INF7235 : Laboratoire #5
Multiplication parallèle de polynomes avec `OpenMP`

20 février 2017

L'objectif de ce laboratoire est de vous familiariser avec l'utilisation des principales constructions du langage OpenMP/C. Pour obtenir le code source (sur `japet`) :

```
$ git clone http://www.labunix.uqam.ca/~tremblay/git/LaboOpenMP.git
```

Ce que vous devez faire

- Pour ces exercices, vous devez paralléliser deux programmes :

  1. `pi.c` : Approximation de la valeur de π par intégration numérique.
  2. `polynomes.c` : Manipulation de polynomes.

  Ces programmes à paralléliser contiennent déjà certaines instructions `OpenMP`, **mais ne contiennent aucune directive (`pragma`) pour rendre l'exécution parallèle**. Votre travail consiste donc à ajouter de telles directives, et ce dans le but de bien comprendre l'effet des diverses directives.

- Un fichier `makefile` vous est fourni, qui permet de compiler les divers programmes sources et lancer certaines exécutions avec des arguments appropriés — voir le fichier ainsi que plus bas.

1 Programme pour le calcul de π (intégration numérique)

Modifiez le programme `pi.c` pour le rendre parallèle, de façon à qu'il ait une bonne accélération. Plus spécifiquement, voici des suggestions de choses à essayer, puis à évaluer/tester avec divers nombres de *threads particulièrement pour les items* c. *vs.* d. *vs.* e :

1. Introduisez une directive `omp parallel` simple et voyez l'effet qui en résulte.
2. Introduisez une directive `omp for` et voyez l'effet.
3. Introduisez une directive `omp critical` et voyez l'effet.
4. Remplacez la directive `omp critical` par `omp atomic` et voyez l'effet — notamment, sur le temps d'exécution.
5. Modifiez la directive `omp for` en ajoutant une clause `reduction` et voyez l'effet — idem.
6. Supprimez l'instruction «`omp_set_dynamic(0);`» et voyez l'effet, notamment lorsque vous spécifiez (sur la ligne de commande ou via le `makefile`) un **très grand nombre de *threads***!

2 Programme pour la manipulation de polynomes

Le fichier `polynomes.c` contient une mise en oeuvre séquentielle d'opérations sur des polynomes. L'interface du type abstrait est dans le fichier `polynome.h` — il s'agit de la même interface et même mise en oeuvre séquentielle que pour la version TBB (labo #4).

Modifiez la mise en oeuvre pour effectuer de façon parallèle **le produit de polynomes** (opérations `coefficient` et `fois`).

Quelle devrait être la forme de «`schedule`» qu'il serait préférable d'utiliser? Faites diverses expérimentations en exécutant «`make bm`» avec diverses tailles de polynomes et diverses `schedule` et tentez d'identifier la meilleure stratégie de répartition du travail. Expliquez/justifiez!

Quelques remarques :

- Pour compiler : «`make compile`».
- Pour tester que les résultats sont corrects : «`make tests`» — ou modifiez le `makefile` pour définir la cible par défaut (voir le fichier)!
- Pour mesurer les performances : «`make bm`».