Partitionnement de maillage sur une grille de calcul

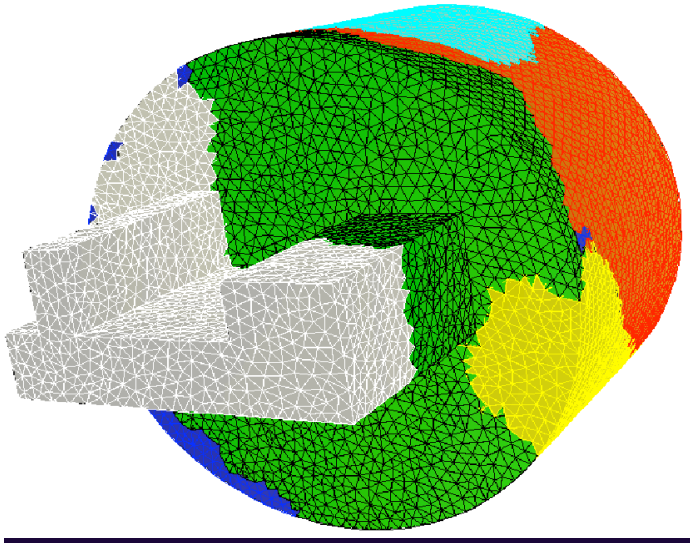
Youssef Mesri(*), Hugues Digonnet(**), Hervé Guillard(*)

Youssef.Mesri@sophia.inria.fr, Hugues.Digonnet@ensmp.fr, Herve.Guillard@sophia.inria.fr,

