TD MPI 2 : Cluster gin – Communications non-bloquantes et collectives

L'objectif de cette séance est d'utiliser un cluster avec gestionnaire de tâches, et de tester des communications point-à-point non-bloquantes et collectives de la librairie MPI. Les exercices seront réalisés sur le cluster gin de l'ENSTA.

Exercice 1. Utilisation de gin et du gestionnaire de tâches

(a) Chaque étudiant reçoit un identifiant et un mot de passe pour le cluster gin. Testez-les en vous connectant par `ssh`. Ensuite, copiez les codes que vous avez développé lors du TD 1 de la station de travail à gin en utilisant la commande `scp`.

(b) Compilez et exécutez le programme `helloworld_mpi.c` sur la frontale de gin.

(c) On fournit un script[1] (`submit.sh`) pour soumettre un job au gestionnaire de tâches de gin. Utilisez ce script pour exécutez le programme `helloworld_mpi.c` sur l'un des nœuds de calcul de gin (`qsub submit.sh`). Analysez les options du script, ainsi que les fichiers générés après l'exécution du programme. En particulier, identifiez comment modifier le nombre de processus MPI.

(d) Lorsque plus de 16 processus MPI sont demandés, ceux-ci sont répartis sur plusieurs nœuds de calcul. Testez le programme `helloworld_mpi.c` en ajoutant l'option `-display-map` après `mpirun` pour obtenir de l'information du placement des processus. Croisez avec l'information obtenue grâce à la fonction `MPI_Get_processor_name`.

Ensuite, testez les options `-bynode` et `-byslot` en combinaison avec `-display-map`. Quel est l'effet de ces options sur le placement des processus MPI lorsqu'il y en a plus que 16 ?

Exercice 2. Communications point-à-point non-bloquantes

(a) Reprenez le code utilisé lors de l'exercice 2(a) du TD 1 *(communications point-à-point bloquantes)*, et exécutez-le sur gin en utilisant le gestionnaire de tâches. Testez les quatres cas[2] considérés lors du TD 1 pour des tableaux avec 10000 éléments. Obtenez-vous les mêmes résultats qu'au TP1 ?

(b) Pour les cas ou vous observez un *deadlock*, utilisez les fonctions de communication point-à-point non-bloquantes `MPI_Isend` et `MPI_Irecv` pour résoudre le problème. Testez et vérifiez vos solutions sur gin.

Exercice 3. Communications collectives et étude de scalabilité

(a) Reprenez le code développé lors de l'exercice 3 du TD 1 *(méthode du trapèze)*. Modifiez le code afin d'utiliser une fonction de communication collective à la place des fonctions de communication point-à-point. Quelle fonction est la plus adaptée ?

(b) Exécutez le code sur gin en utilisant le gestionnaire de tâches. Pour $N = 10^9$, analysez le temps de calcul que vous obtenez en fonction du nombre de processus MPI *(testez 1, 2, 4, 8, 16, 32 et 64 processus)*.

(c) Idem que (b), mais avec $N = n_{\text{proc}} \cdot 10^7$, où n_{proc} est le nombre de processus.

(d) En comparant les résultats de (b) et (c), que pouvez-vous dire de la scalabilité du programme ?

[1] Téléchargement en ligne de commande : `wget http://perso.ensta-paristech.fr/~modave/AMS301/codesMPI2.tar`

[2] Pour les cas menant à un *deadlock*, vous devrez éventuellement tuer votre job en utilisant la commande `qdel` ... très utile lorsqu'on apprend à utiliser un cluster.