Mesh Partitioning for Parallel CFD Application on a Grid

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

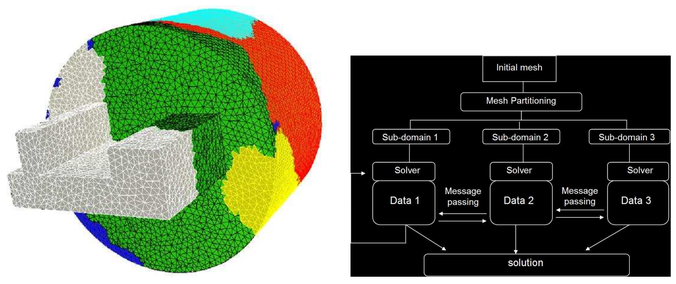
Youssef.Mesri@sophia.inria.fr, Hugues.Digonnet@ensmp.fr,herve.Guillard@sophia.inria.fr

Journées PaRISTIC, Labri, Bordeaux 21-23 November 2005

(*) SMASH Project INRIA Sophia-Antipolis

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

① CFD and parallelism SPMD model



② Overview

- Mesh partitioning = efficiency of parallel CFD application
- Homogeneous partitioning : load balancing
- Existing schemes:
  - ParMetis
  - Jostle
  - Chaco
- But, grid architectures:
  - heterogeneity in CPU
  - heterogeneity in network links

③ Mesh/Architecture models

- Mesh application graph:

$W(V, E)$: Workload graph
$w(v)$: The weight represents the work load associated to this node
$w(\{u, v\})$: The weight represents data dependency between two nodes

- Architecture graph: $A(P, E)$: Complete graph

S_p: Processing power of p
v_{pq}: Bandwidth between p and q

④ Example of architecture: The MecaGrid graph

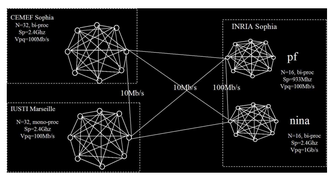
Mecagrid: project started 11/2002

Connects 3 sites in the PACA(Provence Alpes Côte d'Azur) region

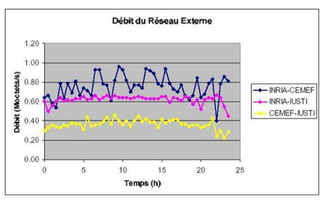
Performs experiments in grid computing applied to multimaterial fluid dynamics

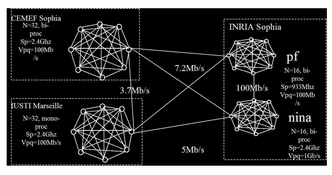
⑤ The MecaGrid: heterogeneous architecture of 162 processors

ⓐ The MecaGrid: Graph architecture



ⓑ The MecaGrid: mesured performances





⑥ Heterogeneous mesh partitioning

The mapping problem: Find a mesh partition that minimize CPU time.

Homogeneous (cluster architecture): Standard load balancing

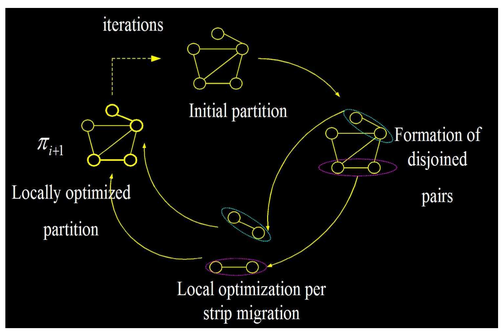
Heterogeneous (Grid):

$$F(W, A, m) = max(t_p)_{p \in V(A)} + max_{p \in V(A)}(C_p)$$

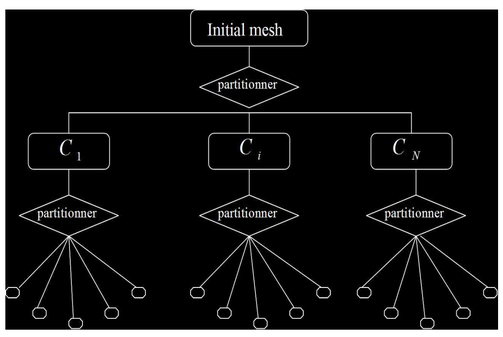
$$t_p = \frac{C(p,m)}{s_p} \quad \text{where} \quad C(p, m) = \sum_{\substack{v \in V(S) \\ m(v)=p}} w(v)$$

$$c_p = \sum_{\substack{i \in V(S) \\ p \neq i}} c_{pi} \quad \text{where} \quad c_{pi} = \frac{C(\{p,i\},m)}{v_{pi}}$$