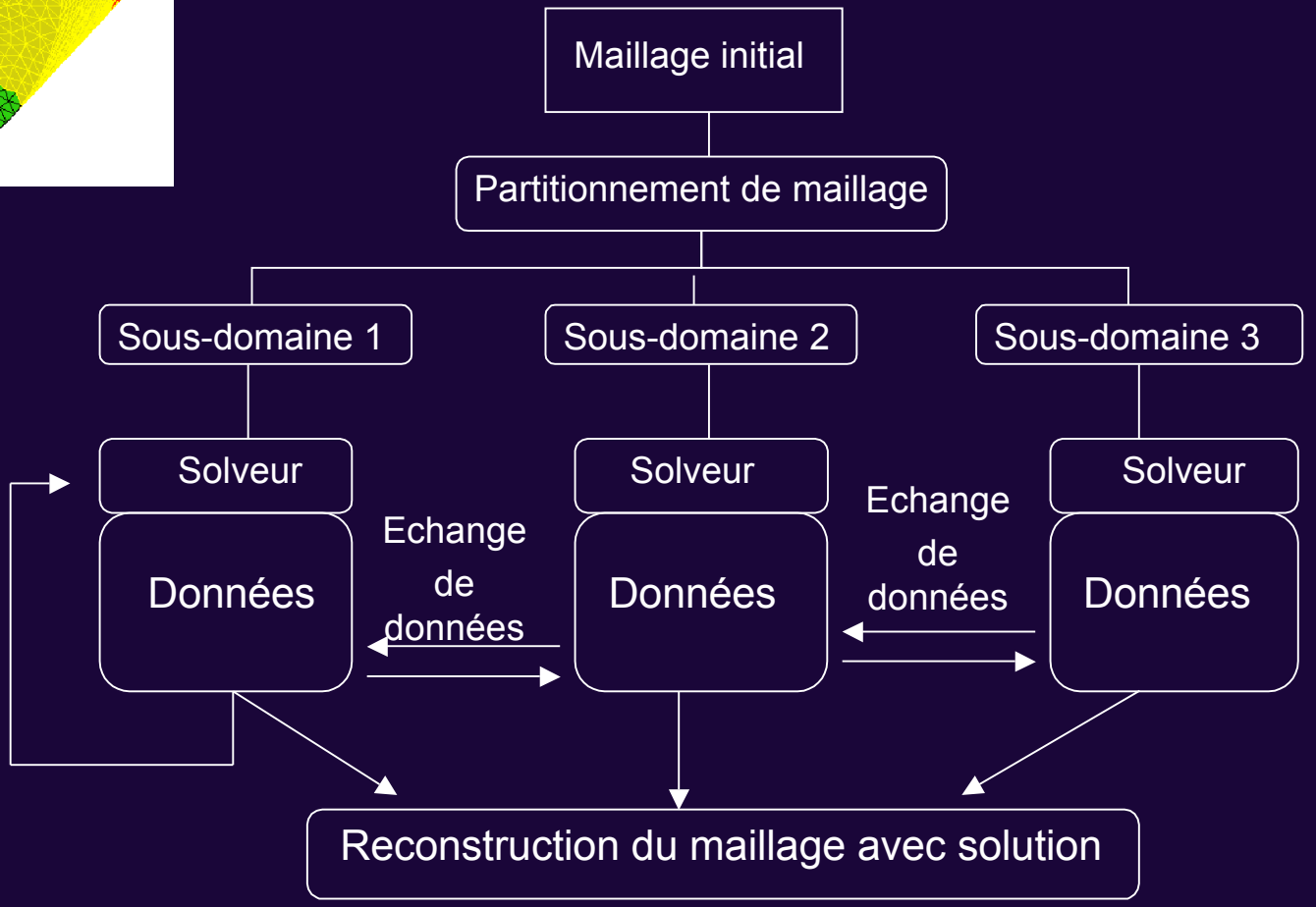
(*) Projet SMASH INRIA Sophia-Antipolis

(**) Ecole Nationale des Mines de Paris Sophia-Antipolis

INRIA



Exemple de parallélisation d'une application EF-VF (modèle SPMD)



INRIA

Plan de la présentation

- Problématique
- Modèles application/architecture
- Algorithme de partitionnement
- Résultats et validations
  - Tests et exemples
  - Validation sur un code EF

Problématique

Recherche de la Meilleure Partition du maillage

- Meilleure: pour un certain critère: Fonction coût
- Fonction coût = temps parallèle de l'application

L'objectif ultime → minimiser cette fonction coût

Modèles de représentation du maillage et de l'architecture:

- Graphe représentant le maillage/application:

$W(V,E)$: le graphe pondéré représentant le maillage/application

$w(v)$: Le poids représentant la charge de travail associé à ce nœud

$w(\{u,v\})$: le poids représentant la dépendance des données entre les deux nœuds de cette arête

- Graphe complet représentant l'architecture:

$A(P,E)$: le graphe pondéré représentant l'architecture

s_p : La vitesse du processeur p

v_{pq} : la bande passante entre p et q

- Cas d'une architecture homogène:(clusters,calculateurs,…)

  Le graphe pondéré A(P,E) est réduit à un graphe simple: $s_p = cte$ et $v_{pq} = cte$

  NB : Le problème de placement "mapping" du graphe maillage sur le graphe architecture revient un problème de partitionnement du graphe maillage en p sous-graphes(p le cardinal de l'ensemble de processeurs)

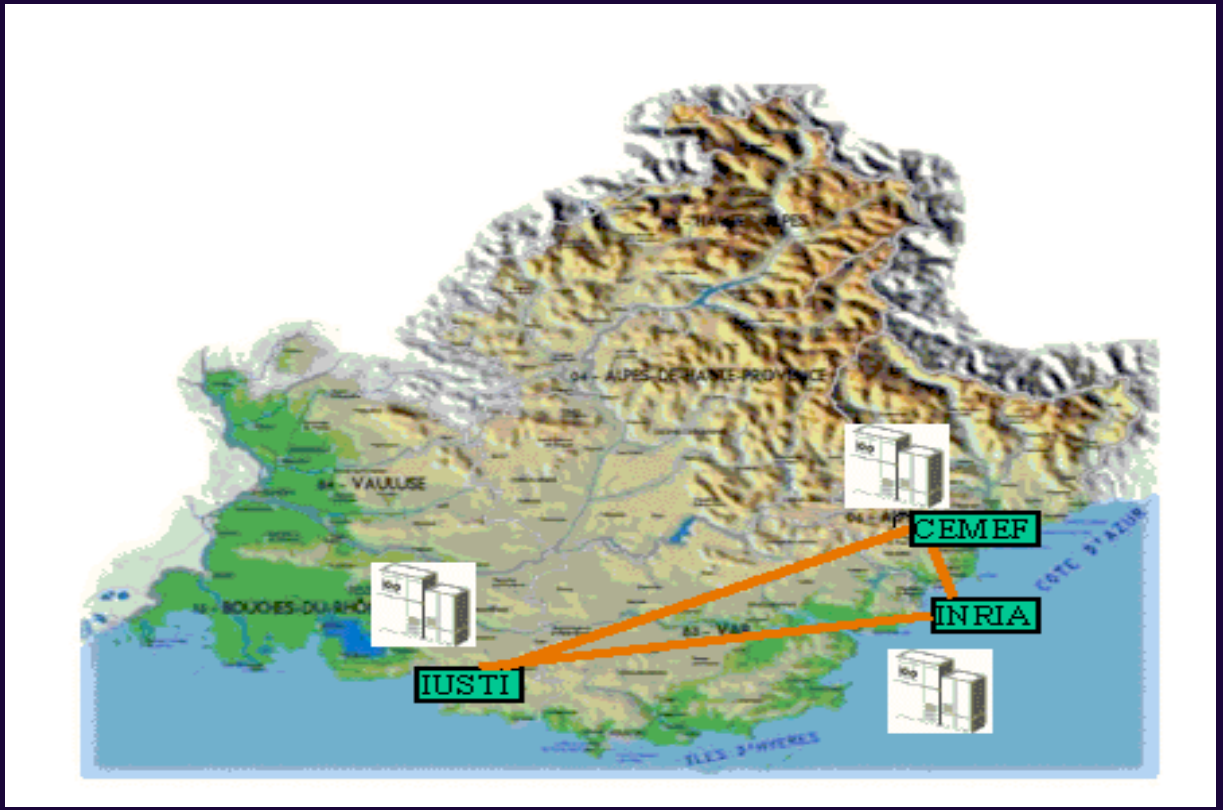
- Cas d'une architecture hétérogène:(grilles, constellation de clusters)

