Laboratoire 2

Parallélisation d'un calcul sériel

Écrit par Gabriel-Andrew Pollo-Guilbert

Idée original de Francis Giraldeau

Pour helene.jiang@polymtl.ca

Pour francois-philippe.ossim-belias@polymtl.ca

Les systèmes hétérogènes sont des systèmes informatiques utilisant plusieurs types de processeur ou de pièces de matériel spécialisé. Il est souvent idéal d'effectuer certains calculs sur des processeurs spécialisés afin d'augmenter la performance ou l'efficacité du système. Par exemple, le décodage d'une vidéo est souvent effectué dans une carte graphique afin de ne pas affecter le reste du système.

Dans ce laboratoire, vous devez convertir un calcul sériel afin de l'exécuter sur plusieurs coeurs avec OpenMP et aussi sur un périphérique spécialisé, comme une carte graphique, avec OpenCL.

Code

- `source/main.c`
  - Contient le point d'entrée du programme qui traite les arguments en ligne de commande et démarre le calcul.
- `source/color.c` `source/kernel/helpers.cl` `include/color.h` `include/pixel.h
  - Contiennent des structures et des fonctions pour manipuler les couleurs et pixels.
- `source/headless.c` `include/headless.h`
  - Contiennent l'interface non-graphique permettant d'effectuer de le calcul.
- `include/log.h`
  - Contient des macros pour formatter et afficher des logs dans la sortie standard.
- `source/sinoscope.c` `include/sinoscope.h`
  - Contiennent la structure et certains fonctions pour le calcul.
- `source/sinoscope-serial.c`
  - Contient l'implémentation sériel du calcul.
- `source/sinoscope-openmp.c` **(À COMPLÉTER)**
  - Contient l'implémentation avec OpenMP demandée.
- `source/sinoscope-opencl.c` `source/kernel/sinoscope.cl` **(À COMPLÉTER)**
  - Contiennent l'implémentation avec OpenCL demandée.
- `source/viewer.c` `include/viewer.h`

    - Contiennent l'interface graphique basée sur OpenGL pour montrer le dessin.

Calcul

Le calcul effectué par le programme est simple. Celui-ci calcul la valeur d'une fonction pour tous les points d'une matrice 2D. Le calcul de chacun des points individuels est triviallement parallélisable. Lisez le fichier `source/sinoscope-serial.c` pour plus d'information.

Spécifications

Les spécifications décrites dans cette section diffèrent d'une équipe à une autre.

L'implémentation avec OpenMP doit utiliser `omp parallel for` et paralléliser les deux boucles. L'ordonnancement des noeuds doit être statique.

L'implémentation avec OpenCL doit passer en premier paramètre le buffer partagé. Ensuite, le second paramètre est une structure contenant toutes les valeurs entières de `sinoscope_t` suivit de tous les paramètres à virgule flotante un à un. Finalement, la répartition du calcul doit se faire en deux dimensions.

- La compilation ne doit pas lancer d'avertissements.
- Le programme ne doit pas avoir de fuite de mémoire durant son exécution autre que OpenMP et OpenCL.

Compilation

Pour compiler l'application, il est recommandé de créer un dossier `build/` à la racine du projet afin de bien séparer les fichiers générés.

```
$ mkdir build && cd build
```

Ensuite, on configure le projet avec `cmake`. Celui-ci peut donc être compilé avec `make`.

```
$ cmake ..
$ make
```

Il n'est pas nécessaire de re-exécuter toutes les commandes ci-dessus pour recompiler le binaire, seulement la dernière. Vous pouvez exécuter `./sinoscope --help` pour voir les options du programme.

Exécution

Il y a 3 modes d'exécution: avec graphique, sans graphique (`--headless`) ou vérification (`--check`). Les deux premiers modes sont interactifs, les controles suivants sont disponible:

- `[1]` utiliser l'algorithme sériel
- `[2]` utiliser l'algorithme basé sur OpenMP
- `[3]` utiliser l'algorithme basé sur OpenCL
- `[q]` quitter l'application

De plus, le mode graphique offre aussi ces controles:

- `[-]` diminuer le polynôme de Taylor du calcul
- `[+]` augmenter le polynôme de Taylor du calcul

Commandes

Le `Makefile` généré par `cmake` contient les commandes spéciales ci-dessous.

- `make format`
  - Utilise `clang-format` pour formatter le code source.
- `make remise`
  - Crée une archive ZIP contenant les fichiers pour la remise.
- `make check`
  - Exécute `./sinoscope --check` afin de vérifier les calculs.