

Un exemple de A à Z (IV)

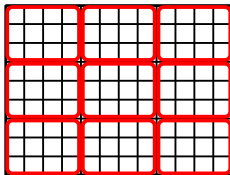
Programmation multi-processus avec MPI

Marc Tajchman

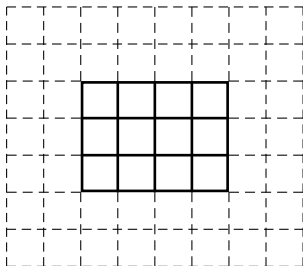
CEA Saclay, DEN/DM2S/STMF/LGLS

8 octobre 2012

La version multi-processus du code utilise un découpage en sous-domaines 2D, analogue à second découpage vu pour la version OpenMP.



- Chaque sous-domaine est affecté à un processus MPI, qui le calcule comme un problème indépendant.
- La cohérence de la solution sur les différents sous-domaines est assurée par un ensemble de “cellules miroir”.



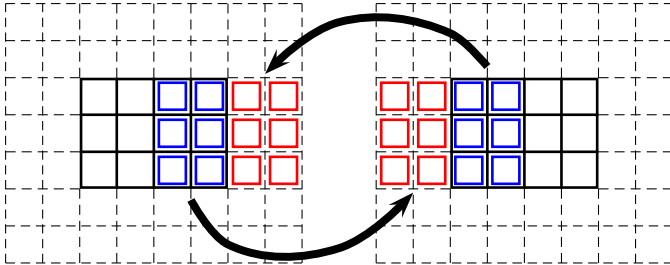
Cellule miroir



Cellule de calcul

Sur un sous-domaine, les cellules miroir contiennent une copie de cellules situées sur les domaines voisins.

- La copie vers les cellules miroir se fait à l'étape de calcul des conditions limites de l'algorithme (utilisation de l'instruction `MPI_Sendrecv`)



(cellules miroir en rouge, cellules de calcul en bleu)

Les parties du code à modifier sont:

- Au début, calcul des relations de voisinage entre sous-domaines
- Dans la partie du calcul de pas de temps, après le calcul du pas de temps local dans chaque sous-domaine, détermination du pas de temps global (par un des opérateurs de réduction de MPI)
- Dans la partie du calcul des conditions aux limites sur chaque sous-domaine, échange de valeurs entre sous-domaines voisins.

Quelques règles:

- Autant que possible, envoyer des messages plus longs plutôt que plusieurs messages plus petits

- Minimiser les interfaces entre sous-domaines

Par exemple, préférer des sous-domaines “carr’es” à des sous domaines “allongés”.

- Remplacer certains transferts par des calculs redondants

Exemple dans le code:

les variables physiques sont calculées à la fois sur les cellules miroir et sur les cellules d'origine

- Recouvrir les communications avec des calculs qui n'en dépendent pas (i.e. faire des communications asynchrones)

Exemple dans le code:

Les parties du calcul “calcul des variables physiques” et “application de la loi d’état” sont purement locales \Rightarrow on peut les effectuer sur les cellules de calcul et compléter à la fin des communications, les calculs sur les cellules miroirs .

Tester les temps relatifs de communication et de calcul pour voir si cela en vaut la peine !

Parfois les changements dans le code sont importants.

De plus, on diminue l'efficacité des calculs locaux (voir plus loin prog. hybride MPI+OpenMP).