Exemple: la grille de calcul MecaGrid

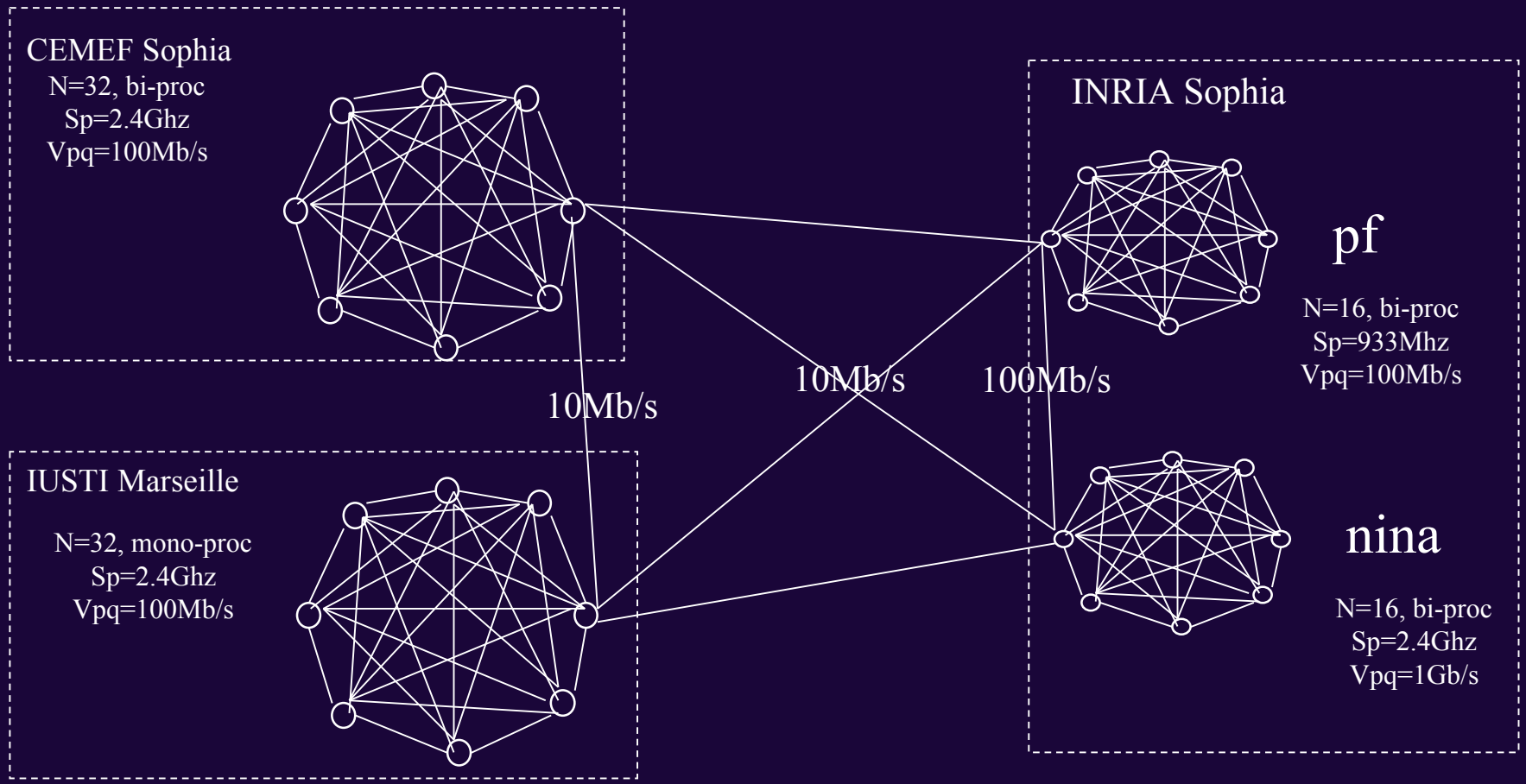
Projet MecaGrid
lancé en 11/2002

Connectant 3 sites de
la région PACA

Expérimentation sur des
applications des fluides
dynamiques
multi-matériaux dans une
grille de calcul



