

Série de Travaux Dirigés : 4 - Threads C++11

Pour faciliter les exercices, vous disposez sous Celene d'une archive *TD4.tgz* contenant la trame de tous les exercices. Vous trouverez un répertoire par exercice avec le programme principal et les indications du code à compléter.

Pour un premier test, le programme **hello word** (répertoire Hello) est disponible. Vous pouvez le compiler et le tester avec la commande `./hello`.

Exercice 1. Addition de vecteurs (répertoire VecAdd)

Ce répertoire contient le programme **VecAdd** qui prend en argument la taille des vecteurs à additionner et le nombre de threads total à utiliser. Il propose une version séquentielle du calcul dans la fonction **vecadd**.

1. Complétez la fonction `vecadd_parallel` qui effectue le calcul en parallèle.
2. Vérifiez vos résultats.
3. A l'aide de la macro **BENCHMARK** comparez les temps de calcul en utilisant les deux fonctions (pour les tailles de tableaux suivantes : 10, 1000, 100000, 10000000).
4. Quelle est la partie la plus lente du programme ? Parallélisez-là.

Exercice 2. Addition (répertoire Add)

Soit *a* une variable de type `int` définie pour chaque thread. Ecrivez la fonction qui permet de calculer la somme cumulée de toutes les variables *a*. La dernière addition doit être réalisée par le thread principal (le père de tous les autres threads). Modifiez ensuite votre programme pour garantir l'ordre des additions (thread 0 puis thread 1...).

Exercice 3. Conditions - Deadlock (répertoire Deadlock)

Compilez et exécutez ce programme. Qu'observez-vous ? Essayez de corriger l'erreur.

Exercice 4. Tri fusion (répertoire Tri)

À partir du squelette fourni, proposez dans la fonction `tri-fusion`, une version parallèle de ce tri.
