Centre de calcul de l'uB

Formation

Présentation et utilisation du cluster de Calcul

Antoine Migeon

ccub@u-bourgogne.fr

Le Centre de Calcul de l'uB (ccub)

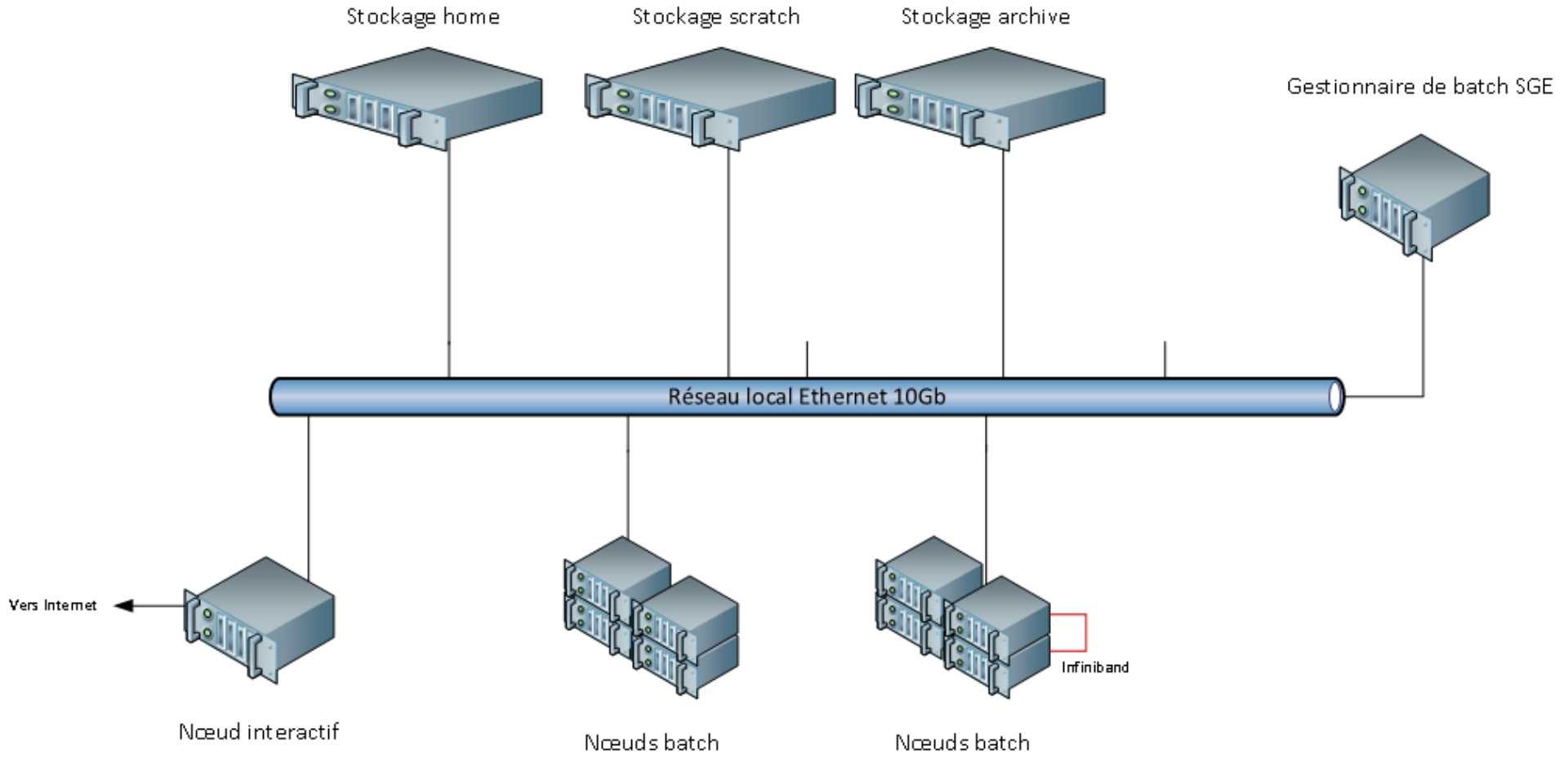
- Dédié à l'enseignement et à la recherche.
- Met à disposition des étudiants et des chercheurs des ressources informatiques pour le calcul numérique intensif :
  - Cluster linux
  - Espaces de stockage
  - Sauvegardes
  - Logiciels scientifiques
  - Compilateurs et bibliothèques
  - Messagerie de l'uB
- Géré par une équipe de 5 personnes (directeur : Didier Rebeix).
- Nous contacter : ccub@u-bourgogne.fr
- Plus d'infos : http://www.u-bourgogne.fr/dnum-ccub