Le graphe architecture associé à la grille MecaGrid



Fonction coût

$$m : V \longrightarrow P$$

$m(v) = p$, si le noeud v est assigné au processeur p

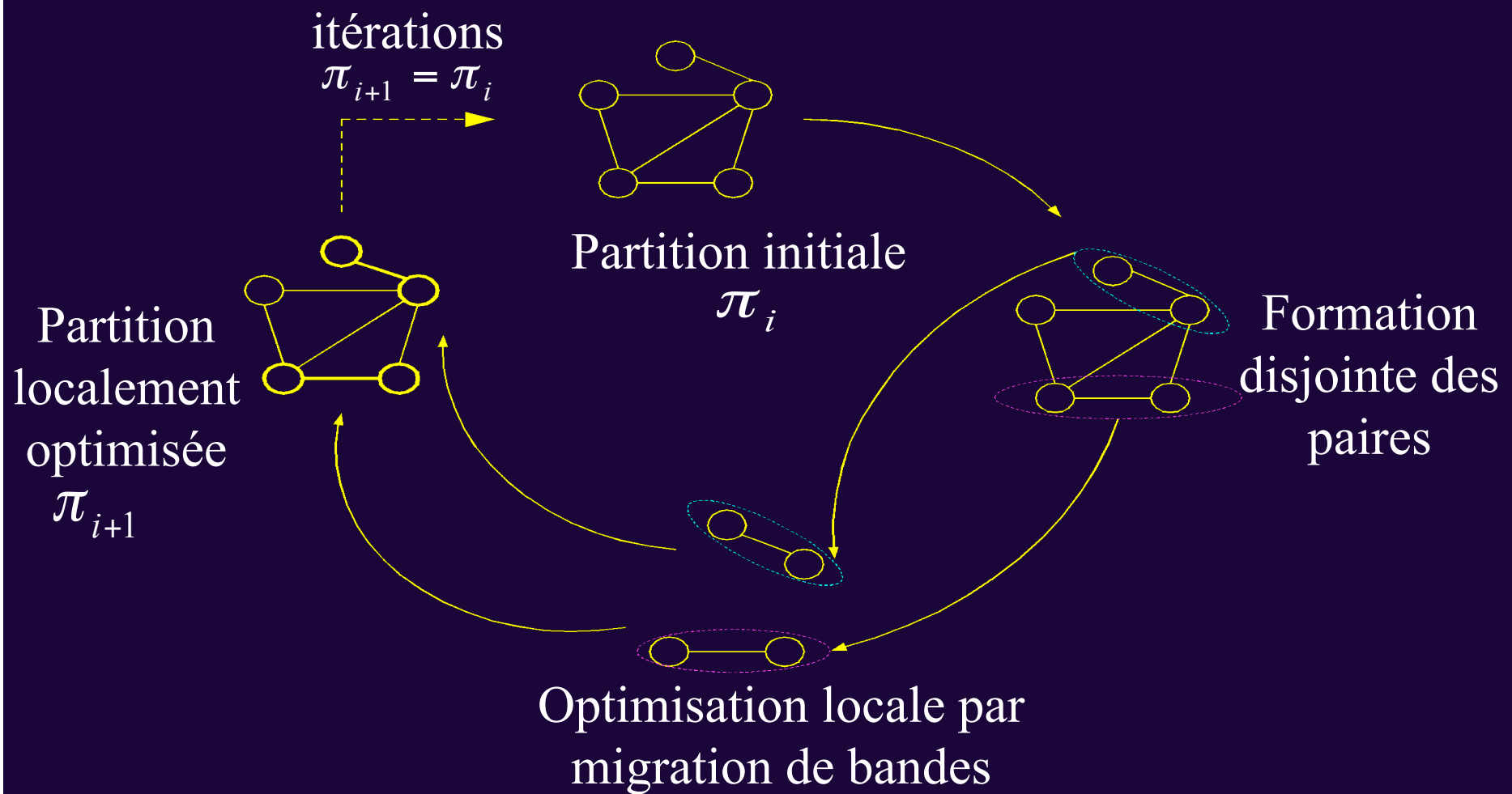
La fonction coût choisie: $F(W, A, m) = \max_{p \in V(A)} (t_p) + \max_{p \in V(A)} (c_p)$

$$t_p = \frac{c(p,m)}{s_p} \quad \text{avec} \quad c(p,m) = \sum_{v \in V(A), m(v)=p} w(v)$$

$$c_p = \sum_{q \in V(A) \neq p} c(\{p,q\},m) \text{ avec } c(\{p,q\},m) = \sum_{\substack{m(u)=p, m(v)=q \\ \{u,v\} \in E(W)}} w(\{u,v\})/v_{pq}$$

