Optimisation des accès mémoire

L'objectif est de mettre en œuvre des techniques d'optimisation pour éviter les défauts de cache inutiles.

Scrollup

Nous allons modifier le noyau `scrollup` du programme `2Dcomp` afin provoquer de nombreux défauts cache et observer leurs impacts sur les performances.

1. Modifier le fichier `makefile` pour compiler en niveau d'optimisation `-O2`. Cela désactivera la vectorisation automatique et nous permettra de mieux distinguer l'influence du cache sur les performances du programme.
2. Produire une version `scrollup_compute_ji()` en dupliquant la fonction `scrollup_compute_seq()` et en inversant l'ordre des boucles en `i` et `j` :

```
for (int j = 0; j < DIM - 1; j++)
     for (int i = 0; i < DIM; i++)
       next_img (i, j) = cur_img (i + 1, j);
```

3. En utilisant l'image `shibuya.png`, vérifier visuellement le résultat. Comparer « visuellement » la fluidité de l'animation obtenues par les versions `seq` et `ji` du noyau `scrollup`.
4. Utiliser les scripts `expe-scrollup.sh` et `tracer-courbes-log.r` afin d'obtenir un graphique présentant les performances des versions `seq` et `ji` en fonction de la taille de l'image.
5. Interpréter ce graphique sachant qu'un pixel est codé sur 4 octets et que les tailles des caches de la matrice peuvent être obtenues via la commande `lstopo`.

Multiplication de matrices

Le programme `mul_mat.c` effectue une multiplications de matrices de façon classique.

1. Modifier le code de `mulMat2` afin d'utiliser plus efficacement le cache du processeur. Le gain obtenu est-il décevant, correct ou plus que satisfaisant ?
2. Il est probable que quelques défauts de cache évitables subsistent. Les repérez-vous ? Quelle permutation des boucles sur i, j, k induit le plus petit nombre de défauts de cache ? Modifier votre code en conséquence.

Bonus Lorsque N est assez grand il est probable quelques défauts de cache évitables subsistent dans votre code. Supposons que le cache fasse 8 Mo pour quelle valeur de N apparaissent ces défauts de cache ? Quelle technique faudrait-il utiliser pour limiter ces défauts ?

Transposée d'une image

Nous allons étudier l'impact du cache sur les performances du noyau `transpose` du programme `2Dcomp`.

1. Produire une version optimisée du noyaux `transpose` en utilisant la technique de pavage (tuilage). On s'inspirera du code de la fonction `mandel_compute_tiled()` et utilisera la variable `GRAIN`. Vérifier visuellement le bon fonctionnement pour différents grains.
2. En utilisant les scripts `expe-transpose.sh` et `tracer-courbes-log.r`, produire un graphique présentant les performances de la version `tiled` du noyau `transpose` en fonction du `GRAIN` utilisé.
3. Analyser le graphique obtenu.
4. Produire une version parallèle de chaque code (`omp` et `omp_tiled`, comparer les performances obtenues pour une image de taille `4096`.