Présentation du cluster de calcul

- Cluster linux
  - Machines interactives
  - Machines batch
  - Espaces de stockages
- Logiciels scientifiques
- Compilateurs et bibliothèques

Le cluster linux

- ~ 300 serveurs calcul
- ~ 7300 cœurs (cores)
- ~ 32 To de RAM
- 3 systèmes de stockage
- ~ 4000 To d’espace disque
- Interconnexions réseaux :
  - 1Gbit/s Ethernet, 10Gbit/s Ethernet, 25Gbit/s Ethernet
  - 56 Gbit/s et 100 Gbit/s Infiniband
  - 100 Gbit/s Omni-Path
- 700 Teraflop/s crête estimés (~ 100Gflop/s pour un PC)
- ~ 100 KWatt d'électricité, + les climatisations


Stockage home
Stockage scratch
Stockage archive
Gestionnaire de batch SGE
Réseau local Ethernet 10Gb
Vers Internet
Infiniband
Nœud interactif
Nœuds batch
Nœuds batch

Les machines

Le cluster se compose de 2 grands types de machines :

- Machines interactives
- Machines batch

Machines interactives

- **krenek**2002 + **krenek**2003
  - 16 cœurs
  - 256 Go RAM
  - GPU Nvidia

- **Connexion directe**

- Convient pour :
  - Travail interactif direct : matlab –jvm, vmd, sas, etc.
  - Graphisme, pre et post traitement
  - Développement des codes, tests
  - Soumission des jobs batchs

Machines batch

- **webern** (machines spéciales ou séquentielles)
  - Entre 8 et 192 cœurs, avec ou sans GPU
  - 4Go/cœur RAM (sauf webern05 et webern11 big memory 32Go/cœur)
- **part => 32 cœurs AMD**
  - 4Go/cœur RAM (128Go)
  - Interconnexion Infiniband 100Gbit/s (parallélisme)
- **davis et hauer => 16 cœurs**
  - 4Go/cœur RAM (64Go)
  - Interconnexion Infiniband 56 Gbits/s (parallélisme)
- **bartok => 24 cœurs**
  - 4Go/cœur RAM (96Go)
  - Interconnexion Omni-Path 100 Gbits/s (parallélisme)
- **Pas de connexion directe**
- **Gérées par SGE**

Espaces de stockages

- 3 espaces de stockage partagés avec les nœuds de calcul
  - homedir et scratch accessibles par tous les nœuds
  - archive accessible par les nœuds interactifs uniquement

- => les nœuds voient les mêmes données au même moment
- quotas par groupe ou par utilisateur
- Le CCUB ne supprime jamais les données des utilisateurs (sauf dans /tmp, /tmp2 et /tmp3)

Espaces de stockages

- /**user1**/$GROUP/$USER
  - Faible volumétrie (20 To)
  - Longue durée
  - Très Sécurisé (RAID)
  - Sauvegardé (3 ans)
  - Double contrôleur
  - Quotas par groupe

Home directory = pas de calcul dedans

Commande pour voir les quotas : *espgroup*

Espaces de stockages

- /**archive**/$GROUP/$USER
  - Grande volumétrie (3000 Tio total)
  - Longue durée
  - Sécurisé (RAID)
  - Non sauvegardé (mais dupliqué sur bande)
  - Rétention de 2 mois après suppression
  - Quotas groupes et utilisateurs
  - $ARCHIVEDIR == /archive/$GROUP/$USER
  - Archive = pas de calcul dedans
  - Non disponible sur les nœuds batch

