

L'objectif premier de ce mini-projet est de proposer une implémentation de la transformée de Fourier Rapide d'un vecteur f de $N = 2^n$ complexes (avec $n \in \mathbb{N}$), distribuée sur un réseau de processus en topologie hypercubique et de discuter des performances obtenues. L'objectif second est de discuter d'autres implémentations possibles en s'inspirant de [1] (et des autres documents fournis).

Nous rappelons la topologie de l'hypercube. Soit **rang** le numéro d'un processus (ou processeur). On définit une topologie d'hypercube de dimension n sur un réseau de taille 2^n par : soit $j_{n-1}, j_{n-2}, \dots, j_1, j_0$ la numérotation en base deux de **rang**, c'est à dire $\forall l = 0, \dots, n-1, j_l \in \{0, 1\}$ et

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

les processus $\mathbf{rang}j = \sum_{l=0}^{n-1} j_l 2^l$ et $\mathbf{rang}k = \sum_{l=0}^{n-1} k_l 2^l$ sont voisins ssi $\exists l_0 \in \{0, \dots, n-1\}, j_{l_0} \neq k_{l_0}$ (et $\forall l = 0, \dots, n-1, l \neq l_0 \Rightarrow j_l = k_l$)

1. A l'aide de `MPI_Graph_create`, construisez un hypercube de dimension p (comportant 2^p noeuds) avec MPI.
2. Nous supposons que $p < n$. Proposez une implémentation de la FFT sur un hypercube en supposant qu'à l'initialisation, le processeur 0 détient le vecteur f de taille 2^n . Vous proposez un algorithme tel que le processeur 0 détienne le résultat F en fin de calcul (pour le réaliser, vous vous aiderez du TD sur le FFT parallèle vue en cours et vous supposerez que vos processus ont accès à un algorithme séquentiel performant (algorithme de PEASE vue en M53) `FFT(G, g, m)` qui calcule la FFT séquentiel de g de taille 2^m et retourne le résultat dans G).
3. Testez votre algorithme de FFT distribué en utilisant MPI.
4. Discutez d'autres implémentations possibles en vous aidant de la documentation fournie, ou de tout autres documents.

Références

- [1] D. Gannon and W. Jalby The influence of memory hierarchy on algorithm organisation programming FFTs on a vector multiprocessor. Characteristics of parallel algorithms, MIT Press, 1987.XS