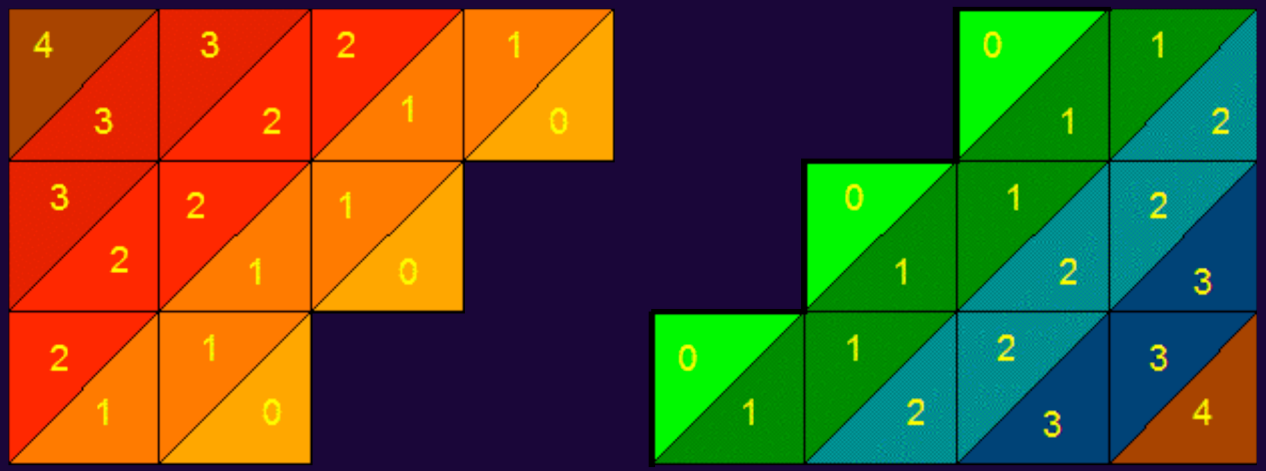
Problème: trouver le « mapping » m qui minimise F

Algorithme parallèle de partitionnement



migration de bandes sur le graphe application

Exemple sur un maillage 2D:



Formation des paires de processeurs

On considère le couple de processeurs (p,q),

La fonction coût associée à une partition initiale entre p et q est donnée par:

$$F_{0|pq} = \max(t_p, t_q) + c(\{p,q\},m) \quad \text{avec} \quad c(\{p,q\},m) = c(\{q,p\},m)$$

Améliorer la partition initiale:

- Pour p(resp. q), évaluer la fonction coût bande par bande jusqu'à trouver la bande associée au minimum de cette fonction noté $F^{p}_{\min|pq}$ sur p (resp. $F^{q}_{\min|pq}$ sur q)

- Calcul de la fonction amitié:

$$amitié(p,q) = \max(F_0|pq - F^{p}_{\min|pq}, F_0|pq - F^{q}_{\min|pq})$$

Formation des paires de processeurs

- La première paire formée est celle qui vérifie:

$$amitié(p,q) = \max(amitié(i,j)) \text{ pour tout } i \text{ et } j \text{ dans } V(A).$$

- La migration des bandes dans une paire formée (p,q):

Si $F^{p}_{\min|pq} < F^{q}_{\min|pq}$ (resp. $F^{q}_{\min|pq} < F^{p}_{\min|pq}$)

Les éléments et nœuds ayant une distance inférieur à celle de la bande associée au minimum $F^{p}_{\min|pq}$ (resp. $F^{q}_{\min|pq}$) vont être migré de p vers q (resp. de q vers p)

