Parallélisation de la résolution de l'équation de la chaleur – OpenMP / MPI

Thomas Poisson – 7 Décembre 2021

Problème considéré

- Température dans une plaque : équation de la chaleur $$\frac{\partial}{\partial t}u(x,y,t) - \alpha\nabla u(x,y,t) = 0$$

- u : température en (x,y) au temps t de la plaque
- α : coefficient de diffusion thermique
- Condition initale : u(x,y,0) = u0(x,y)
- Conditions aux limites de Dirichlet

Schéma différences finies progressive

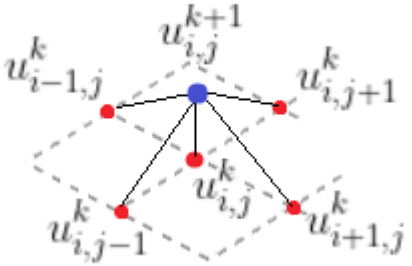
- **Discrétisation du domaine :**

$$x_i = i\delta x, i = 0, ...n_x \quad y_j = j\delta y, j = 0, ..., n_y \quad t_k = n\delta t, k = 0, ..., n_t$$

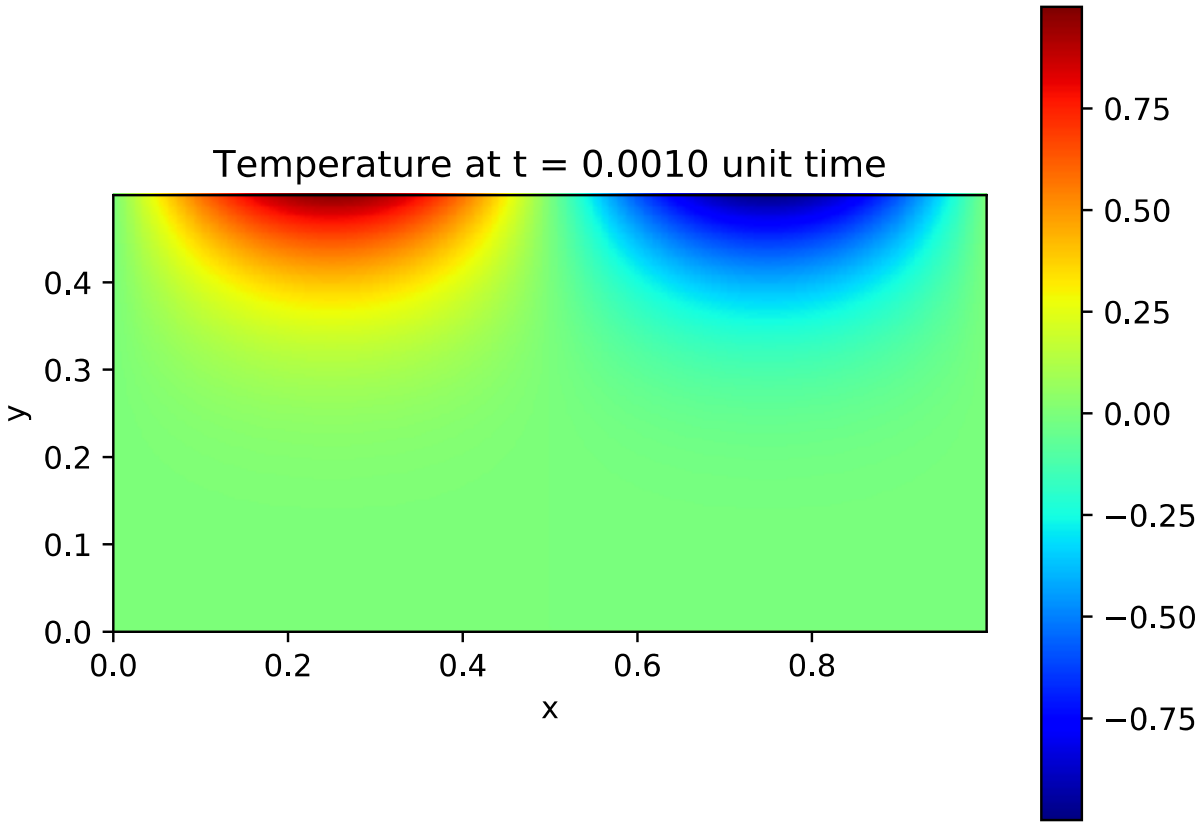
$$u_{i,j}^{(k+1)} = u_{i,j}^{(k)} + \alpha\frac{\delta t}{(\delta x)^2}[u_{i-1,j}^{(k)} - 2u_{i,j}^{(k)} + u_{i+1,j}^{(k)}] + \alpha\frac{\delta t}{(\delta x)^2}[u_{i,j_1}^{(k)} - 2u_{i,j}^{(k)} + u_{i,j+1}^{(k)}]$$
$$\simeq u(x_i, y_j, t_{(k+1)})$$

