De SGE vers SLURM

SYSTÈME DE SOUMSSION DE TÂCHES

Contenu

- Rôle d’un système de soumission de tâches
- État actuel de SGE
- Pourquoi changer?
- Terminologie SGE versus SLURM
- Utilisation
- Module
- Plus de rapidité
- Description des ressources

Rôle d’un système de soumission de tâches

- Allocation de ressources
  - CPU
  - Mémoire
  - Temps
- Gestion de l’exécution des tâches
  - Démarrage, exécution et surveillance des limites
- Gestion d’une file d’attente
  - Manque de ressources
  - Décision de la priorité des tâches en attente
  - Permet d’éviter une surcharge

Rôle d’un système de soumission de tâches

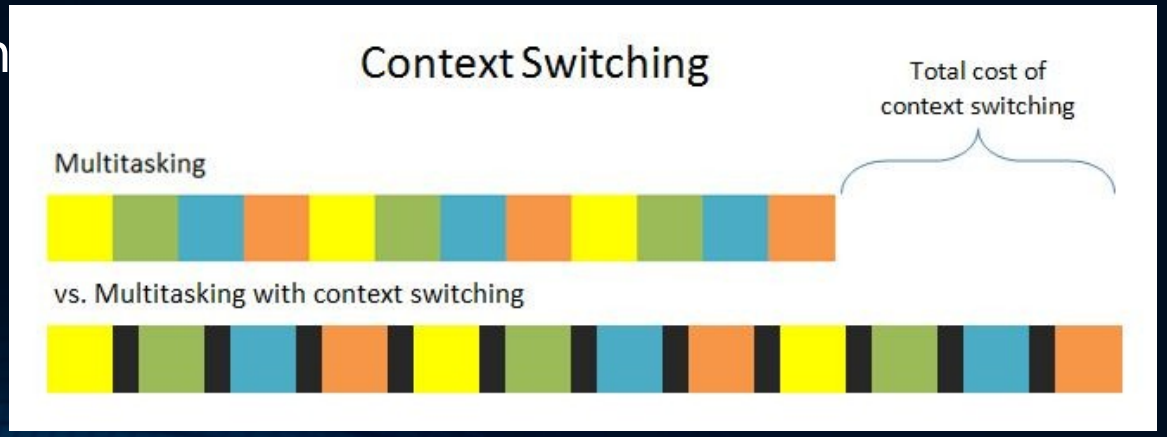
- Éviter la surcharge

  Systèmes d’exploitation récents = multitâches

  Ceci induit la commutation de contexte

  Pour chaque commutation de contexte

  - Sauver la tâche en cours
  - La placer dans la queue
  - Calculer quelle tâche sera la prochaine
  - Restaurer l’ancienne tâch



