

Laboratoire de Programmation Concurrente

semestre printemps 2020

Prise en main des threads

Temps à disposition : 2 périodes (travail débutant en semaine 1)

1 Objectifs pédagogiques

- Se familiariser avec l'environnement de programmation.
- Réaliser son premier programme concurrent en C++.
- Tester les problèmes de concurrence.

2 Cahier des charges

Le programme qui vous est donné effectue une simple incrémentation d'un nombre un certain nombre de fois. Cette incrémentation est faite par plusieurs threads et ce potentiellement en parallèle (ou pseudo-parallèle). Si vous le lancez et le compilez, vous devriez observer des valeurs étranges pour la valeur finale du compteur.

Le dialogue lancé par l'application vous permet de choisir le nombre de threads à lancer et le nombre d'itérations à effectuer. Une pression sur le bouton *Start* lance les threads. Lorsqu'ils ont tous terminé leur travail, la valeur finale du compteur ainsi que le ratio (valeur finale observée/valeur finale attendue) sont affichées.

A partir du code qui vous est donné, tentez quelques expériences pour améliorer la valeur finale observée. Pour ce faire, essayez de mettre en place un mécanisme d'attente active en exploitant des variables partagées par l'ensemble des threads. Utilisez des variables de classe (**static**) pour réaliser cela. Le fichier à modifier est `mythread.cpp`. Vous pouvez y observer l'incrémentation d'un compteur partagé.

Durant vos essais, vous pouvez tenter d'utiliser une barrière mémoire, qui va garantir que tous les load et store présents avant elle sont exécutés avant ceux qui sont présents après :

```
std::atomic_thread_fence(std::memory_order_acq_rel)
```

Ceci permet de synchroniser les mémoires caches, et en le plaçant à des endroits judicieux cela permet de rendre le code plus sûr (bien que moins performant).

Enfin, n'hésitez pas à tester votre code en forçant l'usage d'un seul cœur :

- Sous Linux, en ligne de commande : `taskset 1 ./counter`
- Sous Windows, en ligne de commande : `start /AFFINITY 1 counter.exe`
- Sous MacOS, vous pouvez utiliser l'outil Instruments disponible dans la distribution de XCode, procédure indiquée dans cet article¹.

1. <http://jesperrasmussen.com/2013/03/07/limiting-cpu-cores-on-the-fly-in-os-x/>

3 Développement Qt

Il est fortement recommandé de consulter le guide de démarrage Qt disponible sur [cyberlearn](https://www.cyberlearn.fr/).

4 Travail à rendre

Il s'agit d'une prise en main. Amusez-vous, posez toutes les questions nécessaires et tentez d'améliorer le comportement de votre programme. Le labo ne sera pas évalué.

A la fin de ce laboratoire, rendez votre code selon les consignes de laboratoires disponibles sur le site du cours. Certains codes seront analysés en classe.