

8INF856

Programmation sur architectures parallèles

Devoir 3

(À remettre au plus tard mercredi le 19 novembre 2013)

- Les étudiants sont invités à travailler en équipe de deux ou trois.
- **Vous serez évalué pour la rectitude et la qualité de vos réponses.**

L'objectif de ce travail est de vous familiariser avec MPI. Je vous recommande de ne pas vous restreindre à ce qui a été vu en classe et de consulter la documentation afin d'expérimenter et d'exploiter au maximum les ressources disponibles.

Vous devez faire ce qui suit

1. Sur `dim-linuxmpi1.uqac.ca` vous devez créer un répertoire dont le nom est `TP3` et dans lequel sera placé un fichier unique dont le nom est `tp3.c` qui contiendra toutes les fonctions décrites ci-après. Le fichier `tp3.c` devra contenir une fonction `main()` dont le contenu est décrit plus loin.
2. Vous devez implémenter une structure de donnée distribuée sur le cluster *dim-linuxmpi*. L'idée générale est d'obtenir un type de tableau dont le contenu est réparti sur les p noeuds ($1 \leq p \leq 6$) participant à l'exécution d'un programme suite à la commande:

```
mpicc -np p prog.exe.
```

L'idée de base est que le tableau est réparti sur les p machines de sorte que cela soit transparent pour l'utilisateur. En particulier vous devez définir les fonctions suivantes:

- `T = TD_init(n)` : Créé l'espace mémoire nécessaire pour contenir n entiers et retourne T une valeur d'un type de donnée approprié (que vous devez déterminer). L'espace pour les n entiers doit être réparti entre les p noeuds de telle façon que les

deux fonctions suivantes soient le plus efficaces possible quel que soit le noeud sur lequel elles sont exécutées.

- `TD_get(T,i, &x)` : Place à l'adresse de la variable x le contenu de la i -ième case du tableau distribué T . Retourne 1 si i est un indice valide, sinon retourne 0.
- `TD_put(T,i,x)` : Place dans la i -ième case du tableau distribué T la valeur x . Retourne 1 si i est un indice valide, sinon retourne 0.
- `TD_somme(T)` : Retourne la somme des éléments de T .
- `TD_trier(T)` : Permet de trier le tableau T en ordre croissant. Cette fonction doit être le plus efficace possible.
- `TD_afficher(T,i,j)` : Affiche dans l'ordre les éléments i à j ($i \leq j$) du tableau T .

3. Lors de l'évaluation je tiendrai compte de l'efficacité de votre solution que je comparerai avec celle des autres équipes.
4. Vous devez me remettre (par courrier électronique) un rapport satisfaisant les critères suivants:
 - (a) Votre rapport doit être en format PDF
 - (b) Vous devez indiquer le nom des coéquipiers
 - (c) Vous devez indiquer le nom d'utilisateur du compte sur lequel le fichier `tp3.c` a été placé.
 - (d) Vous devez expliquer de façon détaillée votre implémentation des tableaux distribués
 - (e) Vous devez expliquer de façon détaillée votre implémentation de chacune des fonctions.
5. Au minimum, votre fonction `main()` doit faire ce qui suit: Lire un entier n en entrée et construire un tableau distribué de n entiers. Le tableau doit être initialisé avec des valeurs entières (possiblement générées de façon aléatoire). La fonction `TD_somme()` est ensuite appelée et le résultat est affiché. Finalement, la fonction `TD_trier()` est appelée et un message est affiché demandant à l'utilisateur d'entrer deux entiers $i \leq j$ pour ensuite appeler la fonction `TD_afficher(T,i,j)`.