ANDRA

PaMPA : Parallel Mesh Partitioning and Adaptation

April 24, 2013

Contents

Contents

Context

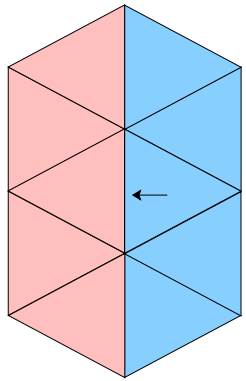
- Distributed meshes
- Subdomain decomposition
- Data exchanges between subdomains

Parallel mesh adaptation algorithms

- Parallel mesh generation (N. Chrisochoides 2005 [3])
  - Delaunay triangulation
  - Frontal method
  - Refinement by subdivisions (B. G. Larwood 2003 [7])
- Communication between subdomains:
  - Data migration
  - Matching algorithms

Problems induced by parallelism

- High complexity to parallelize remesh methods
  - Too much communication on boundaries
  - Boundaries not remeshed:
    - Local remeshing



Sequential algorithms

- Methods:
  - Moving nodes and remeshing locally (O. Hassan 2006 [6])
  - Coarse-grain parallel remeshing through multiple successive sequential remeshings (U. Tremel 2006 [8]):
    - Finding zones to remesh according to error estimator (T. Coupez 2000 [4])
    - Identifying zones to remesh in parallel
    - Remeshing zones in sequential
    - subdomain load balancing
- Main benefits:
  - Scalability of algorithms
  - Re-use of sequential codes

Contents

Generic dynamic load balancing

- General purpose (re)partitioning:
  - Jostle
  - Zoltan (K. Devine 2000 [5])
  - LB_Migrate (R. Chaube 2007 [2])
- Require a lot of extra coding

Specialized dynamic load balancing softwares

- Libraries:
  - DRAMA (A. Basermann 2000 [1])
- Pros and cons:
  - Mainly interfaced with solvers
  - Data structures based on meshes

Contents

Existing tools for handling unstructured meshes

- Partitioners:
  - Chaco
  - MeTiS
  - Mondriaan
  - Patoh
  - Scotch
  - Zoltan
- Intermediate:
  - DRAMA
  - PaMPA
  - PHG
- Advanced:
  - Arcane

Contents

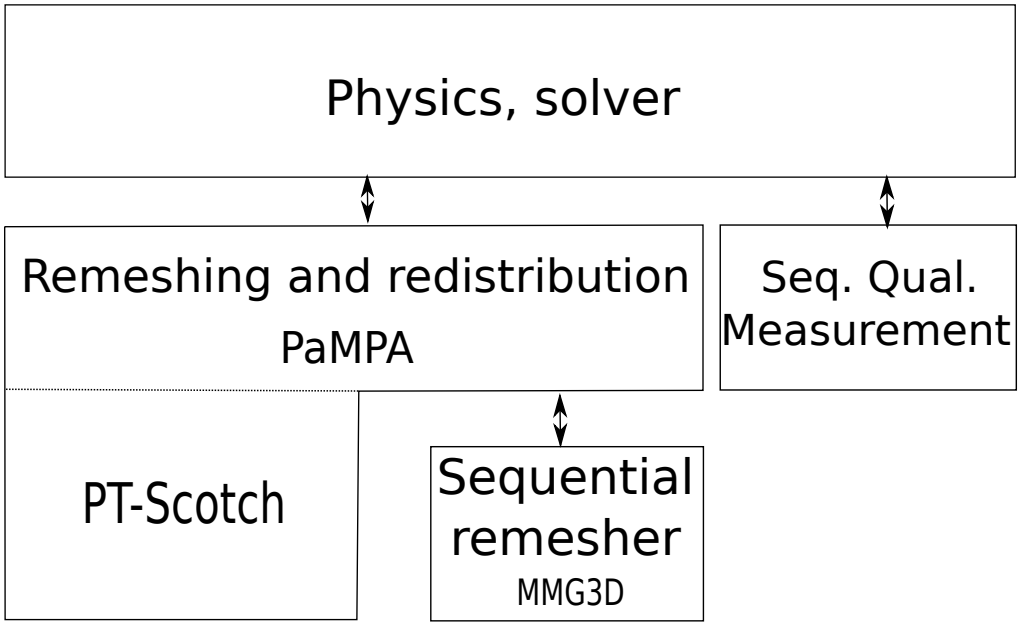
Common needs of solvers regarding meshes

