SAND2008-7514C

What's new in Isorropia ?
Coloring & Ordering

Cédric Chevalier

Erik Boman, Lee Ann Fisk

(thanks to Karen Devine)

Sandia National Laboratories, NM, USA

Trilinos User's Group, Oct 21, 2008.

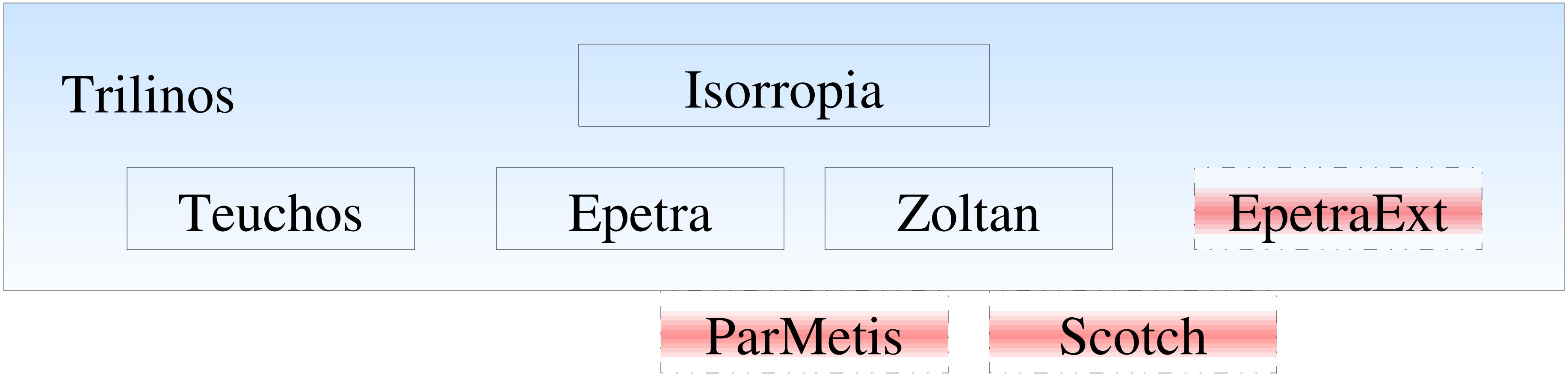
Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company,
for the United States Department of Energy's National Nuclear Security Administration
under contract DE-AC04-94AL85000.

What is Isorropia ?

- « *equilibrium* » in Greek:
  - First goal, providing load balancing in Trilinos
- Now, Isorropia is a toolbox to do combinatoric operations on matrices or graphs:
  - Partitioning and Load Balancing
  - Coloring
  - Sparse matrix ordering
- Mostly built on top of Zoltan

How does it work ?