Espaces de stockages

- /**work**/$GROUP/$USER
  - Volumétrie moyenne (1000 Tio)
  - Faible durée (réservé à l'exécution des travaux)
  - Très performant
  - Sécurisé (RAID)
  - Non sauvegardé
  - Quotas utilisateurs
  - $WORKDIR == /work/$GROUP/$USER

  - Scratch = fait pour le calcul

Espaces de stockages

- Espaces partagés : shared
  - Il est possible de demander la création d'un répertoire partagé pour l'ensemble des utilisateurs d'un groupe
    - /user1/shared/$GROUP
    - /work/shared/$GROUP
    - /archive/shared/$GROUP

Seul le propriétaire d'un fichier peut supprimer le fichier

Logiciels scientifiques

- Chimie-Physique : méthode ab initio
- Matlab, traitement du signal, traitement d'images
- Éléments finis
- Calcul formel
- Génomique
- Climatologie
- Outils graphiques
- Analyse de données
- Bureautique
- **Demandez au CCUB pour faire installer vos logiciels**

Compilateurs et bibliothèques

- fortran 77, fortran 90 et extensions pour le parallélisme, Portland et GNU (OpenMP, MPI)
- Langages c et c++ et extensions pour le parallélisme.
- Langage java.
- Langage python.
- ada : compilateur du domaine public (GNU).
- prolog : interpréteur du domaine public (SWI-Prolog Amsterdam).
- Bibliothèques scientifiques : gsl, acml, cernlib, matlab, blas, FFT..
- Bibliothèques et compilateurs payants **Intel**, Portland, etc.

Utilisation du cluster de calcul

- Connexion, accès aux ressources
- Gestion de l'environnement : les modules
- Compilation
- Concept : mémoire partagée ou distribuée
- Le gestionnaire de job batch : SGE
- Exemple de soumissions de jobs

Connexion au cluster de calcul

- SSH : ssh $USER@ssh-ccub.u-bourgogne.fr
- **NX** : www.u-bourgogne.fr/dnum-ccub/spip.php?article961
  - **NoMachine** = Affichage graphique performant sur connexion ADSL
- VNC : www.u-bourgogne.fr/dnum-ccub/spip.php?article270
  - Affichage graphique 3D sur connexion très haut débit
- XDMCP ou XMING : déconseillé, peu sécurisé et peu performant

Les modules

- lister les logiciels spécifiques installés
- Configurer l'environnement
  - Variables $PATH , $LD_LIBRARY_PATH …
- Afficher une documentation succincte

- https://www.u-bourgogne.fr/dnum-ccub/spip.php?article392

Les modules

- module avail
- module help R/3.5.1
- module load R/3.5.1
- module list
- module unload R

Compilations

- Utiliser de préférence le compilateur Intel
  - icc –axCORE-AVX512 -O3 -fp-model precise -pc 64 -o code.exe code.c

    - -axCORE-AVX512 : compile certaines parties du code de façon optimisées pour chaque génération de processeur Intel jusqu'à l'AVX512
    - -O3 : optimisation pour la vitesse d'exécution
    - -fp-model precise : améliore la précision pour les calculs flottants
    - -pc 64 : compilation en double précision

- https://www.u-bourgogne.fr/dnum-ccub/spip.php?article787

2 modèles de programmation

- **Séquentiel**
  - 1 job, 1 processus, 1 cœur consommé
- **Parallèle**
  - 1 job, N processus, N cœurs, **X serveurs** (N>1 et X>=1)
  - 2 types de programmation parallèle :
    - Mémoire partagée : SMP
    - Mémoire distribuée : DMP