Stratégie de partitionnement sur une grille

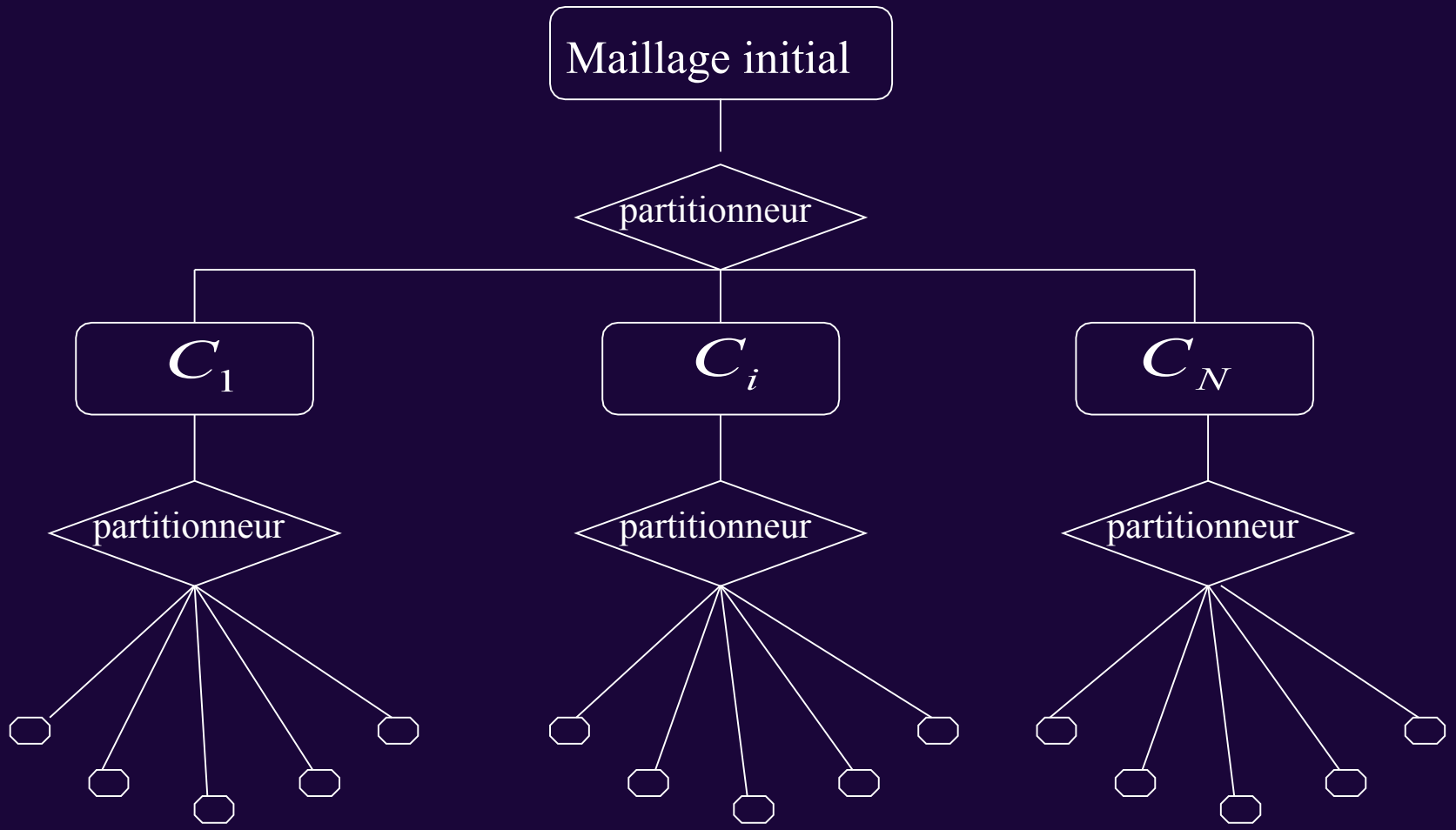
Utilisation du schéma précédent dans une boucle hiérarchique :

1- décomposer le maillage sur les clusters constituant la grille

2- sur chaque cluster, on repartitionne la partition associée sur ces processeurs

Objectif: réduire les interfaces de comminucations inter-clusters (réseau inter-clusters est plus lent qu'un réseau intra-clusters)

Illustration



Algorithme hiérarchique

1- boucle sur le nombre de clusters

Notation: $G = \{C_0,...,C_{N-1}\}$: l'ensemble des clusters constituant la grille

Soient p dans C_i , q dans C_j et $amitié(C_i,C_j) = 0$ pour tout C_i et C_j dans G

$$\begin{cases} if \ \ (C_i \neq C_j \quad and \quad amitié(C_i,C_j) = 0) \\ \qquad amitié(p,q) = \max(F_0\big|pq - F^p_{\min|pq}, F_0\big|pq - F^q_{\min|pq}) \\ \qquad amitié(C_i,C_j) = amitié(p,q) \\ else \quad amitié(p,q) = 0 \end{cases}$$

2- boucle sur les processeurs de chaque cluster

$\Pi = \{\pi_0,...,\pi_{N-1}\}$: l'ensemble des sous-domaines placés sur G

$C_i = \{p^i_0,...,p^i_{I-1}\}$: l'ensemble des processeurs constituant le cluster C_i

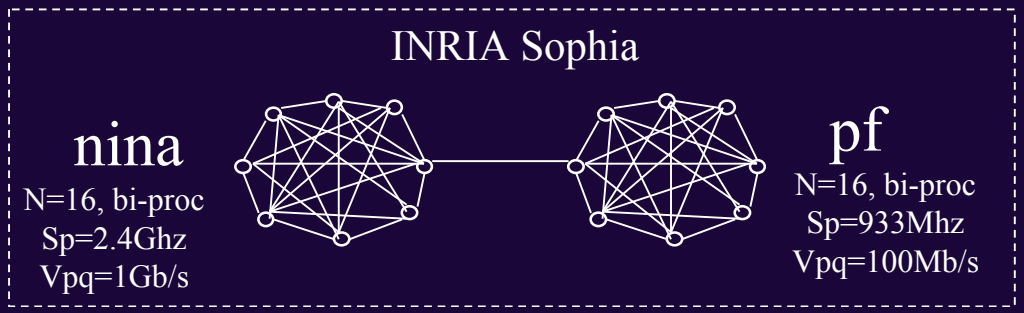
p dans C_i et q dans C_j

$$\begin{cases} if \quad \mathrm{C_i} = C_j \\ \qquad amitié(p,q) = \max(F_0\big|pq - F^p_{\min|pq}, F_0\big|pq - F^q_{\min|pq}) \\ else \qquad amitié(p,q) = 0 \end{cases}$$