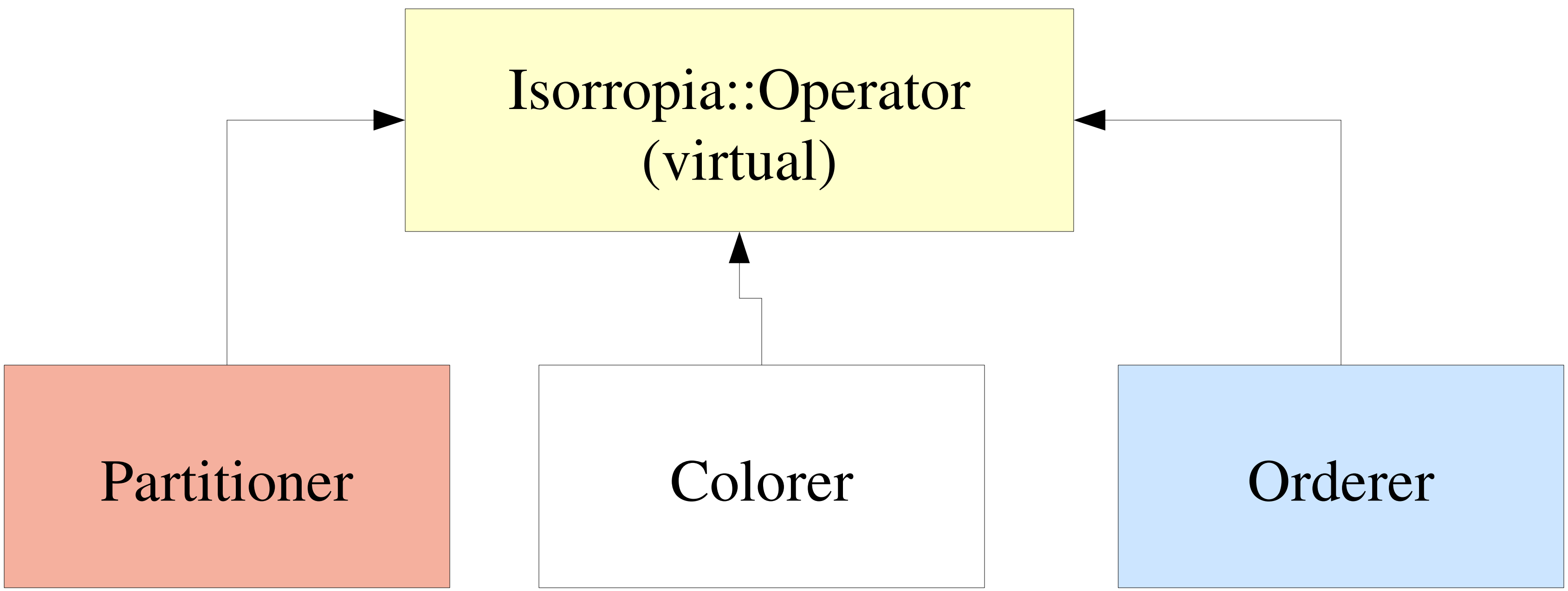
- Dependance on Zoltan, now part of Trilinos
- Dependance on Teuchos
- Some advanced features with EpetraExt



- To compile: ./configure --enable-isorropia
- Works in parallel and in serial

Software design (1)

- An abstract interface, not dependent of the partitioning software or the input type



An Isorropia::Operator is NOT an Epetra Operator !

Software design (2)

- An implementation of the previous interface:
  - Only Epetra input is supported but the design allows to do the same for other packages
  - Only Zoltan is supported but other software can be easily integrated

- This model will be extended by several partitioner class to do different kind of partitioning

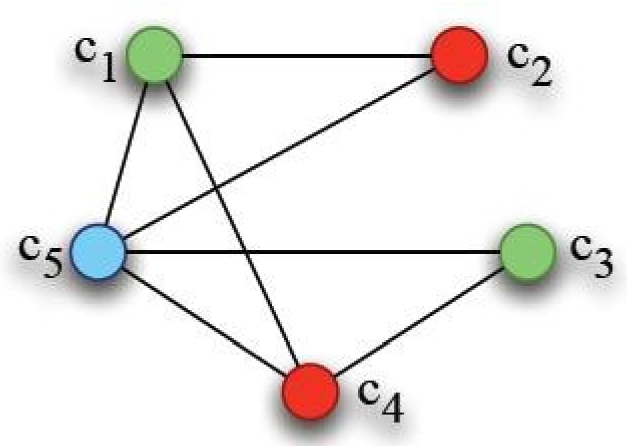
Coloring

Distance-1 Graph Coloring

- Problem (NP-hard)

  Color the vertices of a graph with as few colors as possible such that no two adjacent vertices receive the same color.

- Applications
  - Iterative solution of sparse linear sys
  - Preconditioners
  - Sparse tiling
  - Eigenvalue computation
  - Parallel graph partitioning

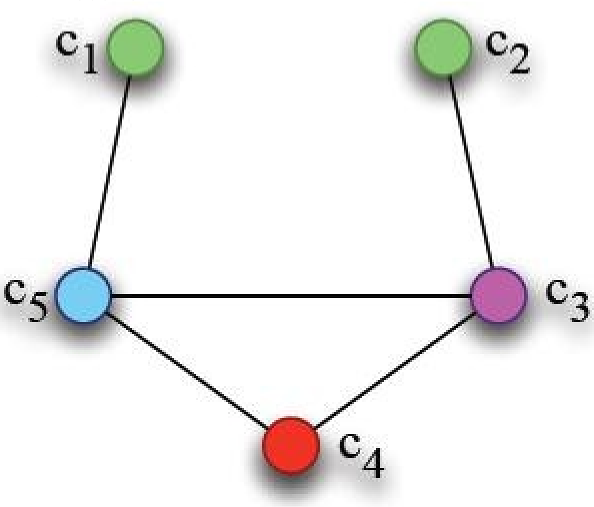


Distance-2 Graph Coloring

- Problem (NP-hard)

  Color the vertices of a graph with as few colors as possible such that a pair of vertices connected by a path on two or less edges receives different colors.

