

HPC - Projet :

Le modèle Shallow water

MAIN 4 - Cours de S.Graillat

TD: Constantinos Malassikis et Salli Moustafa (ANEO)

Romaric Kanyamibwa, Colette Voisembert

Polytech Sorbonne

Table des matières

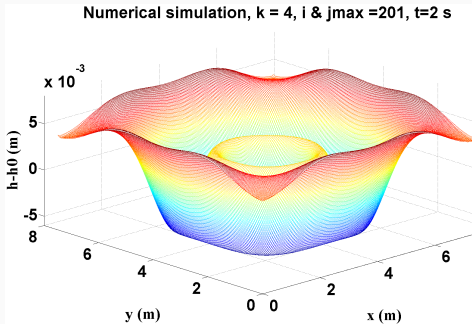
1. Le modèle shallow water
2. Parallélisation MPI
3. Parallélisation:Décomposition par Bande
4. Parallélisation:Décomposition par Bloc
5. Parallélisation:Entrée Sortie Parallèle
6. Conclusion

Le modèle shallow water

Motivations

Les Équations S-W permet la représentation de l'écoulement d'un fluide homogène sur la verticale

- Modélisation avec des EDP hyperbolique
- Résolution avec des schéma Numérique



$$\frac{\partial u}{\partial t} = -g^* \cdot \frac{\partial h}{\partial x} + f \cdot v - \gamma \cdot u$$

$$\frac{\partial v}{\partial t} = -g^* \cdot \frac{\partial h}{\partial y} - f \cdot u - \gamma \cdot v$$

$$\frac{\partial h}{\partial t} = -H \cdot \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right)$$

Les paramètres sont :

- q^* : gravité réduite , f : force de Coriolis
- γ : facteur de dissipation, H : hauteur moyenne de la colonne

Schema Numerique

Sur la grille Arakawa-C nous obtenons le schéma numérique suivant

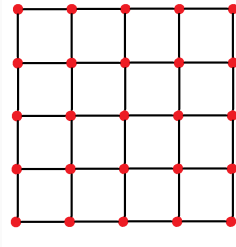
$$\begin{aligned}\frac{u_{i,j}^t - u_{i,j}^{t-1}}{\Delta t} &= -g^* \cdot \frac{h_{i+1,j}^{t-1} - h_{i,j}^{t-1}}{\Delta x} + f \cdot \frac{v_{i,j}^{t-1} + v_{i+1,j}^{t-1} + v_{i,j+1}^{t-1} + v_{i+1,j+1}^{t-1}}{4} - \gamma \cdot u_{i,j}^{t-1} \\ \frac{v_{i,j}^t - v_{i,j}^{t-1}}{\Delta t} &= -g^* \cdot \frac{h_{i,j}^{t-1} - h_{i,j-1}^{t-1}}{\Delta y} + f \cdot \frac{u_{i-1,j-1}^{t-1} + u_{i,j-1}^{t-1} + u_{i-1,j}^{t-1} + u_{i,j}^{t-1}}{4} - \gamma \cdot v_{i,j}^{t-1} \\ \frac{h_{i,j}^t - h_{i,j}^{t-1}}{\Delta t} &= -H \cdot \left(\frac{u_{i,j}^{t-1} - u_{i-1,j}^{t-1}}{\Delta x} + \frac{v_{i,j+1}^{t-1} - v_{i,j}^{t-1}}{\Delta y} \right)\end{aligned}$$

Parallélisation MPI

Probleme Séquentiel

Modelisation Séquentiel (Algorithme)

- Tableaux hFil, uFil, vFil, hPhy, uPhy, vPhy de taille 2*size_x*size_y
- Initialisation de hFil(i,j) à t=0 avec
$$\exp\left(\frac{\left(\frac{i*dx-gmx}{gsx}\right)^2}{2}\right) * \exp\left(\frac{\left(\frac{j*dy-gmy}{gsy}\right)^2}{2}\right)$$
- Initialisation des autres tableaux à 0
- Implémentation du Schéma Numérique