- **Condition CFL :**

$$\alpha\delta t(\frac{1}{(\delta x)^2} + (\frac{1}{(\delta y)^2}) < \frac{1}{2}$$

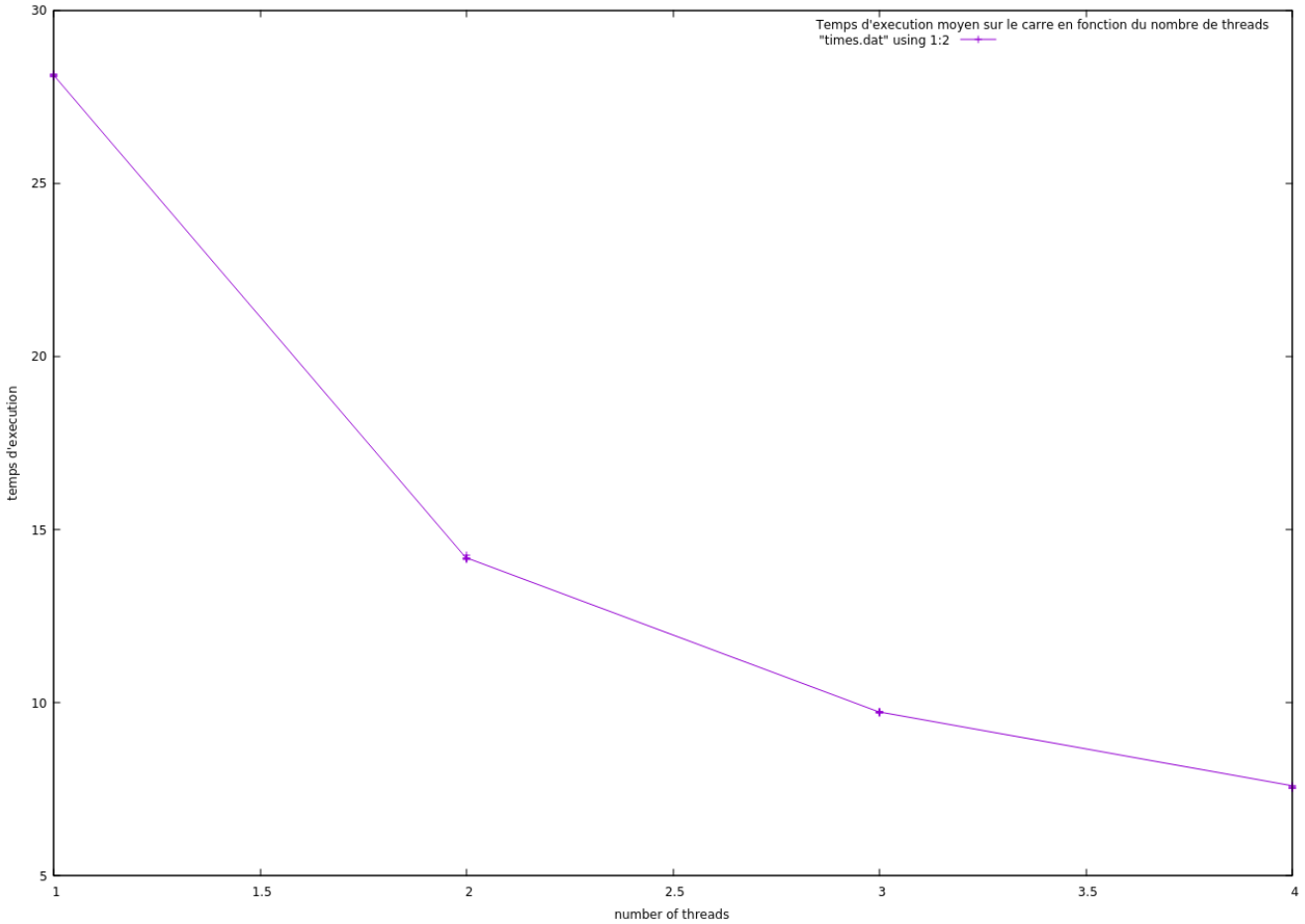


Parallélisation - OpenMP



Parallélisation - OpenMP

- Carré : Nx = 500 ; Ny = 500 ; Nt = 10000



Parallélisation - OpenMP

- Conclusion sur l'utilité :

  Programmes de tests choisis uniquement pour la mesure.

