TP Introduction à la programmation parallèle n° 2

TP à rendre pour le 31 Mars au plus tard.

I) Produit matrice vecteur

On considère l'algorithme de multiplication d'une matrice avec un vecteur $y = y + A \cdot x$, où A est une matrice de taille $n \times n$, x et y sont des vecteurs de n éléments. On considère que la matrice A ainsi que les vecteurs x et y sont déjà distribués sur les processeurs, comme décrit par la suite. Le vecteur résultat y doit être distribué sur les processeurs en respectant la distribution initiale de y.

a) Distribution unidimensionnelle de la matrice A

Ecrivez un algorithme parallèle en pseudo-code et MPI qui considère que la matrice A est distribuée en utilisant une distribution uni-dimensionnelle au long des lignes. Ceci signifie que chaque processeur a dans sa mémoire un block de lignes de A. Les vecteurs x et y sont distribués aussi sur les processeurs au long de lignes.

Par exemple avec 4 processeurs, la matrice A est partitionnée en 4 blocks de lignes, A_0, A_1, A_2, A_3, où chaque block A_i est de dimensions $n/4 \times n$ (pour simplicité, on considère que n est un multiple de 4). Le block A_i sera stocké dans la mémoire du processeur P_i. De la même manière, les vecteurs x et y sont divisés en 4 sous-vecteurs x_i, y_i, $i = 1 \ldots 3$ de dimension $n/4 \times 1$. Les sous-vecteurs x_i, y_i sont stockés sur le processeur P_i.

b) Distribution bidimensionnelle de la matrice A

On considère une distribution bidimensionnelle au long des lignes et des colonnes de la matrice A sur les processeurs. Par exemple, pour 16 processeurs, la matrice sera distribuée sur les processeurs ainsi:

$$\begin{bmatrix} P_0 & P_1 & P_2 & P_3 \\ P_4 & P_5 & P_6 & P_7 \\ P_8 & P_9 & P_{10} & P_{11} \\ P_{12} & P_{13} & P_{14} & P_{15} \end{bmatrix}.$$

Avec cette distribution, les vecteurs x et y seront distribués sur $\sqrt{P}$ processeurs. Dans notre exemple, x sera distribué sur P_0, P_1, P_2, P_3,

$$\begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix}.$$

Le vecteur y sera distribué sur P_0, P_4, P_8, P_{12},

$$\begin{bmatrix} P_0 \\ P_4 \\ P_8 \\ P_{12} \end{bmatrix}.$$

Ecrivez l'algorithme pour calculer $y = y + Ax$ en pseudo-code et MPI en utilisant cette distribution bidimensionnelle.

Question 3 Calculez le temps parallèle des deux algorithmes, et expliquez les différences entre ces deux algorithmes. Pour estimer le temps de l'algorithme en parallèle, considérez que le temps d'envoie d'un message de n words est égal à :

$$\text{Message time} = \alpha + \#\text{words} \cdot \beta, \tag{1}$$

où α est la latence du réseau et β et l'inverse de la bande passante du réseau. Un broadcast ou un reduce entre P processeurs qui implique un message de n words est estimé comme $\log P(\alpha + n \cdot \beta)$

c) Remise du TP

Pour rendre votre TP, envoyez le pdf à Laura.Grigori@inria.fr, pour au plus tard le 31 Mars. Le document rendu peut avoir en annexe les codes que vous avez écrit pendant la séance du 17 Mars 2017.