

TP 4: convolution

Algo parallèle : avec ~~des~~ communications non bloquantes (recouvrement commun/calc) et toujours version par bande
 → Seule l'étape 3 change par rapport à la version avec les comm bloquantes.

3/ Convolution en //:

⚠ on ne peut pas utiliser convolution: voir convd-par-recalc

Pour i allant de 0 à nbiter-1

allocation de tmp !
 ↳ gardée à l'intérieur de la bande sur i pour comparaison locale avec comm bloquantes

Si processus précédent p' existe:
 - initier envoi non bloquant de la ligne 1 à p'
 - initier réception non bloquante de la ligne 0 de p'
 Fin si
 Si processus suivant p'' existe:
 - initier envoi non bloquant de la ligne local- $h-2$ à p''
 - initier réception non bloquante de la ligne local- $h-1$ de p''
 Fin si

Faire calcul local:

convolution-bis(filtre, local_data + w , local- $h-1$, w);
 Fin calcul local

↳ si p existe (+ sinon)

si p existe (sinon)

Attendre fin des communications non bloquantes (envoi et réception)

Faire calculs sur bords:

Si processus précédent p' existe:
 convolution-bis(filtre, local_data, 3, w);
 Fin si

Si processus suivant p'' existe:
 convolution-bis(filtre, local_data + w , (local- $h-3$), 3, w);
 Fin si

Fin calculs sur bords

Fin pour

peut être fusionné

Copie de tmp dans local_data

libération mémoire de tmp !

$tmp + w$ (local- $h-3$)