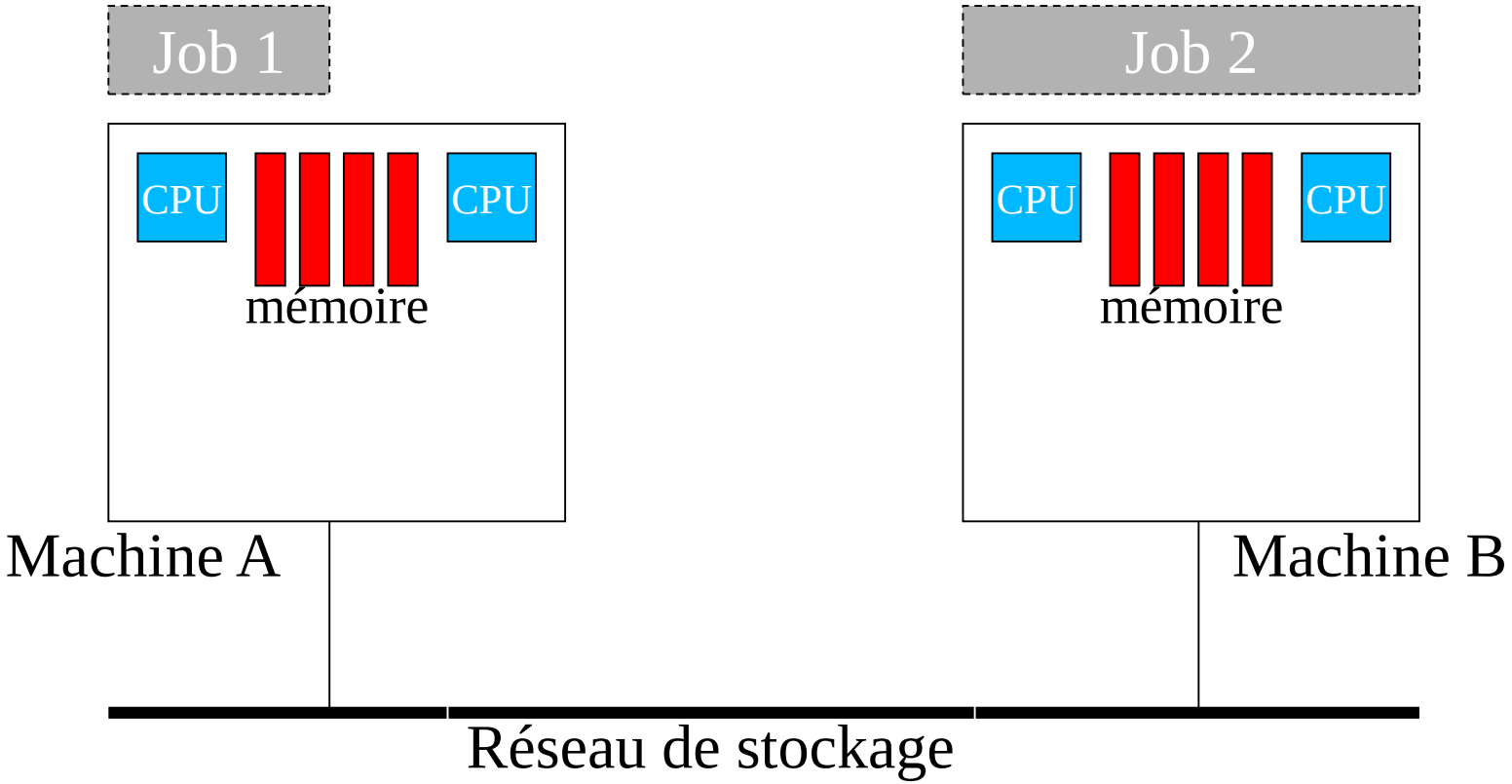
Mémoire partagée SMP

- **SMP** : Shared Memory Processing
  - 1 job, N processus, 1 serveur
  - Utilise uniquement la mémoire et les cœurs locaux du serveur

- Ex : OpenMP, gaussian, …
- Pas besoin de réseau d'interconnexion machine rapide

Mémoire partagée SMP

Un programme est contenu dans une machine et n'accède pas aux ressources d'une autre machine.



