

FFT parallèle et application au filtrage de fréquences

HES-SO//Genève, hepia

ITI - orientation logiciel & systèmes complexes

Filtrage d'une image

La transformée de Fourier 2D appliquée à une image, produit une matrice de coefficients à valeur complexe. Si comme dans notre cas les données de départ sont réelles, les données produites sont en partie redondantes. La figure 1 montre comment distinguer entre les hautes et les basses fréquences dans le spectre.



FIGURE 1 – Filtre pour basses ou hautes fréquences

Les 4 coins correspondent aux basses fréquences. Si on les annule, on produit une image qui contient des hautes fréquences, donc qui met en évidence les bords des objets.

1 Implémentation

Dans un premier temps, votre travail consistera à implémenter avec MPI, l'algorithme de la FFT parallèle 1D pour un nombre de processus $p \leq n$ où n est le nombre total de données (p et n sont une puissance de 2). On vous met à disposition deux squelettes de code dans les fichiers `fftSeq.cc` et `fftPar1D.cc`. Par rapport à ces squelettes, vous ne devez rien changer en ce qui concerne ni les entrées-sorties, ni la génération aléatoire des données. Ainsi, pour le rendu, les seuls affichages seront ceux déjà présents dans les squelettes (les deux appels aux fonctions d'impression). L'initialisation du générateur aléatoire et le choix du nombre d'éléments par processeur se fera via les arguments passés en ligne de commande. Par exemple, le lancement de la FFT parallèle sur 16 processeurs avec une graine de 19 pour le générateur aléatoire et 8 valeurs par processeur se fera selon la syntaxe : `mpirun -np 16 fftPar1D 19 8`

Voici les étapes à suivre pour produire le programme parallèle `fftPar1D.cc` :

- Ecrivez un code séquentiel `fftSeq.cc` en vous basant sur le squelette fourni.
- Testez les résultats de votre code séquentiel en les comparant avec ceux produits par Octave/Matlab.
- Parallélisez la fonction `bitInv()` qui procède au réordonnement par inversion binaire.
- Parallélisez la fonction `fft()`.
- Vérifiez les résultats du programme parallèle avec ceux du programme séquentiel ou de Octave/Matlab.

Remarques

- L’emploi de Octave/Matlab peut faciliter votre travail.
- Les fonctions `main()`, `randomInit()` et `printAll()` n’ont en principe pas besoin d’être modifiées.
- Faites des tests pour vérifier que vos implémentations fonctionnent correctement.
- Votre code doit être bien structuré et commenté.

Dans un deuxième temps, prenez une image au format PGM (p.ex. le fichier `evert.pgm`) de dimensions une puissance de deux. L’image peut être visualisée à l’aide de Matlab/Octave et du script `plotImg.m`.

Ecrivez ensuite un programme parallèle `filtragePar.cc` qui :

- lit un fichier au format PGM ;
- applique une FFT pour filtrer l’image en éliminant les basses fréquences ;
- sauvegarde l’image filtrée dans un fichier au format PGM.

Vous devez pouvoir montrer une exécution de ce programme et l’image filtrée résultante. L’application du filtrage avec 16 processeurs sur une image au format PGM (p.ex. `evert.pgm`) et un seuil de filtrage (en pourcent) produira un fichier filtré au même format (p.ex. `evert_filtre.pgm`).

Le lancement du programme se fera selon la syntaxe :

```
mpirun -np 16 filtragePar evert.pgm 10
```

Un seuil de 10 signifie que chaque coin du filtre a une taille égale au 10% de celle d’un côté de l’image (p.ex. pour une image de taille 256×256 un coin du filtre aurait une taille de 25×25).

2 Analyse de performance théorique

Etablissez les formules pour la complexité, le speedup, l’efficacité et la fonction d’isoeffacité de la FFT parallèle.

3 Rendu et évaluation

- Placez votre code de FFT parallèle 1D et de filtrage d’image (obligatoirement nommés `fftPar1D.cc` et `filtragePar.cc`) dans un dossier zippé en une archive à votre(vos) nom(s) (p.ex. `fahy_hohn.zip` pour un groupe formé de Axel Fahy et Rudolf Höhn). Cette archive devra être déposée dans le dossier **TP_FFT** via l’onglet **Travaux** du cours **High Performance Computing** sur le site dokeos.eig.ch.
- Rendez les formules établies dans l’analyse de performance théorique.
- Vous serez aussi interrogés oralement sur votre travail.