- Handling of mesh structures
- Distribution of meshes across the processors of a parallel architecture
  - Handling of load balance
- Data exchange across neighboring entities
- Iteration on mesh entities
  - Entities of any kind: e.g. elements, faces, edges, nodes, ...
  - Entity sub-classes: e.g. regular or boundary faces, ...
  - Inner or frontier entities with respect to neighboring processors
  - Maximization of cache effects thanks to proper data reordering
- Dynamic modification of mesh structure
  - Dynamic redistribution
- Adaptive remeshing

Contents

What is PaMPA

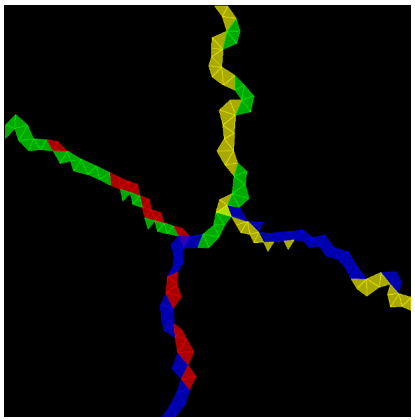
- PaMPA: "Parallel Mesh Partitioning and Adaptation"
- Middleware library managing the parallel repartitioning and remeshing of unstructured meshes modeled as interconnected valuated entities
- The user can focus on his/her "core business":
  - Solver
  - Sequential remesher
    - Coupling with MMG3D provided for tetrahedra



