Projet Parallélisme
Produit matriciel distribué

Chloé Bensoussan

Université Nice Sophia-Antipolis
Master 1 IFI

1 Structure

J'ai utilisé une repprésentation matricielle linéarisée pour simplifier l'envoie des lignes et colonnes.

```
struct Matrice{
    int ligne, colonne;
    long long int *matrice;
};
```

2 Fonctionnalités implémentées

— Calcul du produit matriciel avec N multiple de P
— Gestion des matrices très grandes
— Gestion du déséquilibre dans le calcul, i.e N non multiple de P

3 Circulation en anneau

La circulation des lignes entre les processeurs se fait ainsi :

A chaque tour, les processeurs calculent le produit matriciel des lignes et colonnes reçus, en stockant le résultat dans une matrice. Puis le processeur root envoie ses lignes au suivant, et attend d'en recevoir depuis son précédent. Tous les autres processeurs reçoivent d'abord les lignes puis envoient les leurs au suivant. On répète cette opération jusqu'à une circulation complète des lignes vers tous les processeurs actifs (ceux qui calculent).

Ainsi, à la fin de tous les calculs, les processeurs possèdent une ou plusieurs colonnes de la matrice finale. Ils envoient la matrice calculée au processeur root qui les assemble dans la matrice finale.

4 Fonctions parallélisées

Les fonctions importantes parallélisées dans le programme sont :
— `void echange_ligne_colonne()`
— `void copie_matrice()`
— `void scatterv()`

La fonction `scatterv()` calcule la répartition des lignes pour chaque processus par rapport à la taille de la matrice et au nombre de processus. Elle calcule aussi le nombre de valeur que contiendra chaque processeur. Grâce à cela, chaque processus connaît ne nombre d'élements qu'il recevra à chaque tour.