

TP - ASR7 Programmation Concurrente

Première utilisation de threads

Septembre 2018

I Début des threads

Vous devez paralléliser un petit programme : `lancement.cpp` sur le site de l'UE. Ce dernier répète un certain nombre de fois un calcul qui n'a pas d'importance. Le but de l'exercice est de savoir si vous pouvez paralléliser les appels à la fonction `fct()`, récupérer le résultat (qui est le temps écoulé pendant le calcul d'après la fonction) et comparer ce temps avec le temps réellement écoulé.

Pour cela :

- Q.I.1)** - Dans la fonction `main()` remplacez la boucle qui appelle `nbthreads` fois la fonction `fct()` par le lancement de `nbthreads` threads qui exécutent la fonction.
- Q.I.2)** - Si cela ne vous est pas déjà arrivé, faites une erreur d'argument entre le lancement du thread et la fonction afin de vous habituer à lire les messages d'erreur de compilation. Par exemple, essayez un passage par référence en oubliant `std::ref()` au moment de l'appel : le message d'erreur est rarement convivial!
- Q.I.3)** - Normalement, dès qu'il y a un nombre de threads important, vous pouvez observer que les messages qu'ils affichent se mélangent. Sachant que les fonctions système sont prévues pour fonctionner en environnement multithread, pourquoi y a-t-il ce mélange et comment l'éviter?
- Q.I.4)** - À la fin des threads obtenez le résultat de la fonction et affichez le temps que chaque thread pense avoir mis pour s'exécuter ainsi que la somme de ces valeurs.

La commande `time` permet de voir les différents temps utilisés par une commande :

```
$ time ./lancement.exxb 20
Th principal : lancement de 10 fois la fonction
0 lancement de la fonction pour 5 itérations
...
Th principal : Le temps total de calcul est 9.87181

real    0m9.876s
user    0m9.863s
sys     0m0.002s
```

- *real* est le temps écoulé pendant le calcul
- *user* est la somme des temps processeur utilisés par les threads en mode utilisateur (pour faire les calculs eux-mêmes)
- *sys* est la somme des temps processeur passés en mode noyau (pour faire les opérations système comme les affichages).

- Q.I.5)** - Faites plusieurs essais avec 2, 3, 4, 8, 10, 20 threads.
 - 5(a)** - Pourquoi le temps utilisateur est-il supérieur au temps réel ?
 - 5(b)** - La somme des temps que vous calculez est-elle liée au temps réel ? Au temps utilisateur ? Au temps système ?
 - 5(c)** - Est-elle identique, et s'il y a une différence pourquoi ?

Pour la dernière question il faut comparer cela avec le nombre de processeurs/cores de l'ordinateur (`cat /proc/cpuinfo`).