Patterns de parallélisme implicite efficaces appliqués aux SIG

Kevin Bourgeois

LIFO - Géohyd

Introduction

Introduction

- La quantité des données à traiter croît pouvant monter jusqu'à plusieurs péta octets de données à traiter.
- Paralléliser pour accélérer les calculs mais les spécialistes sont rares.

Mais des solutions existent :

- Des bibliothèques *clé en main* avec des fonctions spécifiques à un domaine.
- Des squelettes de programmation parallèle qui fournissent des modèles abstraits paramétrables que l'on peut dériver vers des programmes concrets.

Géosciences

Les géosciences ne font pas exception

- Certains problèmes peuvent être très complexes comme, les études sur l'érosion des sols, le cycle de l'eau
- Les géomaticiens programment généralement en python mais ne sont pas experts en parallélisme.

Plan

Le reste de la présentation est organisé ainsi :

1. Patterns de parallélisme implicite
2. Résultats
3. Conclusion

Patterns de parallélisme implicite

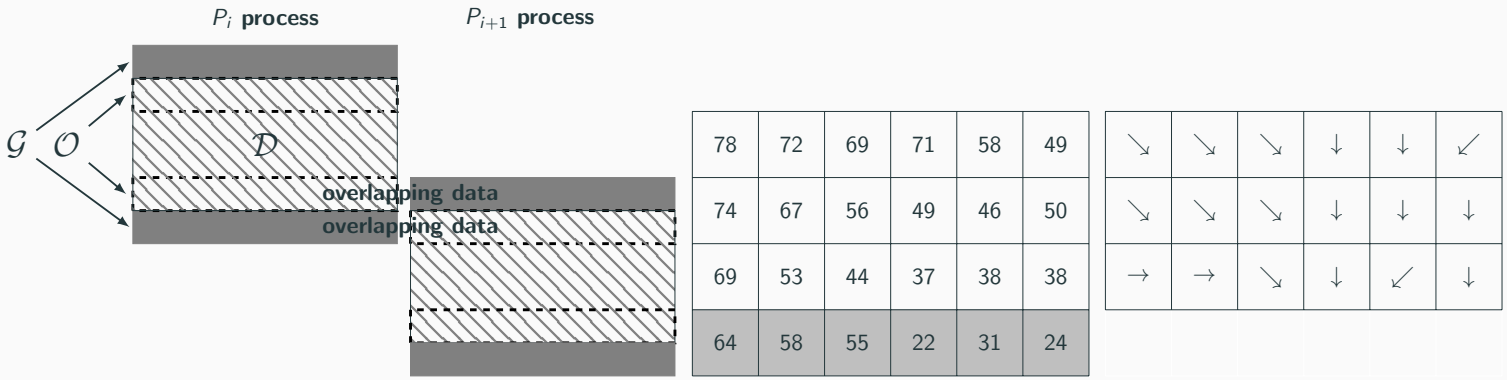
Qu'est-ce que c'est ?

- Extraction des communications, des synchronisations et la distribution des données d'un algorithme parallèle.
- On obtient un modèle abstrait qui peut être spécialisé avec des fonctions fournies par le scientifique.
- Ensuite le pattern spécialisé peut être transformé en un programme parallèle.

Stencil

- Pattern récurrent pour les simulations, les automates cellulaires...
- Algorithme itératif
- La valeur d'une cellule à l'itération courante dépend de la valeur des cellules voisines à l'itération précédente.

La distribution des données



(a) Domain definitions for two processes with line bands **(b)** Local data with a ghost line in gray **(c)** Computed Directions

Figure 1 – Ghost cells on the flow direction example

```
1  Function STENCIL(m_in :mesh, N_size :int, f_update :func, f_border :func,
     nb_iter :int) :mesh is
2      mesh m_out;
3      for i=1 to nb_iter do
4          Exchange_Ghosts(m_in);
5          foreach Cell c at the borders of D\G do
6              m_out(c) = f_border(c, m_in);
7          foreach Cell c ∈ D\G not at the border do
8              m_out(c) = f_update(c, m_in);
9          m_in = m_out;
10     return m_out;
```

