

PPAR

4

TP 4 : convolution

Algo parallèle : avec envois bloquants,
version - par bandes

Hypothèse "nombre - lignes" est divisible par "nombre - processus"

1/ Initialisation :

- tous les processus récupèrent "filtre" et "nbiter" sur la ligne de commande
- le processus "root" :
 - lit le fichier "Karter"
 - récupère $\begin{bmatrix} "h" \\ "w" \end{bmatrix}$ (nombres de lignes / nombre de colonnes)
 - vérifie l'hypothèse (*)
- opération Broadcast depuis processus root pour diffuser $"h"$ et $"w"$ à tout le monde.
- $\text{local-}h = h / \text{NP} + (p > 0 ? 1 : 0) + (p < \text{NP} - 1 ? 1 : 0)$
- Allocation mémoire du tampon local dans chaque processus
↳ taille: local-h * w
↳ nom: local-data

2/ Répartition de l'image sur tous les processus :

- Avec NPI-Scatter (2 aux paramètres : processus ~~particulier~~)
- ou "à la main" → distinguer processus p et processus $\text{NP} - 1$ (en envoi et en réception)

3/ Convolution en// :

- Pour i allant de 0 à nbiter-1

Si processus précédent existe :

- envoyer la ligne 1 à p'
- recevoir la ligne 0 de p'

Fin si

Si processus suivant p'' existe :

- envoyer la ligne local-h-2 à p''
- recevoir la ligne local-h-1 de p''

Fin si

convolution (filtre, local-data, local-h, w)

Fin pour

éviter tout inter-lance avec des gros messages il faut inverser envoi & réception dans "si processus suivant existe"

⚠ de car
Send non
synchrone
car petit
message...
(et car on ne
pas un envoi
périodique)

CA