- Coût de l'overhead sur les cas pratiques

Parallélisation – MPI – Work in progress

- **Division de la matrice en blocs / sous-matrice**

  Chaque process résout localement son itération à chaque pas de temps

  Il transmet et reçoit ensuite, si nécessaire, les frontières du bloc résolu au process voisin

  → cette opération est bloquante au niveau de la réception !

- **Communicateur topologique – obtention des voisins**

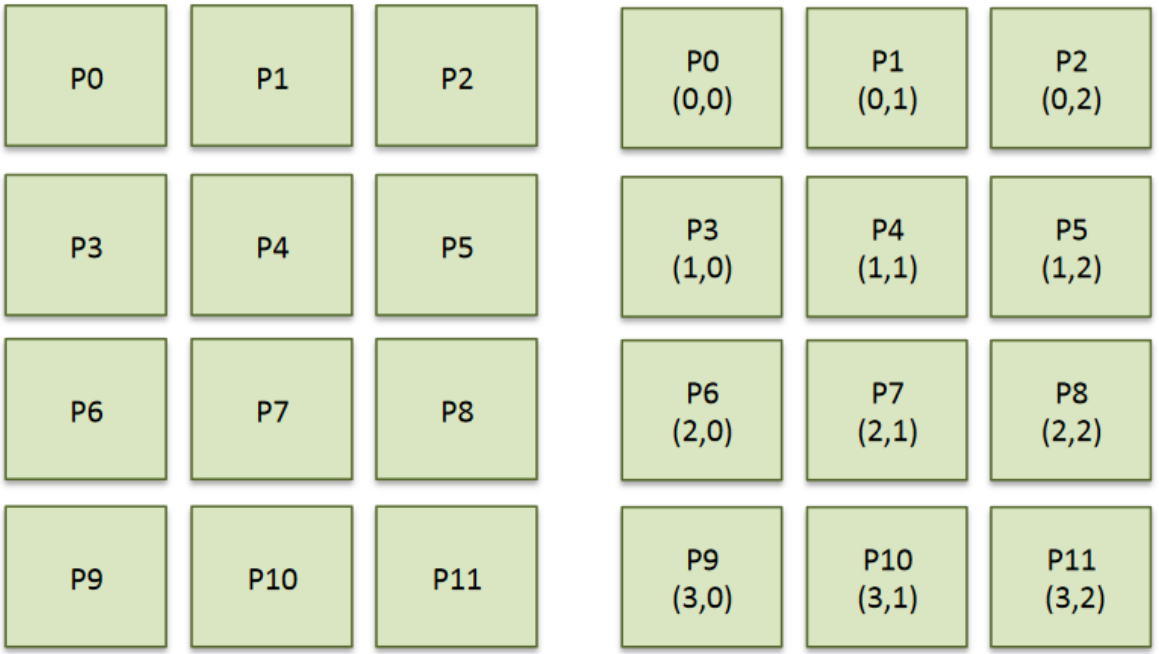
  MPI dispose de communicateurs adaptés à ce type de problème :

  MPI_Cart_shift()

Parallélisation - MPI

Pour 12 = 3x4 process :

- Process 0 → coord(0, 0)
- Process 1 → coord(0, 1)
- Process 2 → coord(0, 2)
- Process 3 → coord(1, 0)
- etc...



→ Permet de s'adapter dans le cas d'une plaque rectangulaire

→ Obtention des voisins :

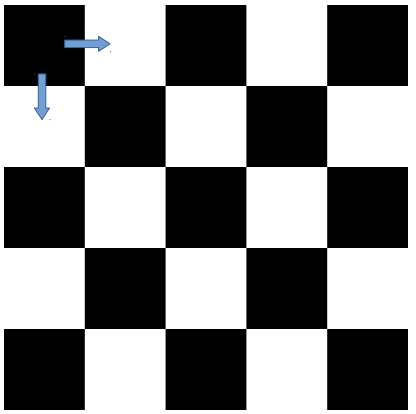
MPI_Cart_shift()

Parallélisation - MPI

- **Transmission des bords**

  décision de si le process reçoit

  ou envoie ses bords en premier



- **Implémentation : traitement particulier lors de la résolution ?**

  MPI_PROC_NULL() → Permet de ne pas avoir à tester

  lors de l'exécution si l'on se situe sur un bord ou pas

Merci de votre attention