Travaux Pratique 10

Exercice 1 :

Pour la résolution du système triangulaire, j'ai choisi d'envoyer les données par bloc de données calculées, plutôt que chaque donnée une à une.

Au niveau de l'algorithme, je crée d'abord une recopie de B. Ensuite :

- Le premier processeur fait son calcul des premiers X avec la formule vue en td, puis il les envoie.
- Le dernier processeur, pour chaque processeur précédent i, reçoit un bloc de données de taille m. Ce sont les X calculés par ce processeur. Et les utilise pour mettre à jour sa copie de B. Ensuite il fait son calcul des X.
- Les autres processeurs reçoivent pour chaque processeur précédent i un bloc de données de taille m, et les transmettent aussitôt. Ce sont les X déjà calculés, il faut s'en servir pour mettre à jour B mais il ne faut pas y toucher et les envoyer le plus vite possible aux autres processeurs pour qu'ils ne soient pas bloqués. Ensuite il faut calculer les X et les envoyer au processeur suivant.

Je me suis aperçu que j'aurais pu utiliser les méthodes Start() et End() de B pour me faciliter le calcul des index, mais j'ai préféré laisser comme ça. Au moins, ça me semble fonctionner correctement.

J'ai aussi regardé ce que donnait les temps de calculs lors de l'envoi par donnée, cela me semblait plus lent que l'envoi par bloc. Peut-être qu'on gagne du temps car on limite le nombre d'appel aux primitives de communication, mais en faisant cela, les processeurs peuvent être plus souvent en attente.

Exercice 2 :

Pour le produit matrice vecteur, c'est une adaptation de l'algorithme vu en cours en C++.

Donc on initialise le vecteur X et on copie le vecteur B dans un buffer. Ensuite, pour chaque processeur, on envoie le buffer et on reçoit le buffer du précédent, tous deux de manière asynchrone. Ensuite, on utilise son buffer pour faire une partie des calculs en local. Quand les deux requêtes sont effectuées, on échange les deux buffers et on boucle. Cela permet d'utiliser le buffer du précédent processeur dans la boucle suivante, puis du précédent du précédent dans celle d'après, et ainsi de suite pour tous les processeurs.