THALES
Building a future we can all trust

FORUM HPC XIII 28/11/2023

UTILISATION DE SLURM SUR LE HPC DU CNES

www.thalesgroup.com

Présentation de SLURM

- **Qu'est-ce que SLURM ?**
  - **Simple Linux Utility for Resource Management**
  - **Logiciel d'ordonnancement des jobs de calcul**
  - **Objectifs :**
    - *Gérer l'exécution des jobs en fonction des ressources demandées par rapport aux ressources disponibles*
    - *Maximiser l'occupation du cluster et la repartition équitable des ressources*
    - *Mécanisme de suivi de jobs, enchaînement, etc.*

Ressources = CPUs, mémoire, durée (walltime), GPUs.

- **Pourquoi avoir choisi SLURM ? Le logiciel est opensource et très répandu dans le monde HPC.**

SLURM – Les accounts

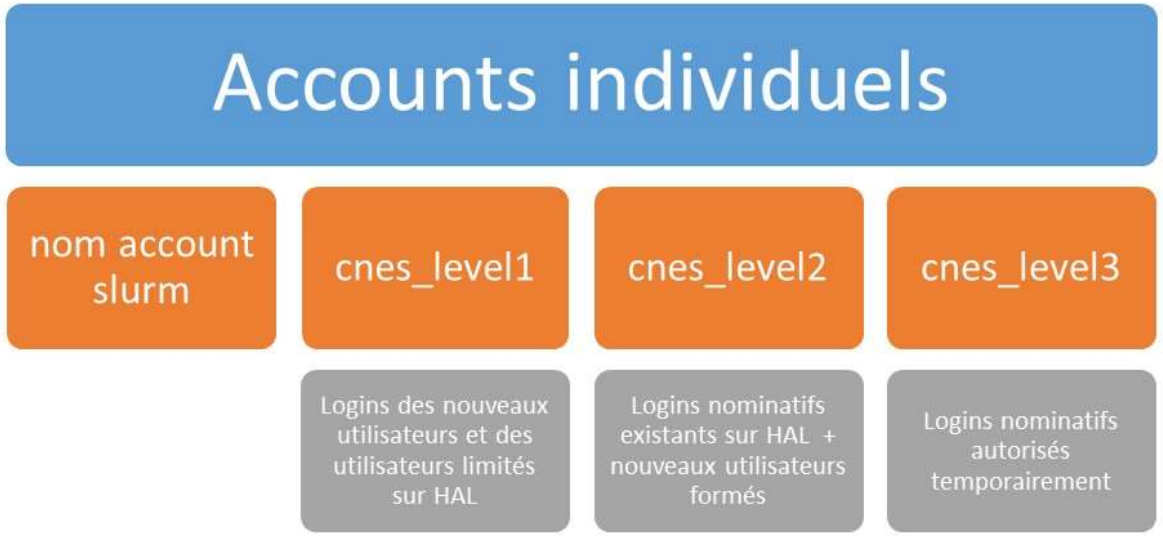
- **Account** : groupe SLURM gérant l'utilisation du cluster (accès aux ressources, priorités, décompte des heures de calcul)
- **Un account est associé à un ou plusieurs groupes unix.**
- Il y a des accounts individuels **cnes_level[123]** et des accounts projets.
- **myaccounts** : liste les accounts de l'utilisateur.

```
loginxx@trexvisu $ myaccounts
 User         Account        Def Acct   MaxJobs MaxSubmit
-------------- -------------- -------------- ------- --------
-
loginxx      cnes_level2     cnes_level2     500      5000
loginxx      projet1         cnes_level2    1000     10000
loginxx      projet2         cnes_level2    1000     20000
```

Le paramètre --account=<account_name> est obligatoire pour lancer un job sur TREX

SLURM – Les accounts individuels

- Uniquement pour les **comptes nominatif**.
  - ***cnes_level1*** : nouveaux utilisateurs (100 jobs run + 1000 jobs soumis)
  - ***cnes_level2*** : utilisateurs ayant participés aux formations (500 jobs run + 5000 jobs soumis)
  - ***cnes_level3*** : utilisateurs autorisés temporairement (1000 jobs run + 5000 jobs soumis)
- Non destiné à un usage intensif. Favoriser les accounts projets.
- En principe, moins de ressources disponibles qu'un account projet.



SLURM – Les accounts projets

- **Account projet** : associé à des groupes unix
- **Accounts projets dev** : accès à des ressources mutualisées
- **Accounts projets oper** (financement nécessaire) : ressources réservées et priorités élevées.
- **Utiliser un account projet pour faire du calcul intensif !**

SLURM – Gestion des accounts projets

- **ATTENTION : Si aucun des groupes unix d'un compte technique n'est associé à un account projet, le compte technique n'a pas d'account.**

```
$ ssh -X mon_compte_technique@trex.sis.cnes.fr
"salloc: error: Job submit/allocate failed: Invalid account or
account/partition combination specified"
```