Rôle d’un système de soumission de tâches

État actuel de SGE

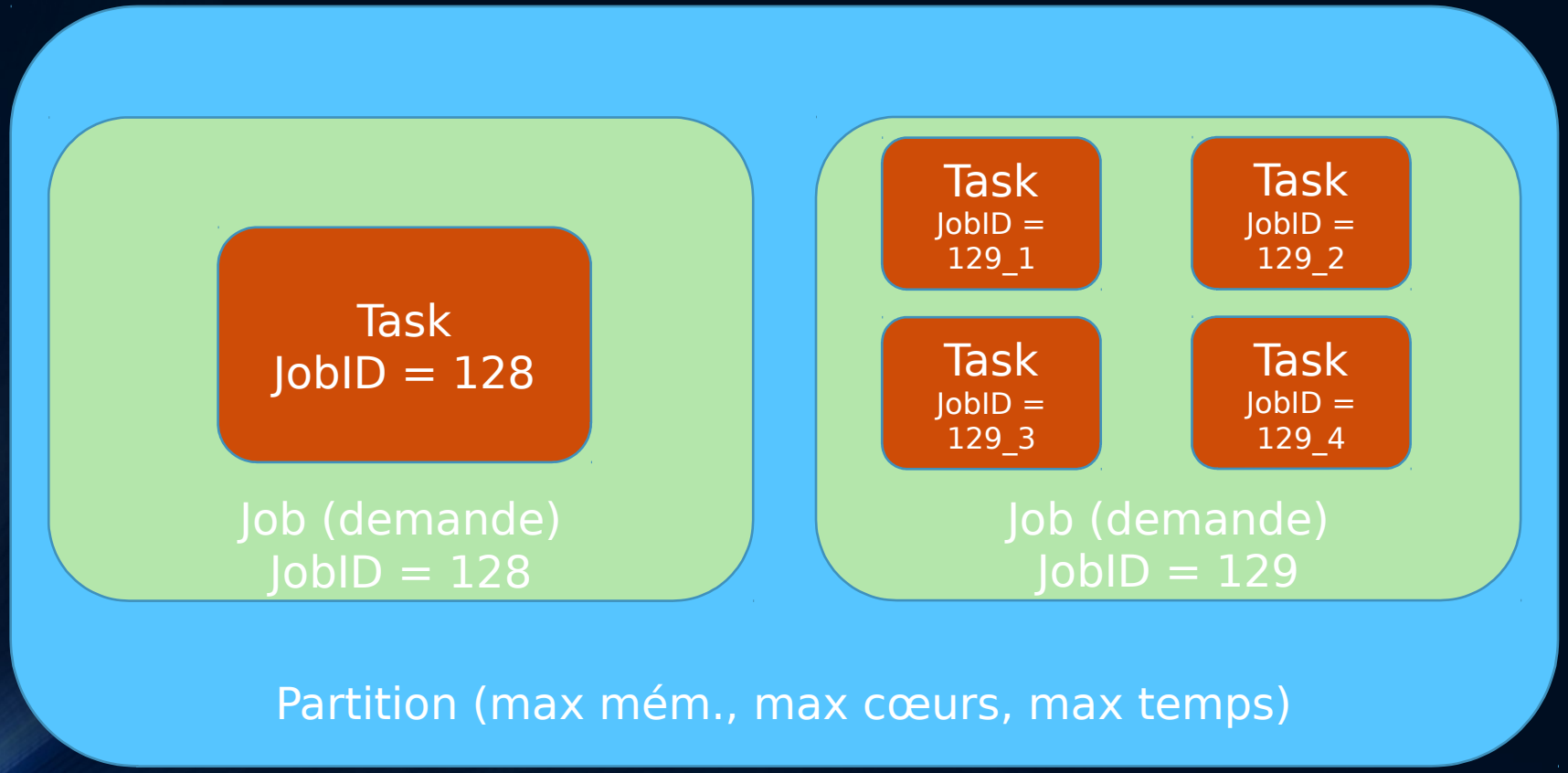
- SGE = Sun Grid Engine
- Sur katak, version Open donc Open Grid Engine
- Début du projet Open en 2009 à partir du code de SGE
- Dernière version 2011.11 p1 annoncé en mai 2012
- Depuis plus rien...

Pourquoi changer?

- Est-ce que j’ai dit que SGE datait de mai 2012…
- Avantages de SLURM pour l’utilisateur
  - Possibilité de modifier la durée d’une tâche en cours
  - Messages plus intuitifs
  - Chaque tâche est maintenant enregistré dans une base de données. Possibilité de consulter toutes les tâches complétées, l’utilisation totale de temps CPU, de mémoire, etc…
- Avantages de SLURM pour l’administrateur
  - Possibilité de redémarrer SLURM sans perdre les calculs en cours
  - Gestion des permissions plus élaborée
  - Intégration de Manitou
  - Vrai contrôle des CPU

Terminologie SGE versus SLURM

- SGE = queue alors que SLURM = partition



Terminologie SGE versus SLURM

Commandes	SGE	Slurm
Soumettre une tâche	qsub [script]	sbatch [script]
Arrêter une tâche	qdel [job_id]	scancel [job_id]
Statut d'une tâche	qstat-j [job_id]	squeue [job_id]
Statut des tâches d'un utilisateur	qstat -u [user]	squeue -u [user]
Statut des noeuds	qhost	sinfo
Environnement graphique	qmon	sview
Soumission		
Directives de scripts	#$	#SBATCH
Queue (partition en Slurm)	-q [queue]	-p [partition]
Coeurs	-pe smp [coeurs]	-c [coeurs]
Temps horloge	-l h_rt=[secondes]	-t [min] -t [minutes:secondes] -t [heures:minutes:secondes] -t [jours-heures] -t [jours-heures:minutes] -t [jours-heures:minutes:secondes]
Mémoire	-l h_vmem=[mémoire]	--mem=[mémoire en MB]
Nom de la tâche	-N [nom]	--job-name=[nom]

Utilisation

- Partitions
  - ibismini 21-00:00:00 cpu=10,mem=102400 cpu=60
  - ibisinter 21-00:00:00 cpu=20,mem=307200 cpu=60
  - ibismax 21-00:00:00 cpu=40,mem=819200 cpu=60
- Valeurs par défaut si vous ne demandez rien
  - 00-24:00:00 cpu=1,mem=10240
  - Ibismini
- Pour l’accès aux autres partitions ou toute demande spéciale, contactez-nous!