Algorithm 1: Stencil Pattern

Pre-Defined Dependency Pattern

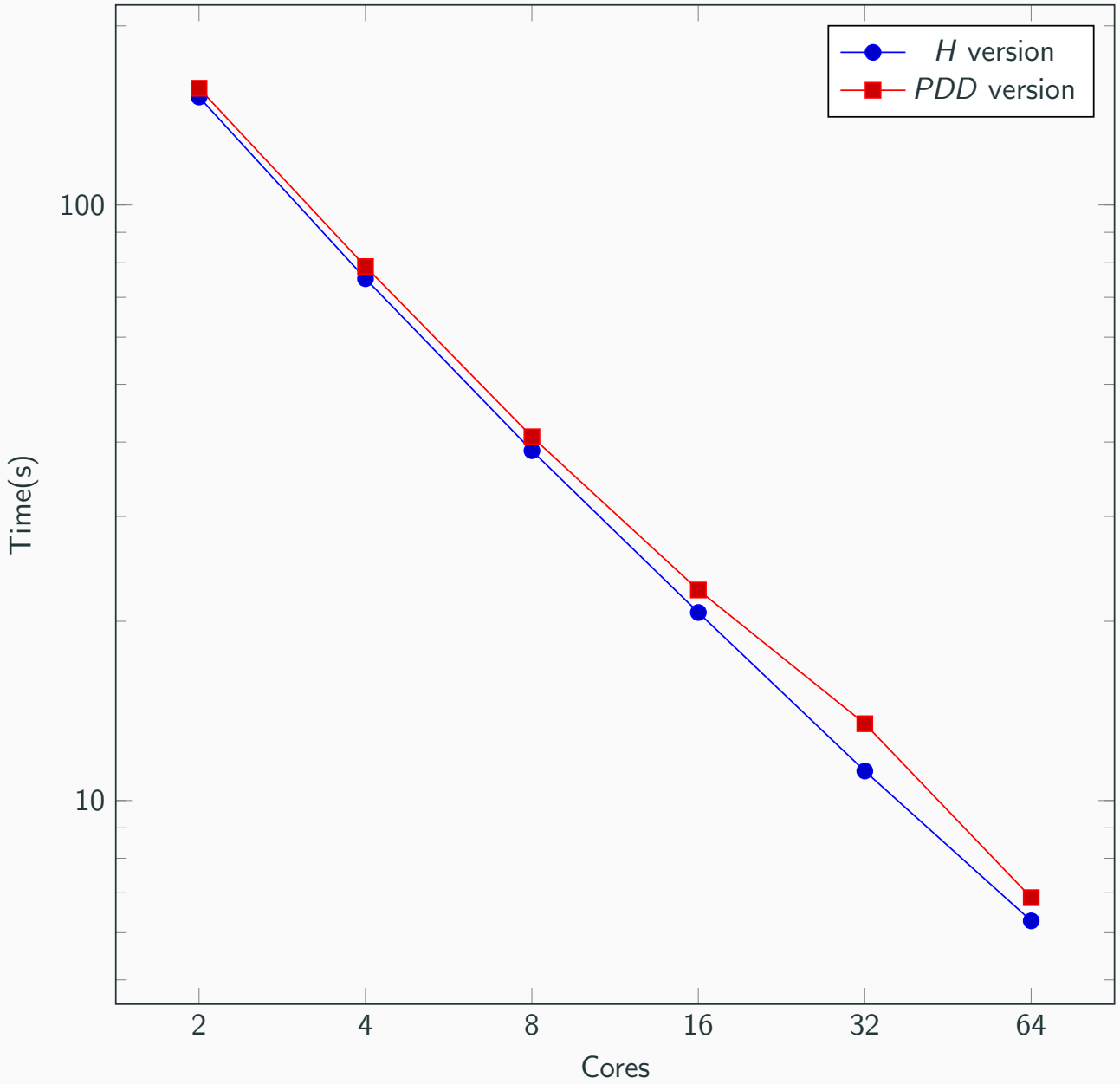
- Paralléliser des problèmes moins prévisibles en utilisant un système de dépendances.
- Pour calculer une cellule il faut d'abord que toutes les cellules dont elle dépend aient été calculées.

```
1  Function PDD(m_in : mesh, r :func, f_update :func) : mesh is
2      mesh m_out ; // The output value
3      mesh n_d ; // The dependency for each cell
4      queue O ← ∅;
5      foreach Cell c ∈ G do
6          n_d(c) = 0;
7      foreach Cell c ∈ D do
8          n_d(c) = f_dep(c, m_in, r);
9          if n_d(c) == 0 then
10             O ← O ∪ c ;
11     bool over = false;
12     while (not over) do
13         while O ≠ ∅ do
14             c ← pop(O);
15             S_c = f_update(c, m_in, m_out);
16             foreach cell c' ∈ S_c do
17                 n_d(c)− = 1;
18                 if n_d(c) == 0 and c ∈ D then
19                     O ← O ∪ c;
20         Exchange_Ghosts(m_out);
21         Send({n_d(c)|c ∈ G});
22         Receive({n_d^up(c)|c ∈ O);
23         foreach cell c ∈ O do
24             n_d(c)+ = n^up(c);
25             if n_d(c) == 0 then
26                 O ← O ∪ c ;
27     return m_out;
```