- **Il existe une demande MVN de gestion des accounts projets pour associer un groupe unix à un nouvel account ou un account existant :**

https://mavienumerique.cnes.fr/s/Catalogue?CODE=CALCUL_SLURM

SLURM – Les commandes

Commandes	Utilisation	Exemple
sbatch	Soumettre un job	sbatch monscript.slurm
squeue	Supervision des jobs	squeue -u <monlogin>
scancel	Supprimer un job	scancel <monjobid>
srun	Soumettre un job interactif	srun -A <monaccount> -N 1 -n 1 --mem=5G --pty /bin/bash
scontrol	Connaître le statut détaillé d'un job en cours	scontrol show job <monjobid>
sacct	Connaître des infos détaillés d'un job	sacct -j <monjobid>
infonodes	Connaître l'occupation du cluster	infonodes (script CNES)
myaccounts	Connaître l'ensemble de ses accounts	myaccounts (script CNES)

- **Page du wiki HPC sur les commandes SLURM :**

https://gitlab.cnes.fr/hpc/wikiHPC/-/wikis/commandes-slurm-trex

SLURM – Quelques options SBATCH

Options	Signification	Commentaire
-J / --job-name	Nom du job	
-N / --nodes	Nombre de nœuds	> 1 pour les jobs //
-n / --ntasks	Nombre de tâches/cœurs	128 max par nœuds g2022
--mem-per-cpu	Mémoire par cœur	8000mb/cœur pour les g2022
--mem	Mémoire totale	1024G/nœud g2022
-t / --time	Durée (walltime)	30 jours max
--gres=gpu:<ngpus>	Nombre de gpus	2 gpus max/personne
-A / --account	Nom de l'account	**OBLIGATOIRE**
-C / --constraint	Type de nœud (nœud rh7 g2019)	Option CNES
-o / --output	Fichier de sortie	%j = jobid
-e / --error	Fichier d'erreur	

- **Faire la commande « `man sbatch` » pour connaître les différentes options de SLURM.**

SLURM – Exemple de job

- Options SLURM dans un script : directive **#SBATCH**

```
#!/bin/bash                      # shell
#SLURM options #SBATCH           # comments
#SBATCH -J monJob                # SLURM option : job name
#SBATCH -N 1                     # SLURM option : total number of computes nodes
#SBATCH -n 1                     # SLURM option : total number of tasks
#SBATCH -t 00:10:00              # SLURM option : time limit
#SBATCH --account cnes_level1    # account  : launch myaccounts to list your accounts
#SBATCH --export=none             # export=none to start the job with a clean environnement and source of ~/.bashrc

cd $SLURM_SUBMIT_DIR             # SLURM_SUBMIT_DIR = directory where sbatch command has been launched
./hello                          # launch programm
```

- **Par défaut, la commande sbatch propage toutes les variables d'environnement aux jobs lancés.**
- **L'option `--export=none` modifie ce comportement. On retrouve le comportement de PBS sur HAL.**

SLURM – Les jobs interactifs

- **Job interactif : l'utilisateur dispose d'un terminal sur le nœud d'exécution.**
  L'option **`--x11`** met en place le X11 forwarding. La commande **« `exit` »** ferme le job interactif.

```
# commande préliminaire pour permettre le lancement d'un job interactif
trexvisu $ unset SLURM_JOB_ID

# lancement du job avec srun
trexvisu $ srun -N 1 -n 8 --time=01:00:00 --mem=64G --x11 --account=<account_name> --pty /bin/bash

# ici le noeud trex069 a été attribué -  exit pour sortir du job interactif
trex069 $ exit
```

- **Job interactif sur un nœud g2019** : utiliser l'option `--constraint/-C 2019`

```
# lancement du job avec srun
trexvisu $ srun -N 1 -n 8 --time=01:00:00 --mem=64G --x11 --account=<account_name> -C 2019 --pty /bin/bash
```

SLURM – Quelques types de job

- **Job array** : Lance en parallèle N jobs de calcul avec des données différentes.
- **Job avec dépendance** : Chaîne des jobs les uns par rapport aux autres.
- **Job parallèle** (*multiprocessing*) : Lance un job avec plusieurs process en parallèle sur plusieurs cpus et nœuds. (OpenMPI, Intel-oneapi-mpi, mpi4py, dask…)
- **Job multithreadé** : Lance un job avec plusieurs threads sur une tâche SLURM. (OpenMP, Intel-oneapi-mkl, pthread,…)

SLURM – Démo basé sur un TP de la sensibilisation HPC niveau 2

```
# chargement du module formations
trexvisu $ module load formations
# création du répertoire de travail du tp2 dans le scratch/data utilisateur
trexvisu $ tp2
trexvisu $ cd /work/scratch/data/$USER/tp_slurm_level2
# pour la demo on se place dans le tp du job parallèle openmpi
trexvisu $ cd job_openmpi
# on analyse le script slurm et on lance le job avec sbatch
trexvisu $ sbatch job_openmpi.slurm
Submitted batch job 2472467
# analyse du déroulement du job avec la commande squeue
trexvisu $ squeue -u bouchae
 JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
           2472467   cpu2022 jobopenm  bouchae  R       0:33      2 trex[030-031]
# pendant la durée du job on peut se connecter en ssh sur les nœuds de calcul
trexvisu $ ssh trex030
trex030 $ top
trexvisu $ ssh trex031
trex031 $ top
# à la fin du job on regarde le fichier de sortie de SLURM
trexvisu $ cat slurm-2472467.out
```

Merci

www.thalesgroup.com