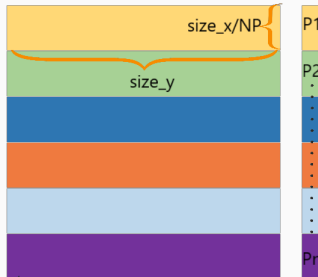


Parallélisation:Décomposition par Bande

Decomposition par Bande

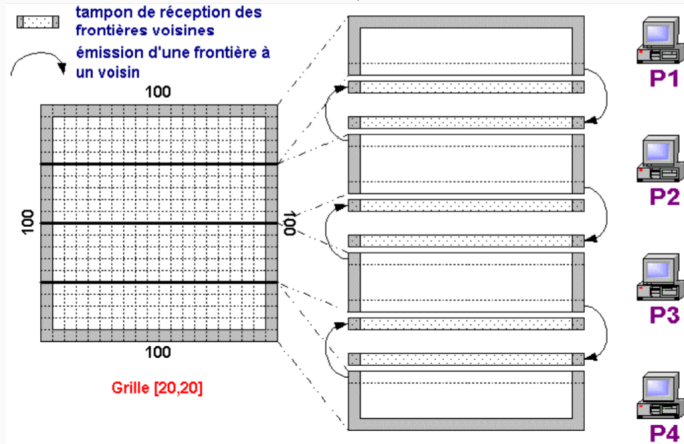
Algorithme Parallèle

- Tableaux hFil_local, uFil_local, vFil_local, hPhy_local, uPhy_local, vPhy_local de taille $2 \times \text{size_x} / \text{NP} \times \text{size_y}$
- Initialisation de hFil(i,j) à t=0 avec
$$\exp\left(\frac{\left(\frac{(i + \text{size_x} / \text{NP} * \text{myrank}) * dx - gmx}{gsx}\right)^2}{2}\right) * \exp\left(\frac{\left(\frac{j * dy - gmy}{gsy}\right)^2}{2}\right)$$
- Implémentation du Schéma Numérique



Decomposition par Bande

Send/Receive nécessaires pour le calcul du schéma



Decomposition par Bande

Block-Mode			
size_x:8192 , size_y:8192 , nbsteps:20			
Nombre de Processus	t [en sec]	Speed-up	Efficacité
1	709.295	-	-
2	312.522	2.26958	113.4792%
4	230.204	3.08116	77.0290%
8	182.878	3.87851	48.4815%
16	164.334	4.31618	26.9761%

Decomposition par Bande: Recouvrement de Communication

Non-block Mode			
size_x:8192 , size_y:8192 , nbsteps:20			
Nombre de Processus	t [en sec]	Speed-up	Efficacité
1	709.295	-	-
2	298.695	2.374 65	118.7323%
4	190.587	3.721 63	93.0408%
8	183.269	3.870 22	48.3780%
16	163.319	4.343	27.1438%

Decomposition par Bande

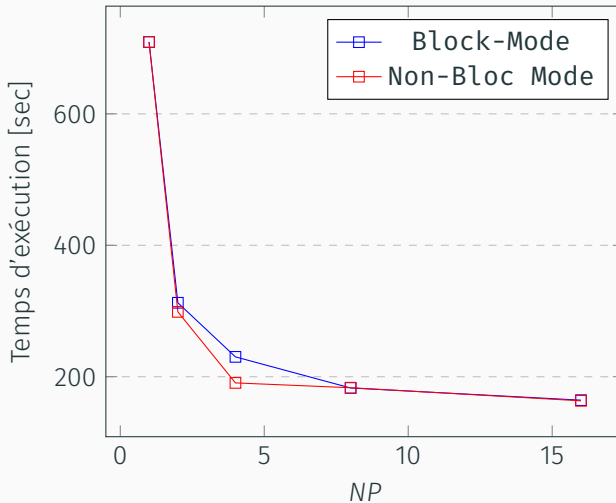
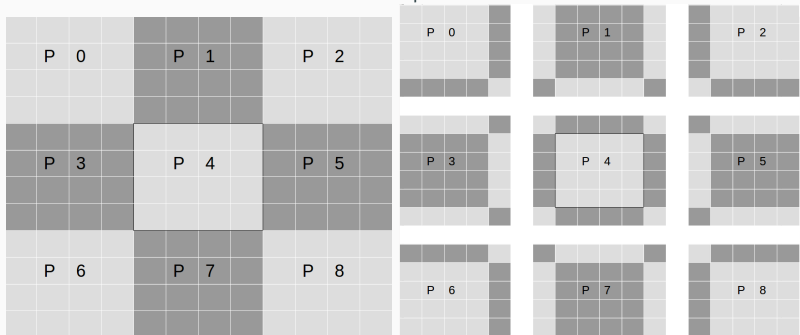


