Introduction à l'utilisation d'Occigen

Vous avez obtenu des heures de calcul sur un supercalculateur national ! Ce petit guide va vous permettre de démarrer en quelques minutes. Avant de commencer, il vous faut votre login et votre mot de passe Occigen. Si vous n'avez pas encore votre mot de passe, appelez le service svp au 04.67.14.14.99.

Connexion à Occigen

- Si vous êtes sous Windows, lancez un client SSH (putty, kitty) pour vous connecter à l'adresse d'Occigen :

  `occigen.cines.fr`

  Dans les paramètres de Putty : `Category/SSH/X11` cochez la case 'Enable X11 forwarding' afin d'activer le renvoi des fenêtres graphiques.

- Si vous êtes sous Linux, ouvrez un terminal, et connectez-vous à Occigen :

  `ssh –X login@occigen.cines.fr`

Vous arrivez dans votre repertoire `/home/login` sur un nœud de connexion. Sur Occigen il y a plusieurs nœuds de connexion qui servent à compiler le programme, à le soumettre sur les nœuds de calcul et à accéder aux fichiers de résultats.

L'environnement d'Occigen

Sur Occigen, il y a un système de modules. Le chargement d'un module permet de modifier ou de positionner des variables d'environnement (PATH, LIBRARY_PATH etc…). Pour voir les modules chargés :

```
login@occigen:~$ module list

      1) No Modulefiles Currently
```

Aucun module n'est chargé par défaut.

Pour voir tous les modules disponibles sur Occigen, tapez :

```
login@occigen:~$ module avail
```

Cette commande permet de voir les logiciels et les bibliothèques installés : Gromacs, Abinit, MKL, Netcdf, Hdf5, etc.
Nous vous conseillons de charger les modules suivants pour compiler et exécuter votre code :

```
login@occigen:~$ module load intel

login@occigen:~$ module load openmpi
```

Le module `intel` contient les compilateurs et les outils Intel dans une version récente.
Le module `openmpi` contient la bibliothèque MPI optimisée pour les communications entre les nœuds d'Occigen.

Ces modules sont bien entendu appelé à évoluer au fil du temps et des mises à jour. Pensez à vérifier à intervalle régulier les modules disponibles.

Pour observer les variables placées par le module `intel/18.0`:

```
login@occigen:~$ module show intel/18.0
```

Transfert de votre code de votre ordinateur personnel ou votre laboratoire vers Occigen

- Sous windows : utilisez une solution SFTP telle que Filezilla
- Sous linux :

```
login@occigen:~$ scp -r code_directory <votre_login>@occigen.cines.fr:~
```

Compilation de votre code

Pour compiler, l'espace approprié est le `/home`.

Placez-vous dans l'espace `cd $HOMEDIR/code_directory`.

Créez un Makefile pour votre code (si ce n'est pas déjà fait), puis compilez avec la commande **`make`**.

Exemple de Makefile avec option de compilation –O2 :

```
# création de l'exécutable 'Programme'
CC=mpicc
CFLAGS=-O2
all: main.o fonctions.o
	$(CC) $(CFLAGS) main.o fonctions.o -o Programme
main.o: main.c fonctions.h
	$(CC)  $(CFLAGS) -c main.c -o main.o
fonctions.o: fonctions.c fonctions.h
	$(CC)  $(CFLAGS) -c fonctions.c -o fonctions.o
# suppression des fichiers temporaires
clean:
	rm -rf *.o
```

Pour plus d'informations sur les Makefile nous vous invitons à rechercher la documentaion sur Internet

Exécution du code

L'exécution d'un programme se fait sur des nœuds de calcul grâce à un système batch. Pour lancer des travaux, on se place dans l'espace `/scratch ($SCRATCHDIR)`. Celui-ci **est le plus efficace** pour les entrées/sorties pendant l'exécution de votre job. Créez donc un répertoire (nous le nommerons ici `run_directory`) dans `$SCRATCHDIR` avec vos fichiers d'entrée.

Création d'un fichier SLURM

Pour exécuter votre code sur les nœuds de calcul, il vous faut écrire un fichier de soumission SLURM (par exemple `job.slurm`). SLURM est le gestionnaire de travaux d'Occigen. Dans le fichier de soumission on renseigne les ressources requises par notre job :

- le nombre de nœuds
- le nombre de cœurs
- la taille mémoire
- le nombre de processus MPI
- le temps d'exécution maximum prévu.