⑦ Parallel partitioning algorithm



⑧ Hierarchical mesh partitionner



⑨ Application examples and validation

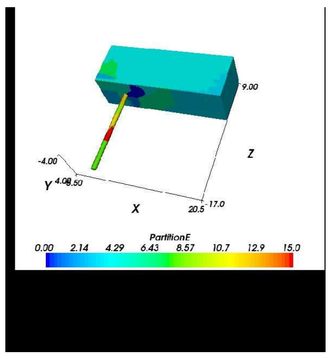
Tools: CIMlib library of CEMEF: a C++/MPI finite element library solving multimaterial flows (H. Digonnet, O. Basset, T. Coupez, H. Guillard)

① Exemple 1:

ⓐ Problem to solve:

$$(1) \qquad \begin{cases} \nabla.(2\eta\epsilon(v)) - \nabla p = 0 \\ \nabla.v = 0 \end{cases}$$

ⓑ Mesh: 3D mesh with 398K nodes and 3.8M elements



ⓒ Results on the INRIA clusters

Sop1-Sop2	CPU time
Resolution (s)	339.97
Assembling (s)	9.40651
Total (s)	349.42

Homogeneous partition

Sop1-Sop2	CPU time
Resolution (s)	174.22
Assembling (s)	6.61217
Total (s)	180.86

Optimized partition

Gain: 50%

ⓓ results on the INRIA-MARSEILLE clusters

16MRS-16Sop2	CPU time
Resolution (s)	561.278
Assembling (s)	18.227
Total (s)	579.505

Homogeneous partition

16MRS-16Sop2	CPU time
Resolution (s)	137.924
Assembling (s)	5.735
Total (s)	143.66

Optimized partition

Gain: 75%

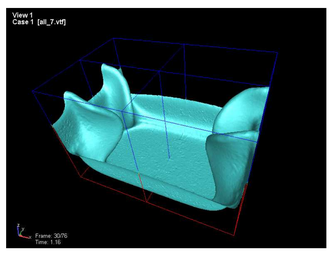
② Example 2:

ⓐ Multi-fluid unsteady incompressible NS equations coupled with transport equation

$$\begin{cases} \rho(\alpha)\frac{dv}{dt} - \nabla.(2\eta\epsilon(v)) + \nabla p = \rho(\alpha)g \\ \nabla.v = 0 \\ \frac{\partial\alpha}{\partial t} + \nabla\alpha.v = 0 \end{cases}$$

Viscosity & Density depend on a phase variable α

ⓑ Mesh: 3D mesh with 1.5M nodes and 8.7M elements



ⓒ NS and levelset + continuous Galerkin + Hamilton-Jacobi reinitialisation

Mesh: 1.5M nodes and **8.7M** elements 600 time steps
2 linear systems per increment of 6 millions of unknowns and of 1.5 millions respectively: a total of **4 billion and 500 millions of unknows**

Clusters(39 procs used)	CPU time
Sop1-Sop2-MRS(hom)	5 days and 6 hours
Sop1-Sop2-MRS (het)	3 days and 1 hour

Gain: 45%

Conclusion

- Presentation of a new parallel partitioning scheme for unstructured meshes on heterogeneous architectures.
- Real life validation of the method with FE code executed on the real life grid.
- Gain of more than of 75% on CPU time computation

ACI GRID

Liberté · Égalité · Fraternité
RÉPUBLIQUE FRANÇAISE
MINISTERE DE LA RECHERCHE

INRIA SOPHIA ANTIPOLIS Cemef

ACI-MECAGRID

http://www-sop.inria.fr/smash/mecagrid/public/