Utilisation

- Soumettre une tâche en interactif
  - srun --pty bash
  - Pour debugger
  - Pour compiler
  - Pour une application en mode graphique
  - Penser à se déconnecter

```
steph@katak:~> squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
steph@katak:~> srun --pty bash
steph@katak:~> # On fait des trucs
steph@katak:~> squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
              1597  ibismini     bash    steph  R       0:26      1 katak
steph@katak:~> exit
exit
steph@katak:~> squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
steph@katak:~>
```

Utilisation

- Soumettre une tâche en batch
  - sbatch votreScript (dans /etc/skel/SLURM_example.sh sur katak)

```
#!/bin/bash


#SBATCH -D /project/
#SBATCH -J Name_Job
#SBATCH -o Name_output-%j.out
#SBATCH -c 1
#SBATCH -p ibismini
#SBATCH --mail-type=ALL
#SBATCH --mail-user=type_your_mail@ulaval.ca
#SBATCH --time=1-00:00
#SBATCH --mem=51200

# Load the software with module if applicable:
# module load python/3.5

# Type your command line here
$PRG/blastp -db $BANK/nr -query 1433_pea.fasta -evalue 1e-5 -num_threads 2 -out 1433_pea_vs_nr.blastp
```

Utilisation

- Soumettre une tâche en batch
  - Exemple avec un « ls »

```
steph@katak:~/test-slurm> sbatch test.slurm
Submitted batch job 1601
steph@katak:~/test-slurm> squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
              1601  ibismini           TestLS    steph  R       0:08      1      katak
steph@katak:~/test-slurm> ls
02_A5assembly.sh  slurm-1601.out  slurm.batch  test.slurm
steph@katak:~/test-slurm> cat slurm-1601.out
02_A5assembly.sh
slurm-1601.out
slurm.batch
test.slurm
steph@katak:~/test-slurm>
```

Utilisation

- Problèmes possibles
  - Demander trop de ressources:
    - sbatch: error: Batch job submission failed: Job violates accounting/QOS policy (job submit limit, user's size and/or time limits)
  - Demander une partition sans y avoir droit:

```
     JOBID PARTITION     NAME     USER  ST       TIME  NODES NODELIST(REASON)
         1603    ibisinter          TestLS    steph  PD       0:00      1 (AccountNotAllowed)
```

  - Plus assez de coeurs (2 tâches de 10 coeurs)

```
     JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
         1607  ibismini      TestLS     steph PD       0:00      1 (QOSMaxCpuPerUserLimit)
       1606  ibismini        TestLS     steph  R        0:07      1 katak
```

Module (en préparation)

- Permet de gérer efficacement les logiciels disponibles ainsi que leurs versions
- Initialise l’environnement du logiciel

```
steph@katak:~> module avail

beagle/3.3.2          emboss/6.5.7          MaSuRCA/3.1.3
ngopt/a5/20150522     samtools/1.3          stacks/1.37

steph@katak:~> module load samtools/1.3

steph@katak:~> module list

Currently Loaded Modulefiles:

  1) samtools/1.3

steph@katak:~> module unload samtools/1.3
```