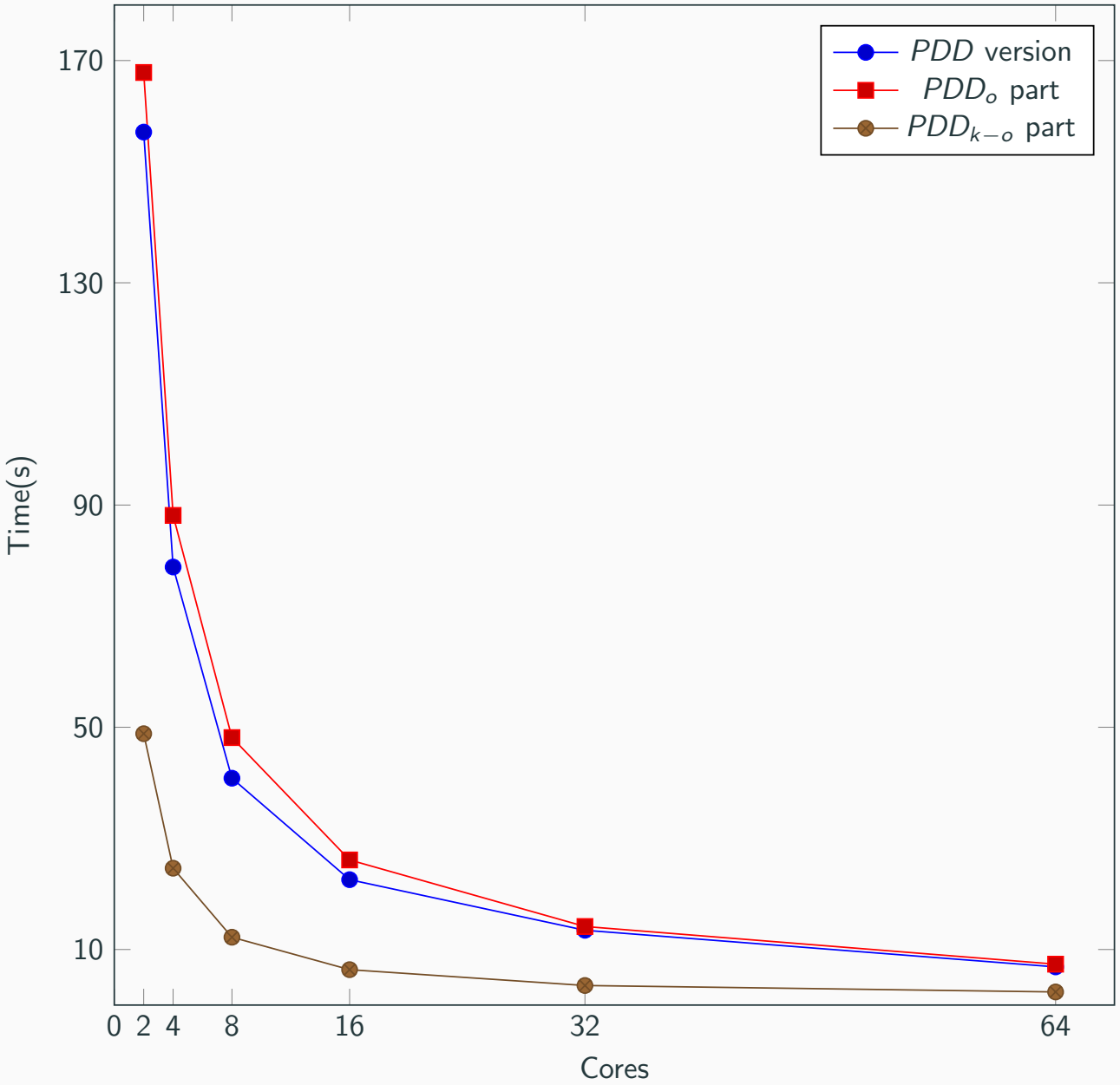
Algorithm 2: Pre-Determined Dependencies Pattern

Résultats

Résultats



Résultats


170
130
90
50
10
Time(s)
0 2 4 8 16 32 64
Cores
PDD version
PDD_o part
PDD_{k-o} part

Conclusion

Perspectives

- Améliorer les pattern et les porter sur d'autres types d'architectures (GPGPU, mémoire partagée).
- Créer un DSL pour les géomaticiens basé sur Python

Questions ?