TP MPI N°2 : Communication collective et mesure du temps d'exécution

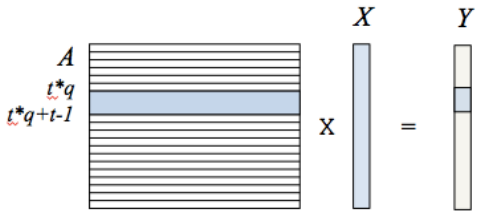
En MPI, la communication collective est celle qui fait intervenir tous les processus d'un communicateur. Elle est omniprésente dans des programmes parallèles. Nous allons manipuler quelques fonctions de MPI pour la communication collective à travers la multiplication d'une matrice carrée et d'un vecteur.

1. Multiplication Matrice-Vecteur : *Y=AX*

$$Y_i = \sum_{j=0}^{n-1} a_{ij} * X_j, i = 0,...,n-1; a_{ij}, X_j, Y_i \in \Re$$

2. Parallélisation de *Y=AX* --- Version ligne

Nous allons paralléliser le calcul des composants de *Y*. Soit *N* le nombre de processus d'une exécution, chaque processus effectue le calcul de *n/N* composants de *Y*. En premier temps, nous allons utiliser les valeurs de *n* et *N* qui sont divisibles. Nous supposons que seul le processus du rang *0* qui dispose de la matrice *A* et du vecteur *X* au début du programme.



L'algorithme parallèle est le suivant :

- Le processus du rang *0* :
  i. Initialisation de la matrice *A* et du vecteur *X*
  ii. Diffusion du vecteur *X* (*MPI_Bcast*)
  iii. Distribution des lignes de la matrice *A* (*MPI_Scatter*)
  iv. Calcul de ses composants de *Y*
  v. Fusion des composants de *Y* (*MPI_Gather*)
  vi. Affichage du *Y*

- Les autres processus :
  i. Diffusion du vecteur *X* (*MPI_Bcast*)
  ii. Distribution des lignes de la matrice *A* (*MPI_Scatter*)
  iii. Calcul de ses composants de *Y*
  iv. Fusion des composants de *Y* (*MPI_Gather*)

Les opérations de communication seront réalisées à l'aide des fonctions de communication collective. **Attention** : les données doivent être contiguës en mémoire (ex. les éléments de *A* et de *X*)

Vérifier le résultat du programme parallèle en comparant avec celui calculé en séquentiel dans le processus du rang *0*.

3. Mesure du temps d'exécution :

Effectuer le mesure de performance de votre programme parallèle selon la même procédure vue en TP d'OpenMP, avec *n* suffisamment grand. Les fonctions de mesure du temps sont similaires :

- *double MPI_Wtime(void) ;* donne le temps en micro/nano secondes à partir d'un point donné. *double MPI_Wtick(void) ;* donne la résolution de l'horloge.
- Une synchronisation de tous les processus (à l'aide de *MPI_Barrier*) est nécessaire avant le 1er mesure du temps, afin que les processus démarrent leur mesure en même temps et que les temps d'exécution de tous les processus soient comparables.