Algorithmique sur anneau de processeurs
TD – Enoncé

Plan

1. Echange total
2. Diffusion pipelinée
3. Factorisation LU

1 Echange total sur l'anneau

Enoncé

– Chaque processeur k veut envoyer un message à tous les autres : p diffusions simultanées

– Hypothèses :

1. Tous les messages ont même longueur L
2. Au départ, chaque processeur dispose d'un message stocké à l'adresse my_adr
3. A la fin, chaque processeur a en mémoire le même tableau, contenant à la case k le message diffusé par P_k.

2 Diffusion pipelinée

Enoncé

– Diminuer le temps de la diffusion pour des messages de grande longueur
– Découper le message en r morceaux (de taille égale, en supposant L divisible par r)
– Le processeur émetteur envoie successivement les r morceaux, qui circulent simultanément sur l'anneau
– Au début du programme, les r morceaux du message sont stockés aux adresses adr[1],...,adr[r] du processeur k
– A la fin du programme, le message est stocké dans tous les processeurs aux mêmes adresses.

1. Ecrire le programme
2. Donner le temps d'exécution en fonction de r
3. Calculer la meilleure valeur de r.

3 Factorisation LU

Enoncé

– Algorithme séquentiel, matrice $n \times n$ indicée de 0 à $n-1$:

```
for k = 0 to n − 2 do
   /* Tâche T_kk : préparation de la colonne k
   prep(k) : for i = k + 1 to n − 1 do a_ik = −a_ik/a_kk
   for j = k + 1 to n − 1 do
     /* Tâche T_kj : mise à jour de la colonne j
     update(k,j) : for i = k + 1 to n − 1 do a_ij = a_ij + a_ik * a_kj
```

– Paralléliser cet algorithme sur un anneau de p processeurs :

1. Distribuer les colonnes de A aux processeurs : fonction d'allocation alloc
2. Supposer d'abord alloc donnée et écrire le code
3. Comment choisir une bonne fonction d'allocation ?
4. Proposer des améliorations.