Certaines informations doivent obligatoirement être positionnées dans votre script. D'autres, au contraire, ne doivent pas apparaître. Par exemple, les paramètres suivants ne doivent pas être positionnés :

```
#SBATCH -p all

#SBATCH --qos normal
```

En cas de non respect des paramètres, votre soumission sera rejetée. Vous n'obtiendrez pas de message expliquant ce rejet, la version actuelle de SLURM ne nous permet pas de vous renvoyer cette information (mais les versions futures corrigeront ce défaut).

Voici un exemple pour lancer un calcul standard sur 2 nœuds, avec 24 taches par nœud sur une durée de 30 minutes :

```
login@occigen:$SCRATCHDIR/run_directory$ cat job.slurm

#!/bin/bash
#SBATCH -J job_name
#SBATCH --nodes=2
#SBATCH --ntasks=48
#SBATCH --ntasks-per-node=24
#SBATCH --threads-per-core=1

#SBATCH --mem=1GB
#SBATCH --time=00:30:00
#SBATCH --output job_name.output



module purge
module load intel/17.2
module load openmpi/2.0.2

srun --mpi=pmi2 -K1 --resv-ports -n $SLURM_NTASKS ./mon_executable param1
param2 …
```

De base **l'environnement du nœud de login** (`module load …`) **est propagé sur les nœuds de calcul**.

Si vous souhaitez avoir la maîtrise totale de vos environnements durant le calcul, pensez à utiliser la commande `module purge`, avant d' ajouter les modules dont vous avez besoin.

Soumission sur les nœuds de calcul

Afin de soumettre votre job, faire :

```
login@occigen:$SCRATCHDIR/run_directory$ sbatch job.slurm

2924824.occigen
```

Le gestionnaire de travaux SLURM vous donne un numéro de job (`job_id`), ici **`2924824`**.

Pour surveiller l'état de votre job :

```
login@occigen~$ squeue –u <votre_login>

      ou

login@occigen~$ squeue –j <job_id>
```

Si votre job est en statut '**Q**' ou '**PD**', c'est qu'il est en attente de libération de ressources, s'il est en statut '**R**' c'est que son exécution a démarré.

Vous pouvez aussi arrêter votre job comme suit :

```
login@occigen:~$ scancel <job_id>
```

Lorsque l'exécution du programme est terminée, vous obtenez un fichier de sortie ainsi que les fichiers résultat produits par votre programme dans votre répertoire de soumission.

Performance

Optimisation de la compilation pour Occigen

Voici quelques options de compilation simples à mettre en œuvre pour optimiser votre exécutable.

Ajoutez les options `-O3 -xHost` dans les options de compilation de votre Makefile. Vous pouvez recompiler et relancer votre job afin de voir si ces options ont eu un impact sur le temps d'exécution de votre programme et aussi de vérifier si les résultats scientifiques sont toujours corrects.

Sauvegarde des données à conserver et nettoyage du /scratch

Votre programme a tourné, et vous avez obtenu des résultats, bravo ! Cependant le travail n'est pas fini : en effet **le `/scratch` n'est pas sauvegardé** et en cas d'incident technique tous les fichiers présents sur cet espace peuvent être effacés.

Rapatriement vers un espace d'Occigen sauvegardé

La première possibilité pour conserver les données qui vous intéressent est de copier ou déplacer les fichiers en question vers un espace sauvegardé sur Occigen cela se situe dans le `/store`. Pour les stocker, nous vous conseillons de créer un fichier compressé au format tar (`<fichier>.tar.gz` par exemple) car le nombre de fichiers est limité sur cet espace.

Rapatriement vers votre laboratoire

La deuxième option est de copier les fichiers vers votre laboratoire avec la commande scp lancée depuis la machine de votre laboratoire :

```
me@myworkstation:~$ scp -r <votre_login>@occigen.cines.fr:
$SCRATCHDIR/repertoire_a_rapatrier repertoire_dans_votre_labo
```

Nettoyage du /scratch

Pensez à régulièrement supprimer les fichiers inutiles sur le `/scratch` afin que cet espace puisse continuer à fonctionner dans les meilleures conditions pour l'ensemble de nos utilisateurs.