Plus de rapidité

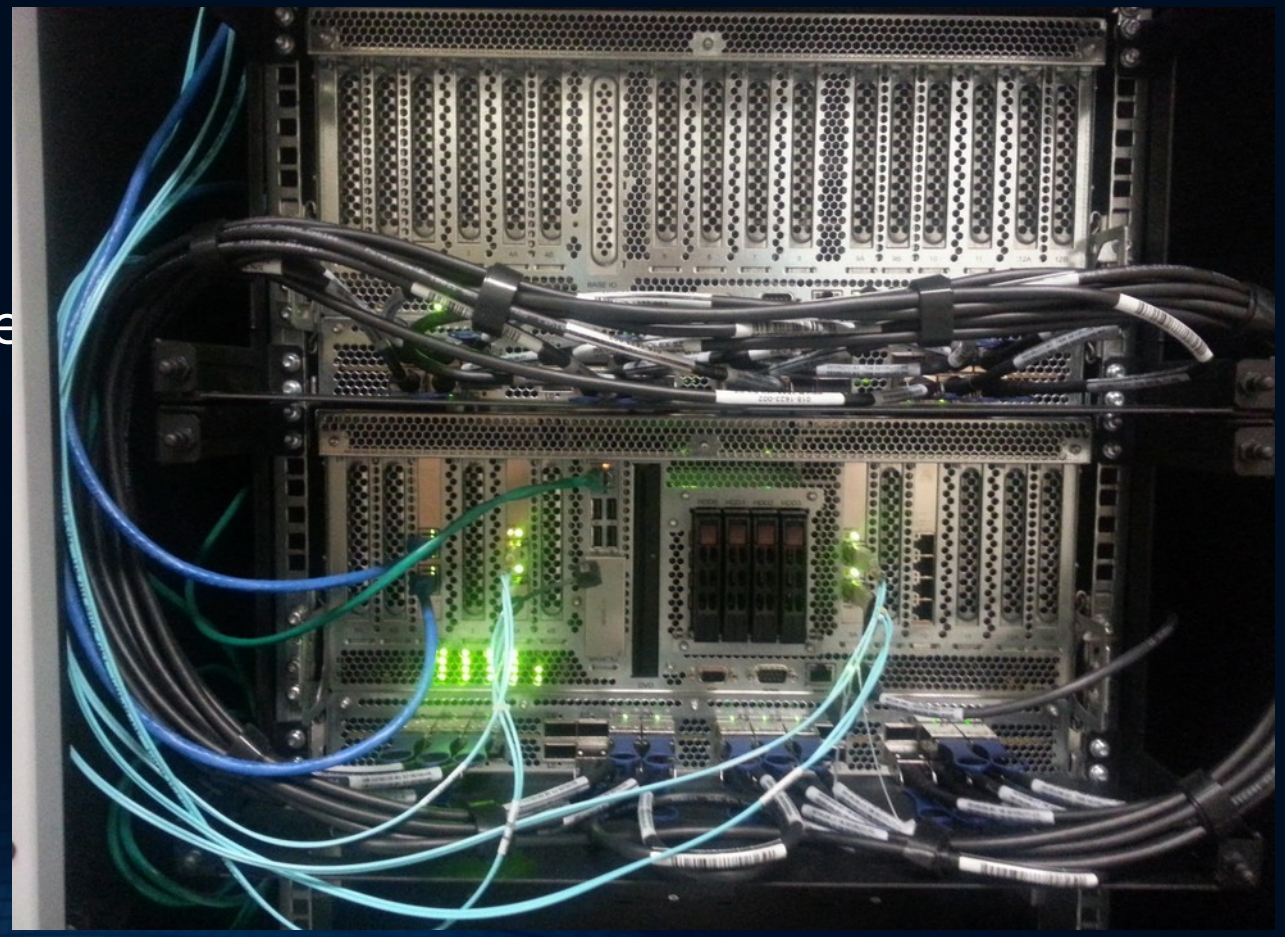
- Compilateur Intel
  - Par défaut, tout est compilé avec GCC
  - 2 licences flottantes d’Intel sont disponibles
- Partition /scratch (manitou seulement – bientôt sur katak)
  - Pour les calculs nécessitant beaucoup d’écritures sur les disques
  - 5 fois plus rapide en écriture que dans vos comptes ou partitions
  - Aucune protection des données

Description des ressources

- Katak
  - 64 coeurs (2 pour le système, 2 pour les connexions, 60 pour le calcul)
  - 1 To de mémoire
  - 2 liens fibre 8 Gbps vers les disques
  - 2 liens 10 Gbps Ether vers Manitou

Description des ressources

- Manitou
  - 128 coeurs (pour le calcul)
  - 2 To de mémoire
  - 2 liens fibre 8 Gbps vers les disques
  - 2 liens 10 Gbps Ethe vers Katak



Description des ressources

- Disques
  - 24 disques de 3 TB
  - 60 disques de 4 TB
  - Total 312 TB

Description des ressources

- Disques
  - Ce jeudi…
    plus de disques

Description des ressources

- Partage des disques entre katak et manitou
  - Katak a présentement un accès direct aux disques
  - Manitou passe par katak pour avoir accès aux disques (protocole NFS)
  - NFS = lenteur + point unique de défaillance
  - Dans un proche avenir
    - Installation de GFS2 (Global File System 2)
    - Permet à plusieurs serveurs d’avoir accès directement aux disques
    - Mécanisme d’accès concurrent
    - Plus rapide que NFS
    - Évite le point unique de défaillance

Prochaine présentation

- Mardi prochain le 31 mai
- Exemples en direct de l’utilisation de SLURM
- Scripts plus complexes
- Comment utiliser les autres partitions et comptes
- Comment savoir à quelles partitions vous avez droit
- Voir les limites de chaque partition
- Consulter le journal de vos tâches (mémoire, CPU, temps, etc…)

Fin!