Pour la réservation de ressources pour travailler de manière interactive il faut utiliser la commande srun :
-
- `$ srun -p [partition] -t [tempsMinute] -J [NomJob] --mem=[memoire] --pty
bash`
-
-

.bashrc :

PATH de CASA/GILDAS possible

- # CASA
- alias casapy540='/shared/apps/casa/casa-release-5.4.0-70.el6/bin/casa' #
2019/02/22 #
- #
- # GILDAS
- export GAG_ROOT_DIR=/shared/apps/gildas/gildas-mar19b/gildas-exe-mar19b # openmp
- export GAG_EXEC_SYSTEM=x86_64-debian9-ifort-qopenmp
- source \$GAG_ROOT_DIR/etc/bash_profile
-
- # SLURM (interactif)
- PS1='\u@\h\${SLURM_JOBID:+:\${SLURM_JOBID}}:\w\\$ '
-
-
- ===== module load ... =====
- **module load casa/5.4.0-70**
- **module load gildas/mar19b**
-

exemple batch SLURM pour CASA

Les comptes utilisateurs d'ALMA doivent lancer leur batch slurm dans la partition=alma

-
- voir ci-joint les 2 exemples pour la réduction avec CASA et l'imagerie avec GILDAS
-
- 1- Réduction/calibration des données avec CASA
- (durée typique du traitement sur Johannes ~0.5-10H/SB, en général < 1 semaine par projet)
-
- `$ sbatch casa_scriptForPI.slurm`
-
- `$less casa_scriptForPI.slurm`
- `#!/bin/bash`
- `#SBATCH --job-name=casa_scriptForPI_alma`
- `#SBATCH --partition=alma`
- `#SBATCH --time=0-24:00:00`
- `#SBATCH -n 1`
- `#SBATCH --mem=24gb`
- `./casa_scriptForPI.com`
-
- `$less casa_scriptForPI.com`
- `/shared/apps/casa/casa-release-5.4.0-70.el6/bin/casa --nogui --log2term -c scriptForPI.py`

example batch SLURM pour GILDAS

- 2- Imagerie avec GILDAS, en général < 4 heures par image
-
- `$ sbatch mapping_clean_map.slurm`
-
- `$less mapping_clean_map.slurm`
- `#!/bin/bash`
- `#SBATCH --job-name=mapping_clean_map_alma`
- `#SBATCH --partition=alma`
- `#SBATCH --time=0-4:00:00`
- `#SBATCH -n 1`
- `#SBATCH --mem=24gb`
- `./mapping_clean_map.com`
-
- `$less mapping_clean_map.com`
- `/shared/apps/gildas/gildas-mar19b/gildas-exe-mar19b/x86_64-debian9-ifort-qopenmp/binmapping @mapping_clean_map.mapping`

Jobs interactifs

- 3- Exemple de jobs interactifs
-
- Voici deux exemples pour 1 heure d'utilisation interactive
- (t = 60 minutes)
-
- `alias srun1h='srun -t 60 --partition=alma --cpus-per-task=1 --mem=24gb --pty bash' # 1h - standard casa/gildas`

Optimisation du nombre de cpu et mémoire

- Après de nombreux test, dans la plupart des cas avec CASA ou GILDAS
- 2 modes générales d'utilisation sont identifié "standard" et "largemap".
- Ils dépendent de la memoire virtuelle (VM=MaxVMSize), que l'on peut estimé avec la commande "sacct" :
- ```
sacct -S 2019-01-01 --format=jobid%5 --format=JobName%10 --format=user%9 --format=partition%5 --format=state%10 --format=AllocCPUS%3 --format=ReqMem%10 --format=CPUTime%12 --format=Priority%5 --format=MaxVMSize%11 --format=end%19 -a -u moreno
```
- 
- - "standard" (VM<24 Gb): ncpu=1, mem=24Gb : e.g. CLEAN image jusqu'a 1024x1024\*1024ch
- - "largemap" (VM>32 Gb): ncpu=int(VM/32)+1 , mem = ncpu\*32 > VM
- 
- **L'utilisation "standard" de 1C/24 Gb de mémoire est suffisant dans la plupart des cas et permet un maximum d'utilisateurs dans la partition=alma .**
- **Lorsque des largemap sont nécessaires utilisez 2C/48 Gb ; 3C/72Gb**
- ...

# Optimisation du nombre de cpu et mémoire

- Dans la plupart des cas,
- le CLEAN de GILDAS est 4 fois plus rapide que celui de CASA
- 
- Le gain maximum de temps avec l'imagerie parallélisé de :
- -GILDAS est d'un facteur 1.1/1.3 pour ncpu=2/4
- -CASA est d'un facteur 1.5/2 pour ncpu=2/4
- 
- exemple de la durée d'un CLEAN AVEC SUPPORT dans Gildas
- - 256x256\*256 < 1 minutes
- - 512x512\*512 < 16 minutes
- -1024x1024\*1024 < 4 heures
- 
- **En résumé, l'utilisation "standard" (cpu=1 mem=24Gb) est optimale elle permet jusqu'à 12 utilisateurs/sbatch simultanées dans la partition=alma (mem max = 288 Gb)**

# Casa version ?

- Dans le directory « qa »
- [file:///home/moreno/project/alma/data/636/qa/pipeline-20180529T093641/html/t2-1.html?sidebar=sidebar\\_uid\\_\\_\\_A002\\_Xcdd033\\_Xcdcf\\_ms&subpage=t2-1\\_details.html](file:///home/moreno/project/alma/data/636/qa/pipeline-20180529T093641/html/t2-1.html?sidebar=sidebar_uid___A002_Xcdd033_Xcdcf_ms&subpage=t2-1_details.html)
-