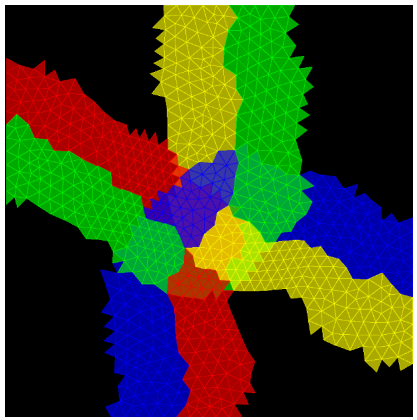
Features of version 0.2

- Overlap greater than 1

Overlap of size 1



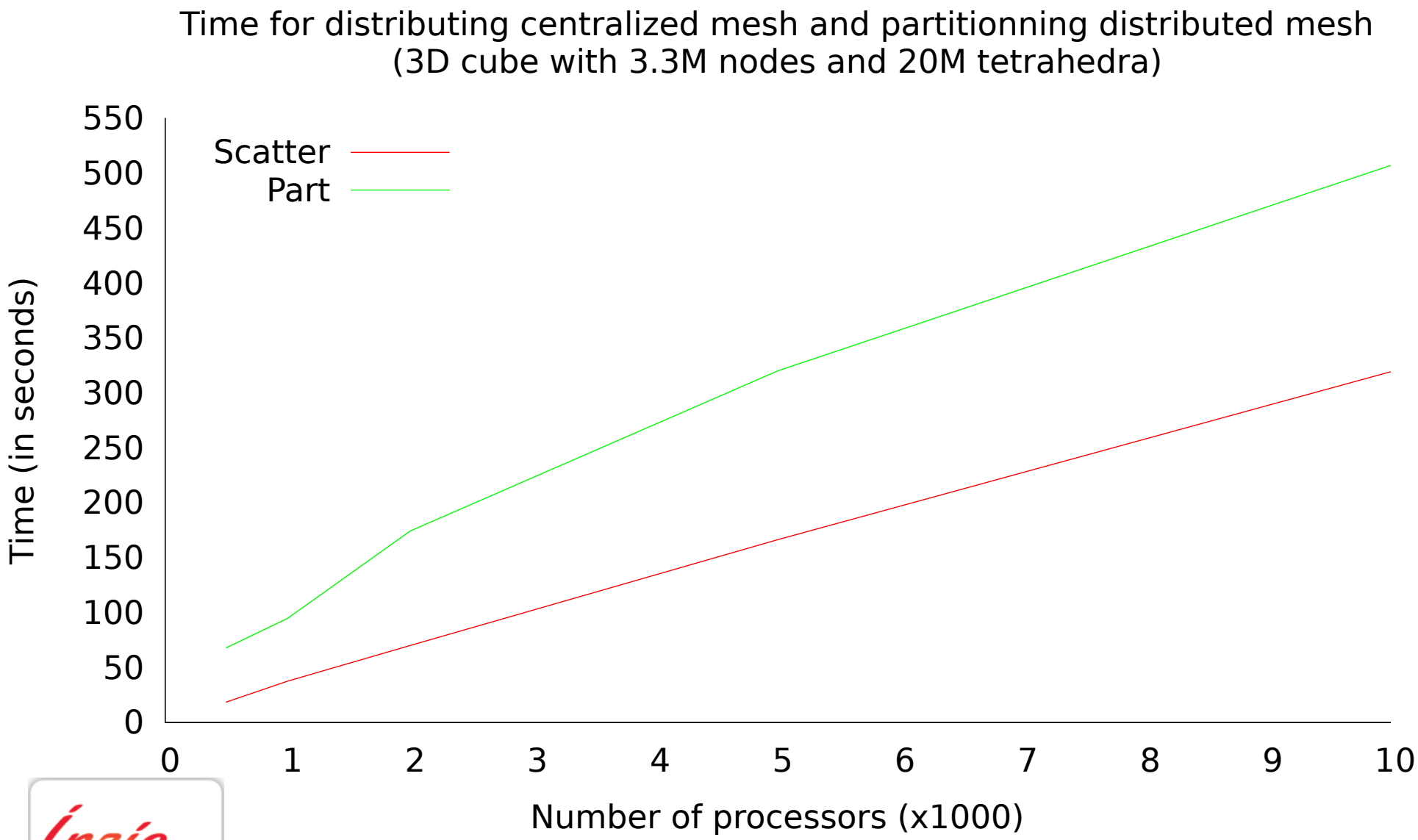
Overlap of size 10



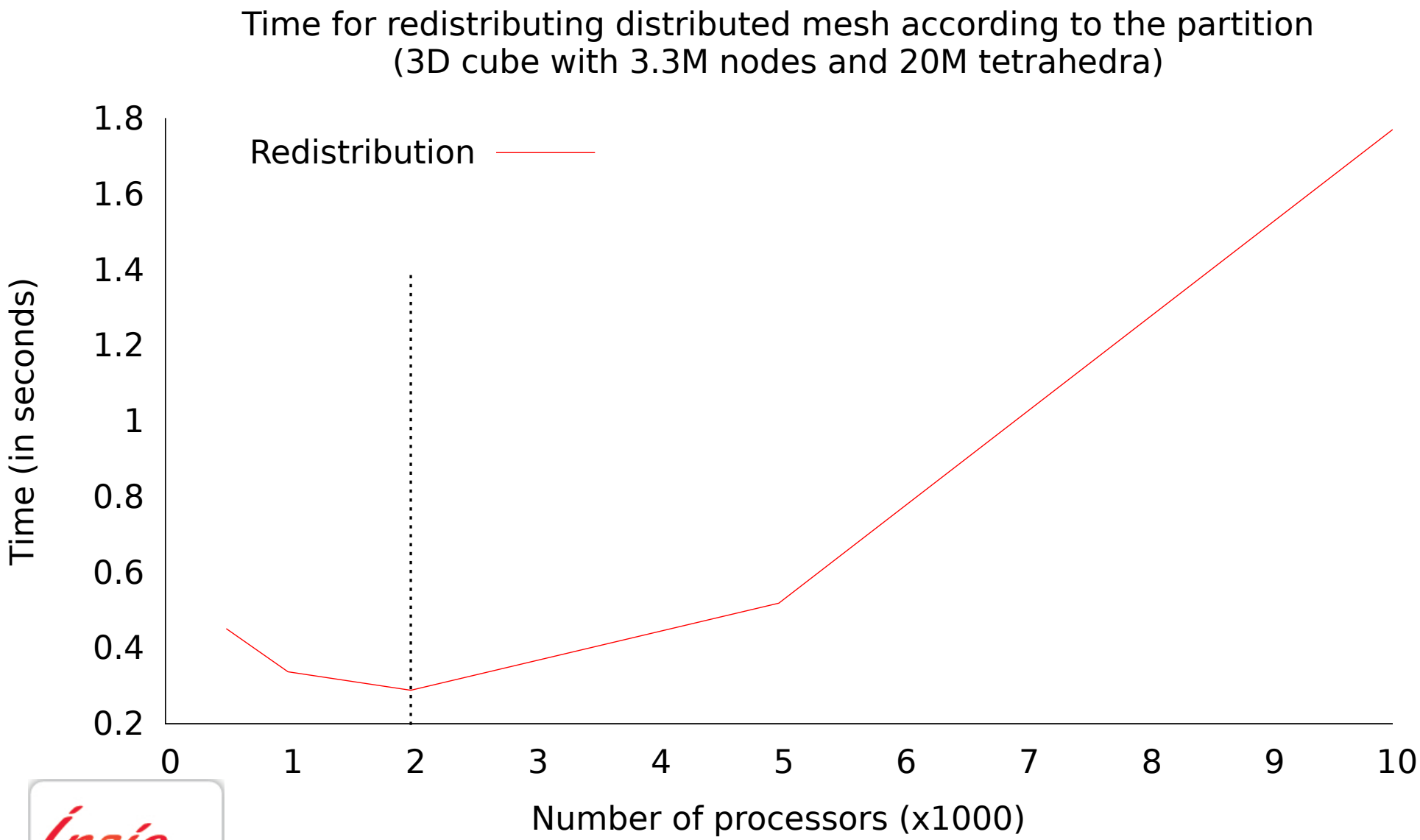
- Parallel I/O
- Parallel partitioning
- Parallel mesh adaptation based on sequential remesher