Figure 1: Temps d'exécution en fonction du nombre de processeurs

Parallélisation:Décomposition par Bloc

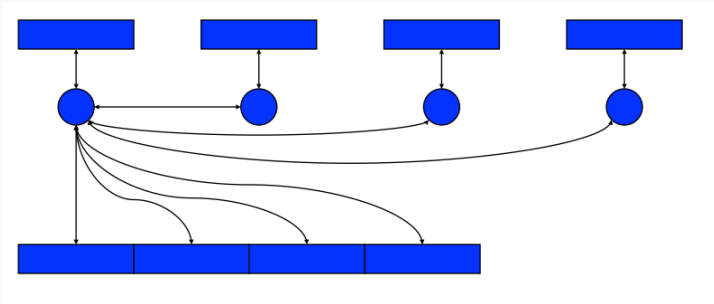
Decomposition par Bloc

Send/Receive nécessaires pour le calcul du schéma

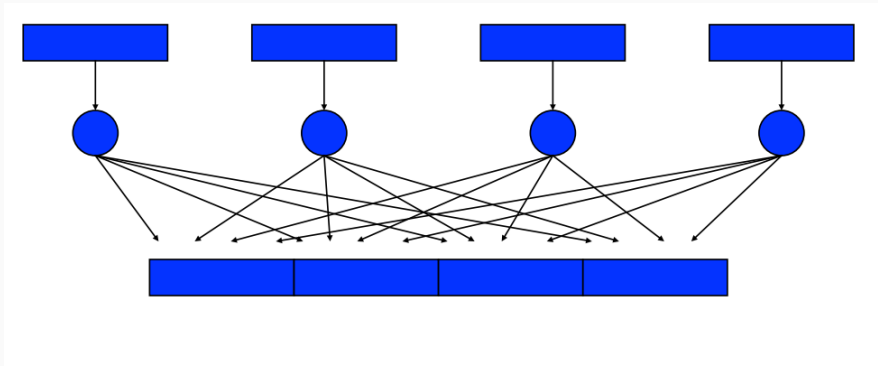


Parallélisation:Entrée Sortie
Parallèle

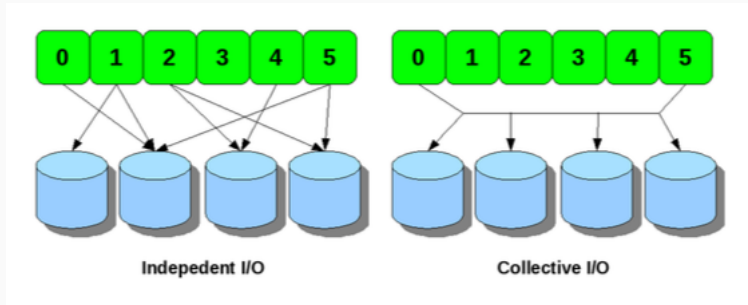
Entrée Sortie Parallèle:E/S Séquentiel



Entrée Sortie Parallèle: E/S Parallèle



Entrée Sortie Parallèle: E/S Parallèle



Entrée Sortie Parallèle: E/S Parallèle

MPI_IO			
size_x:512 , size_y:512 , nbsteps:80 ,NP:4			
Mode d'exécution	t [en sec]	Speed-up	Efficacité
NP:1 - Block-Mode	9.62589	-	-
Block-Mode	5.09481	1.88936	47.23%
NB Mode	4.7561	2.0239	50.60%
BMode MP_IO	4.28357	2.24715	56.18%
BMode NB-MP_IO	4.22961	2.27582	56.90%
NBMode MP_IO	3.78488	2.54323	63.58%
NBMode NB-MP_IO	3.64451	2.64119	66.03%

Conclusion

- De nombreux moyens d'optimiser toujours plus finement le calcul de l'image
- Fort potentiel d'accélération avec SIMD et OpenMp (par bande, 4 fois plus vite)
- Par bloc mieux que par bande car la taille des échanges diminue quand le nombre de processus augmente
- Mettre des Send et Receive non bloquants permettent encore d'accélérer mais relativement peu par rapport aux autres améliorations