

PPAR

Projet MPI

29 octobre 2007

Diffusion sur une grille

Nous allons étudier le phénomène de diffusion de chaleur sur une « grille ». La température en chaque point de la grille est représentée par un nombre réel (en simple précision) positif ou nul.

Dans le cadre de ce projet, nous limitons à une version simplifiée du phénomène de diffusion. La calcul de la valeur $u_{i,j}^{t+1}$ du point de coordonnées (i, j) au temps $t + 1$ est donné par l'équation suivante :

$$u_{i,j}^{t+1} = u_{i,j}^t + C (u_{i-1,j}^t + u_{i,j-1}^t + u_{i+1,j}^t + u_{i,j+1}^t - 4u_{i,j}^t),$$

le coefficient C devant vérifier $C < 1/2$.

En fixant $C = 1/5$, chaque point conserve donc un cinquième de son ancienne valeur, auquel viennent s'ajouter les contributions reçues de ses quatre voisins immédiats (Nord, Sud, Est, Ouest, s'ils existent). On considérera que toute la chaleur émise sur les parois de la grille est renvoyée.

L'algorithme se termine lorsque $\Delta < \epsilon$, où l'erreur Δ est défini par :

$$\Delta = \max_{\forall(i,j)} |u_{i,j}^{t+1} - u_{i,j}^t|.$$

Une implémentation possible de cet algorithme consiste à utiliser deux grilles : la grille N^0 contenant les valeurs correspondant au temps t , on calcule pour le temps $t + 1$ la grille N^1 à l'aide de la grille N^0 . A chaque pas de temps, les grilles sont inversées.

La configuration visée est une grille de dimensions 68040×68040 ($68040 = 2^3 \times 3^5 \times 5 \times 7$) avec tous les points initialement à 25 degrés exceptées des lignes de points à 100 degrés : il y a 1 ligne de points à 100 degrés toutes les 100 lignes. Pour obtenir une valeur quasiment uniforme sur l'ensemble de la grille, il faut utiliser $\epsilon = 10^{-5}$. En pratique, on pourra se limiter à 40 itérations au maximum lors du calcul parallèle de la grille de dimensions 68040×68040 .

Comme la valeur de chaque point de la grille est stockée sous la forme d'un réel en simple précision, cette grille occupe plus de 17 Go en mémoire, et une telle simulation ne peut donc être effectuée en séquentiel. Afin d'effectuer cette simulation en parallèle, nous allons décomposer cette grille entre 12 processus, chaque processus s'exécutant sur une machine distincte.

Travail à effectuer

Un exemple possible de code séquentiel vous est fourni : ce programme n'effectue bien sûr pas la simulation pour la configuration visée (68040×68040). Comme dans ce code séquentiel, votre programme parallèle devra afficher le temps « horloge » écoulé (*wall-clock time*), le nombre d'itérations nécessaires et la valeur finale de Δ .

1. Que devient la condition de terminaison en parallèle ?
2. Deux décompositions sont a priori possibles pour la distribuer la grille sur les 12 processus :

- une décomposition par bandes,
 - une décomposition par blocs (rectangulaires).
- Donner les avantages et les inconvénients de chaque décomposition. Ecrire un programme MPI qui effectue cette simulation en parallèle avec les deux décompositions, et déterminer quelle est en pratique la meilleure décomposition pour les ressources matérielles dont nous disposons.
3. Comment montrer que le programme parallèle donne le même résultat que le programme séquentiel ?
 4. Améliorer la solution retenue en introduisant, par exemple :
 - un recouvrement des communications par le calcul,
 - une programmation hybride MPI+OpenMP permettant de mieux exploiter les processeurs dual-core dont vous disposez,
 - des communications MPI persistantes (voir documentation MPI),
 - des topologies MPI cartésiennes (voir documentation MPI),
 - ...

Travail à remettre

Vous devrez remettre les documents suivants :

- le code source, sous la forme d'une archive `tar` compressée. L'archive ne doit contenir aucun exécutable, et elle devra contenir un fichier `Makefile` : la commande `make` devra lancer la compilation, et la commande `make exec` devra lancer l'exécution parallèle (pour la configuration visée, et avec le code final).
- un rapport présentant vos choix, vos implémentations (sans code source), vos résultats et vos conclusions. Une comparaison des différentes implémentations, ainsi qu'une réflexion sur le comportement de votre programme, seront particulièrement appréciées.

Quelques précisions importantes

- Vous devez lire le document intitulé « Projet PPAR : conditions d'utilisation de LAM ». Vous veillerez notamment à n'utiliser qu'une salle à la fois pour vos tests (12 machines sont nécessaires), et vous n'oublierez pas de tuer tous vos processus sur toutes les machines utilisées à la fin de vos tests.
- Les machines dont vous disposez ont 4 Go de mémoire vive, mais la compilation s'effectuera en 32 bits : la taille en mémoire de vos processus sera donc limitée à 3 Go.
- Attention aux quotas sur les disques : il n'est pas demandé de (et vous ne devez donc pas) sauvegarder la grille dans un fichier à la fin de la simulation.
- Le projet est à réaliser par binôme.
- Le projet devra être remis au plus tard le vendredi 30 novembre à minuit (heure locale) par courriel à : `Jean-Luc.Lamotte@lip6.fr` et `pierre.fortin@lip6.fr`