Contents

First results (1/3)



First results (2/3)



First results (3/3)

	Sequential	Parallel on 24 processors
Processor frequency	2,40 GHz	3,06 GHz
Elapsed time	09:24:37	00:12:23
Number of elements	75 529 964	77 579 275
Smallest edge length	0.2443	0.2178
Largest edge length	6.5983	6.4553
Worst element quality	144.7514	78.4903
Element quality between 1 and 2	99.64%	99.46%
Edge length between 0.71 and 1.41	97.71%	97.74%

Contents

Work in progress

- Release of version 0.2
  - Available soon from Inria Gforge
  - Licensed under GPL
- Quality of parallel adapted meshes
- Periodic meshes

Contents

Upcoming features

- Code industrialisation
- Mesh definition with a grammar
- Face orientation and displacement
- Unbreakable relations
  - Partitioner will not cut these edges
  - E.g. to implement DG methods
- Multi-grid meshes
- Parallel I/O with HDF5
- Parallel mesh adaptation scalability

Thank you


A. Basermann, J. Clinckemaillie, T. Coupez, J. Fingberg, H. Digonnet, R. Ducloux, J.-M. Gratien, U. Hartmann, G. Lonsdale, B. Maerten, D. Roose, and C. Walshaw.
Dynamic load balancing of finite element applications with the drama library, 2000.


R. Chaube, R. L. Carino, and I. Banicescu.
Effectiveness of a dynamic load balancing library for scientific applications.
In *ISPDC '07: Proceedings of the Sixth International Symposium on Parallel and Distributed Computing*, page 32, Washington, DC, USA, 2007. IEEE Computer Society.


N. Chrisochoides.
Parallel mesh generation.
In *in Numerical Solution of Partial Differential Equations on Parallel Computers*. Springer, 2005.


T. Coupez, H. Digonnet, and R. Ducloux.
Parallel meshing and remeshing.
Applied Mathematical Modelling, 25(2):153 – 175, 2000.
Dynamic load balancing of mesh-based applications on parallel.


K. Devine, B. Hendrickson, E. Boman, M. St. John, and C. Vaughan.
Design of dynamic load-balancing tools for parallel applications.
In *ICS '00: Proceedings of the 14th international conference on Supercomputing*, pages 110–118, New York, NY, USA, 2000. ACM.


O. Hassan, K. A. Sørensen, K. Morgan, and N. P. Weatherill.
A method for time accurate turbulent compressible fluid flow simulation with moving boundary components employing local remeshing.
In *International Journal for Numerical Methods in Fluids*, volume 53, pages 1243–1266. John Wiley & Sons, 2007.


B. G. Larwood, N. Weatherhill, O. Hassan, and K. Morgan.
Domain decomposition approach for parallel unstructured mesh generation.
In *International Journal for Numerical Methods in Engineering*, volume 58, pages 177–188, 2003.


U. Tremel, K. A. Sørensen, S. Hitzel, H. Rieger, O. Hassan, and N. P. Weatherill.
Parallel remeshing of unstructured volume grids for cfd applications.
In *International Journal for Numerical Methods in Fluids*, volume 53, pages 1361–1379. John Wiley & Sons, 2007.