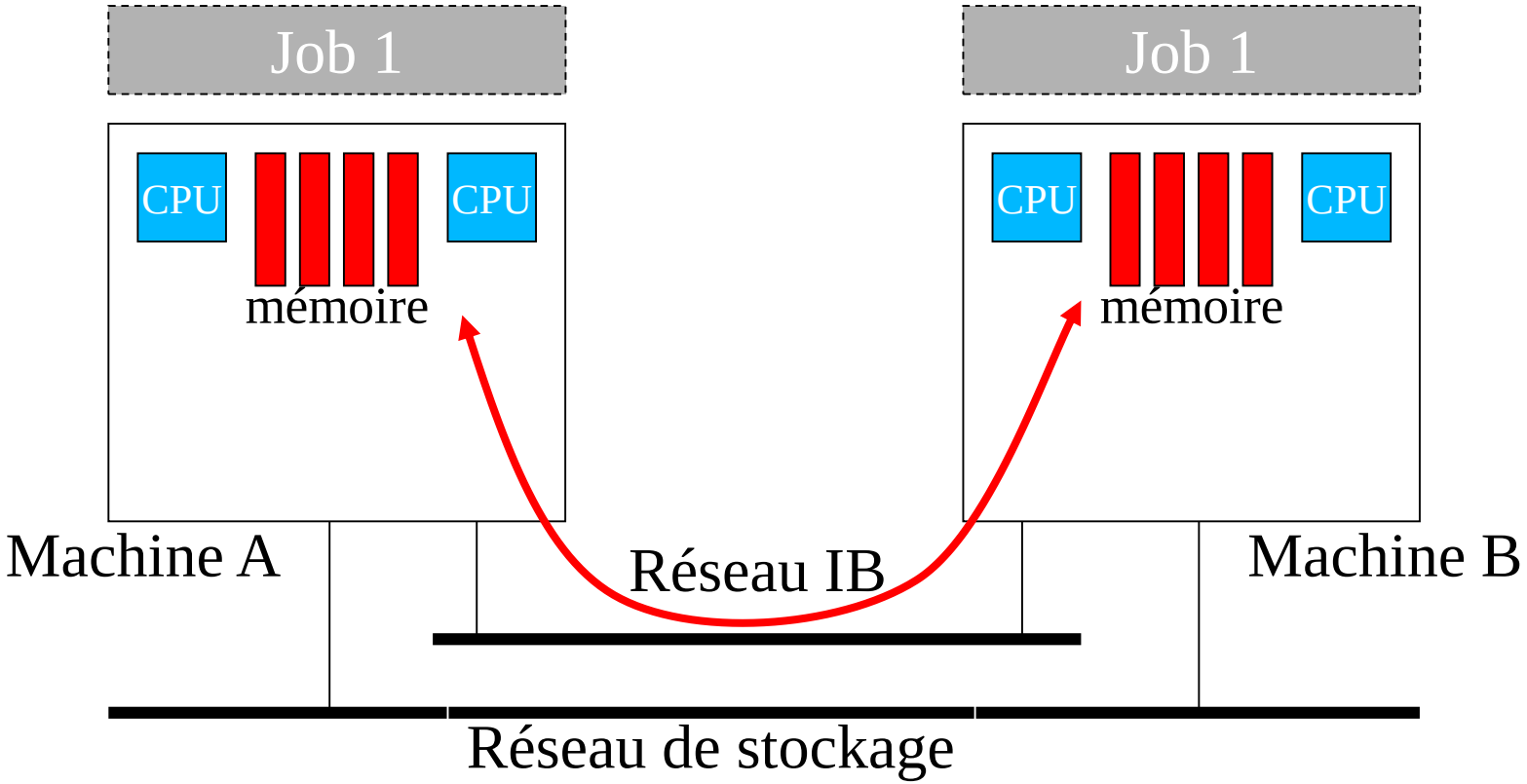
Mémoire distribuée

- **DMP** : Distributed Memory Processing
  - 1 job, N processus, X serveurs
  - Utilise la mémoire et les cœurs de X serveurs

- Ex : **MPI**, WRF, Vasp, …
- Nécessite un réseau d'interconnexion machine rapide (Infiniband 10Gbit/s mini et une très faible latence) pour obtenir de bonnes performances.

Mémoire distribuée DMP

Un programme est exécuté sur plusieurs machines et échange des informations à travers le réseau haute performance.



