IUP MAI3 - TD M54

FFT : transformée de Fourier Rapide sur Hypercube

Soit $n \in \mathbb{N}$, nous notons $N = 2^n$ la dimension du vecteur f dont nous souhaitons faire la FFT. Soit `rang` le numéro d'un processeur (en MPI on dirait d'un processus et on l'obtiendrait par `MPI_Comm_rank(MPI_COMM_WORLD,&rank);`). On définit une topologie d'hypercube de dimension n sur un réseau de taille 2^n par : soit $j_{n-1}, j_{n-2}, \ldots, j_1, j_0$ la numérotation en base deux de `rang` , c'est à dire $\forall l = 0, \ldots, n-1, j_l \in \{0, 1\}$ et

$$\texttt{rang} = \sum_{l=0}^{n-1} j_l 2^j,$$

les processus $\texttt{rangj} = \sum_{l=0}^{n-1} j_l 2^j$ et $\texttt{rangk} = \sum_{l=0}^{n-1} k_l 2^j$ sont voisins ssi $\exists! l_0 \in \{0, \ldots, n-1\}, j_{l_0} \neq k_{l_0}$ (et $\forall l = 0, \ldots, n-1, l \neq l_0 \Rightarrow j_l = k_l$)

1. Dessinez les hypercubes de dimension n pour $n = 1, 2, 3, 4$.

2. Combien chaque processus (ou processeur) a de voisins dans un hypercube ?

3. En réutilisant l'idée de l'algorithme papillon, proposez un algorithme parallèle qui en n étapes réalise la permutation miroir d'un vecteur de taille N sur un hypercube en supposant qu'à l'initialisation le processeur `rang` possède la composante $f_{\texttt{rang}}$ et qu'en sortie le processeur `rang` possède la composante $f_{P_n(\texttt{rang})}$ où P_n est la permutation miroir de taille 2^n.

4. Supposons que nous disposons de $P = 2^p$ processeurs reliés par un réseau de topologie hypercubique. Supposons qu'à l'initialisation, le processeur 0 detient le vecteur f de taille 2^n. Nous proposerons des algorithmes tels que le processeur 0 détienne le résultat F en fin de calcul. Nous supposons enfin que nos processus ont accès à un algorithme séquentiel performant `FFT(G,g,m)` qui calule la FFT de `g` de taille 2^m et retourne le résultat dans `G`.

   (a) Si $p = 1$ (cad le nombre de processeurs $P = 2$), en vous inspirant de l'algorithme papillon, proposez un algorithme SPMD ("à la mode MPI") qui calcule une FFT de taille 2^n en parallèle avec potentiellement une accélération d'un facteur 2 pour un n grand.

   (b) Même question avec $p = 2$.

   (c) Même question avec $p \in \mathbb{N}$ (tel que $p < n$).

IUP MAI3 - TD M54

FFT : transformée de Fourier Rapide sur Hypercube

Soit $n \in \mathbb{N}$, nous notons $N = 2^n$ la dimension du vecteur f dont nous souhaitons faire la FFT. Soit `rang` le numéro d'un processeur (en MPI on dirait d'un processus et on l'obtiendrait par `MPI_Comm_rank(MPI_COMM_WORLD,&rank);`). On définit une topologie d'hypercube de dimension n sur un réseau de taille 2^n par : soit $j_{n-1}, j_{n-2}, \ldots, j_1, j_0$ la numérotation en base deux de `rang` , c'est à dire $\forall l = 0, \ldots, n-1, j_l \in \{0, 1\}$ et

$$\texttt{rang} = \sum_{l=0}^{n-1} j_l 2^j,$$

les processus $\texttt{rangj} = \sum_{l=0}^{n-1} j_l 2^j$ et $\texttt{rangk} = \sum_{l=0}^{n-1} k_l 2^j$ sont voisins ssi $\exists! l_0 \in \{0, \ldots, n-1\}, j_{l_0} \neq k_{l_0}$ (et $\forall l = 0, \ldots, n-1, l \neq l_0 \Rightarrow j_l = k_l$)

1. Dessinez les hypercubes de dimension n pour $n = 1, 2, 3, 4$.

2. Combien chaque processus (ou processeur) a de voisins dans un hypercube ?

3. En réutilisant l'idée de l'algorithme papillon, proposez un algorithme parallèle qui en n étapes réalise la permutation miroir d'un vecteur de taille N sur un hypercube en supposant qu'à l'initialisation le processeur `rang` possède la composante $f_{\texttt{rang}}$ et qu'en sortie le processeur `rang` possède la composante $f_{P_n(\texttt{rang})}$ où P_n est la permutation miroir de taille 2^n.

4. Supposons que nous disposons de $P = 2^p$ processeurs reliés par un réseau de topologie hypercubique. Supposons qu'à l'initialisation, le processeur 0 detient le vecteur f de taille 2^n. Nous proposerons des algorithmes tels que le processeur 0 détienne le résultat F en fin de calcul. Nous supposons enfin que nos processus ont accès à un algorithme séquentiel performant `FFT(G,g,m)` qui calule la FFT de `g` de taille 2^m et retourne le résultat dans `G`.

   (a) Si $p = 1$ (cad le nombre de processeurs $P = 2$), en vous inspirant de l'algorithme papillon, proposez un algorithme SPMD ("à la mode MPI") qui calcule une FFT de taille 2^n en parallèle avec potentiellement une accélération d'un facteur 2 pour un n grand.

   (b) Même question avec $p = 2$.

   (c) Même question avec $p \in \mathbb{N}$ (tel que $p < n$).