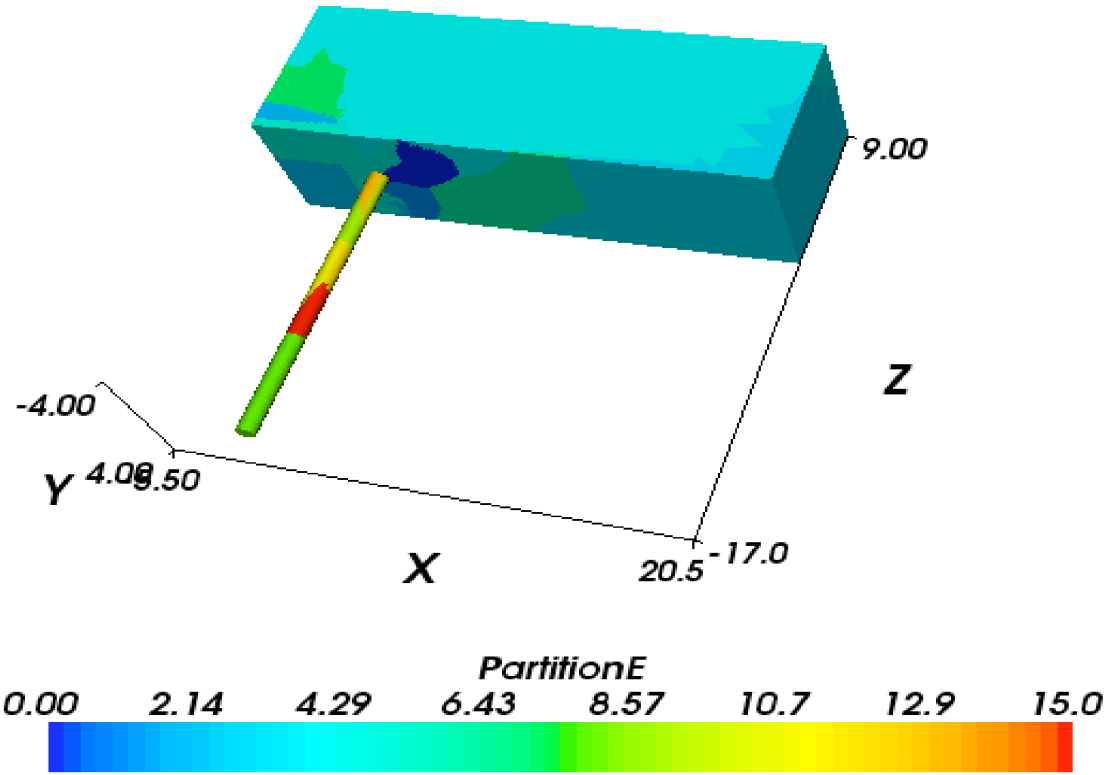
Exemple: maillage 3D de 398K noeuds et 3.8M d'éléments .

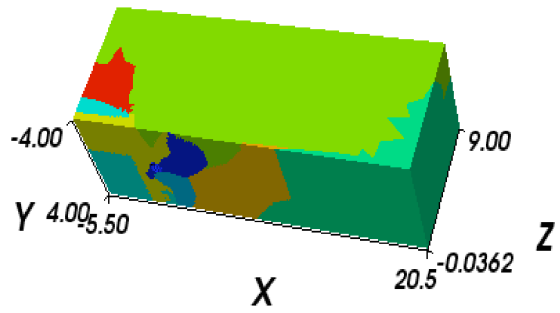
Les résultats correspondant à une évaluation de la fonction coût:

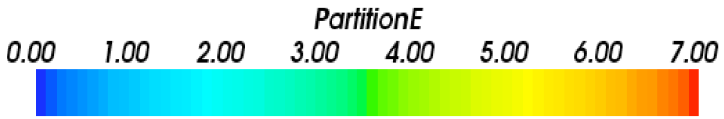
	1pf-1nina	2pf-2nina	4pf-4nina	8pf-8nina	16pf-16nina
F homogène	1.37e+06	699951	352804	266230	123602
F hétérogène	805537	421440	217025	137211	59278.7
Gain	41.3%	40%	40%	48.4%	53%

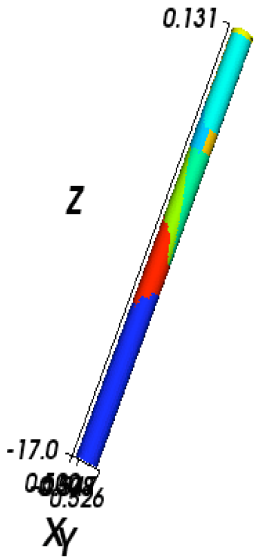


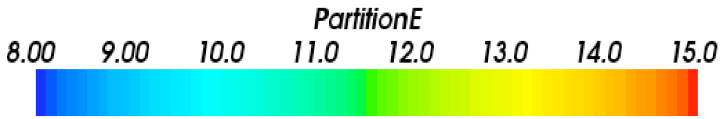
Maillage 3D de 398K nœuds partitionné sur 32 CPU hétérogènes