SGE (Sun Grid Engine)

- Répartit sur les machines batch les jobs soumis à partir des machines interactives.
- Répartition équitable des ressources (processeur) entre les utilisateurs. (+arbitrage manuel fait par l'équipe du ccub)
- Gestion des files d'attentes
- Travail en mode déconnecté
- Notifications par mails (ex : fin d'exécution)
- Management des jobs (soumission, suivi, kill, suspend, restart, …)
- Statistiques

Différence entre slot et cœur

- Le nombre de core ou cœur d'un processeur dépend de son architecture physique
- Le nombre de slot déclaré sur une machine dépend de la configuration du gestionnaire de batch SGE

- On déclare généralement autant de slot qu'il y a de cœurs sur une machine
- Si on souhaite attribuer plus de mémoire à chaque jobs, il suffit de déclarer moins de slots
- Déclarer plus de slots qu'il n'y a de cœurs sur une machine reviendrait à surcharger la machine

Les files d’attentes

– **batch** (+ 6000 slots)
  - file générique (jobs SMP ou DMP)
  - Limitée à 600 slots/user
  - Divisée en environnements parallèles : 1 par switch Infiniband ou Omni-Path
  - Par défaut sur les machines Intel,
  - ajouter `-l vendor=amd` pour partir sur les AMD

  - batchbm (16 slots)
  - Big Memory : pour les jobs SMP qui ont besoin de beaucoup de mémoire (>4Go/coeurs)
  - Limitée à 4 slots/user

Les files d’attentes

- spécifiques :
  - gpu (12 slots bi GPU) :
    - Dédiée à la visualisation graphique haute performance et au deep learning
    - Limitée à 2 slots/user
  - uv (192 slots sur une seule machine) :
    - Pour les gros jobs SMP (2 To de RAM)
    - Limitée à 64 slots/user

Les files d’attentes

- transfer (12 slots) :
  - Dédiée aux tranferts de données
  - Limitée à 4 slots/user
  - S’exécute sur les machines interactives qui sont pourvues du système de stockage **/archive et /work**
  - Permet de créer des jobs dédiés aux transferts de données entre le /work et le /archive

1/ transferBeforeJob1 => cp ou rsync /archive vers /work

2/ job1 => calcul

3/ transferAfterJob1 => cp ou rsync /work vers /archive

```
qsub -q transfer -N transferBeforeJob1
qsub -q batch    -N job1               -hold_jid transferBeforeJob1
qsub -q transfer -N transferAfterJob1  -hold_jid job1
```

Soumission d’un job

- Séquentiel
  - qsub –q batch *mon_script*
  - qsub –q batch –pe smp 1 *mon_script*
  - qsub –q batchbm –pe smp 1 *mon_script*

  - SGE configure l’environnement sur une machine batch et exécute le script mon_script (variable $NSLOTS)

Soumission d’un job

- Parallèle :
  - SMP :
    - qsub –q batch –pe smp NSLOTS *mon_script*
  - DMP
    - qsub –q batch –pe dmp* NSLOTS *mon_script*

NSLOTS = nombre de slots

Soumission d'un job

qsub –q batch *mon_script*

mon_script :

```
#!/bin/ksh
#$ -q batch -pe dmp* 32
#$ -M antoine.migeon@u-bourgogne.fr

echo 'Bonjour Dijon'
mpirun mon_programme mon_jeu_de_donnée
```

Fichier de sortie d’un job

- Contient les sorties STDOUT et STDERR
- Emplacement : même emplacement que lors de la soumission qsub
- Nom : script_de soumission.o$jobid ou script_de soumission.po$jobid (parallèle)

- Exemple : *job_script_1.sh.o317609*

Suivi des jobs

- Lister ses job :
  - qstat
- Lister tous les jobs
  - qstat –u ‘*’
- Obtenir les infos sur un job particulier
  - qstat –j *jobid*
- Supprimer un job
  - qdel *jobid*

Etats des jobs

- Rr : running
- r : running
- qw : queue wait, pas de place actuellement pour le job
- Rq : limite de slots atteinte

Etat du cluster

- Commande qhost

- https://krenek2000.u-bourgogne.fr/clustermap

Documentation

- man sge
- man qsub
- man qstat
- module help *nom_module*

- http://www.u-bourgogne.fr/dnum-ccub