-4.00
9.00
Y 4.00
-5.50
Z
20.5
-0.0362
X


PartitionE
0.00 1.00 2.00 3.00 4.00 5.00 6.00 7.00


0.131
Z
-17.0
0.526
XY


PartitionE
8.00 9.00 10.0 11.0 12.0 13.0 14.0 15.0

Validation sur un code EF

- Utilisation du code parallèle éléments finis(Stokes) exécuté sur la grille(librairie Cimlib Olivier Basset et Hugues Digonnet):

Ce code résout des équations de type:

$$\begin{cases} \nabla.(2\eta\varepsilon(v)) - \nabla p = 0 \\ \nabla.v = 0 \end{cases}$$

Conditions aux limites :

$$\begin{cases} p = p_0 \quad \text{dans} \quad \Gamma_{\text{in}} \\ p = 0 \quad \text{dans} \quad \Gamma_{out} \\ v = 0 \quad \text{dans} \quad \Gamma - \Gamma_{out} - \Gamma_{\text{in}} \end{cases}$$

Résultats obtenus sur le cluster INRIA

16pf-16nina	résultats
Résolution(s)	339.97
Assemblage(s)	9.4065
Total(s)	349.42

Partition homogène

16pf-16nina	résultats
Résolution(s)	174.22
Assemblage(s)	6.61217
Total(s)	180.86

Partition optimisée

Réalisation d'un gain de 50%

Résultats obtenus sur les clusters INRIA-IUSTI

16iusti-16nina	résultats
Résolution(s)	561.278
Assemblage(s)	18.227
Total(s)	579.505

Partition homogène

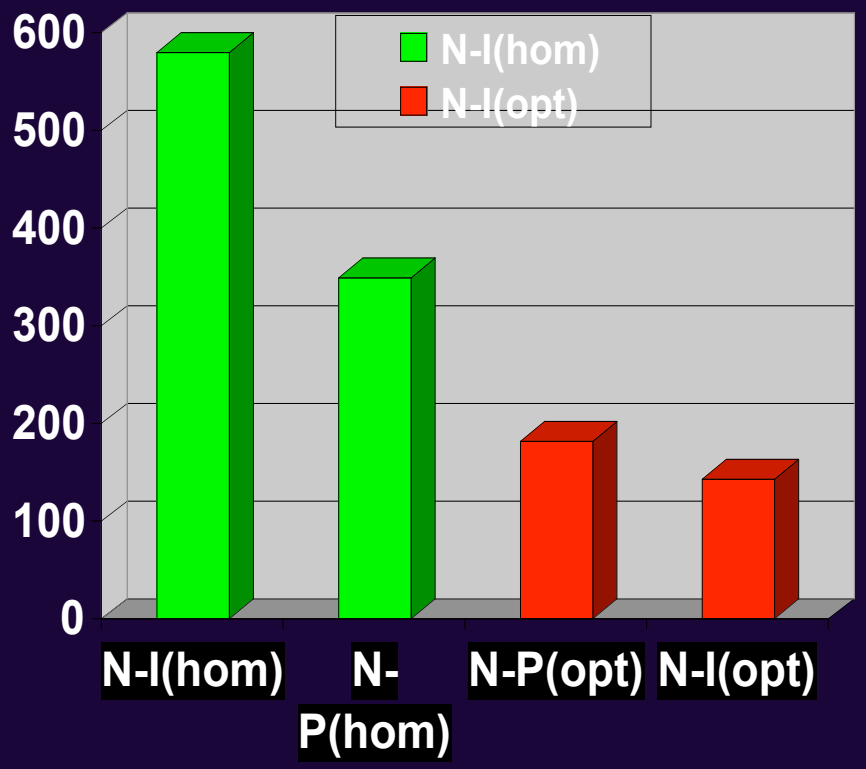
16iusti-16nina	résultats
Résolution(s)	137.924
Assemblage(s)	5.735
Total(s)	143.66

Partition optimisée

Réalisation d'un gain de 75%

Comparaison des résultats

clusters	Temps total
NINA-IUSTI(optimisée)	143.66
NINA-PF(optimisée)	180.8
NINA-PF(homogène)	349.42
NINA-IUSTI(homogène)	579.505



Gain de 75% !

Conclusion

✓ Présentation d'un nouveau schéma parallèle de partitionnement de maillages EF/VF sur des architectures hétérogènes type grilles de calcul

✓ Validation de notre méthode de partitionnement avec un code éléments finis exécuté sur la grille.

✓ Réalisation d'un gain en temps d'exécution de l'application qui dépasse 75%

Questions ?