- Applications
  - Derivative matrix computation in numerical optimization
  - Channel assignment
  - Facility location
- Related problems
  - Partial distance-2 coloring
  - Star coloring



What Isorropia can do ?

- Can compute:
  - Distance-1 coloring
  - Distance-2 coloring
- Deals with:
  - Undirected graphs, distributed or not (Epetra_CrsGraph)
  - Symmetric matrices, distributed or not (Epetra_RowMatrix)
- Isorropia uses Zoltan's coloring capabilities
- Isorropia doesn't implement any coloring algorithms

Software interface

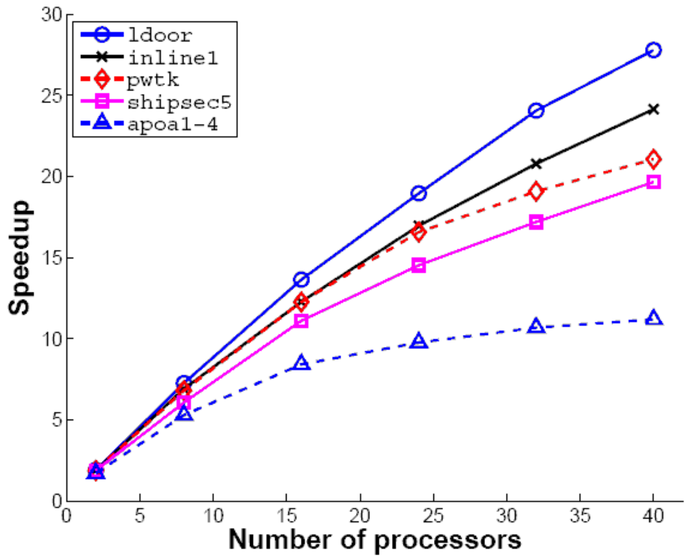
- One abstract class: `Colorer`
- The coloring can be perform by the method `color()`
- Object Colorer provides different accessors:
  - `operator[]`
  - `numColors()`: global number of colors used
  - `numElemsWithColor()`: number of local elements with the given color
  - `elemsWithColor()`: array of these local elements
  - `generateMapColoring()`: Epetra_MapColoring object (requires EpetraExt)

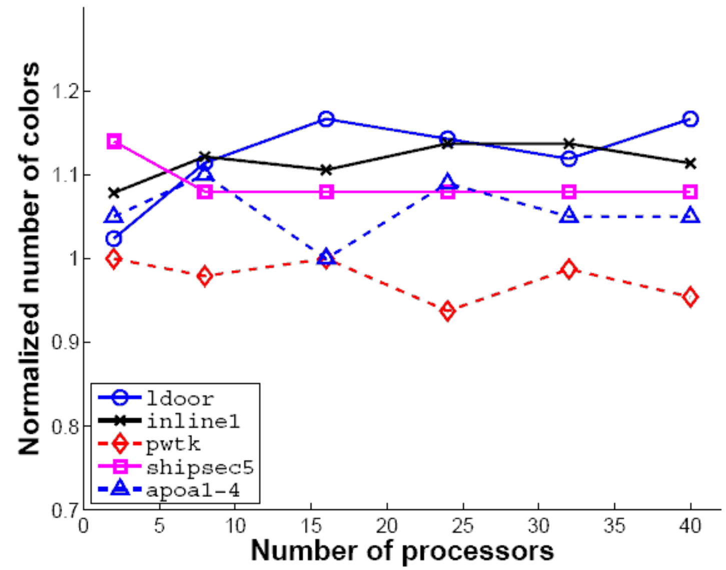
Example

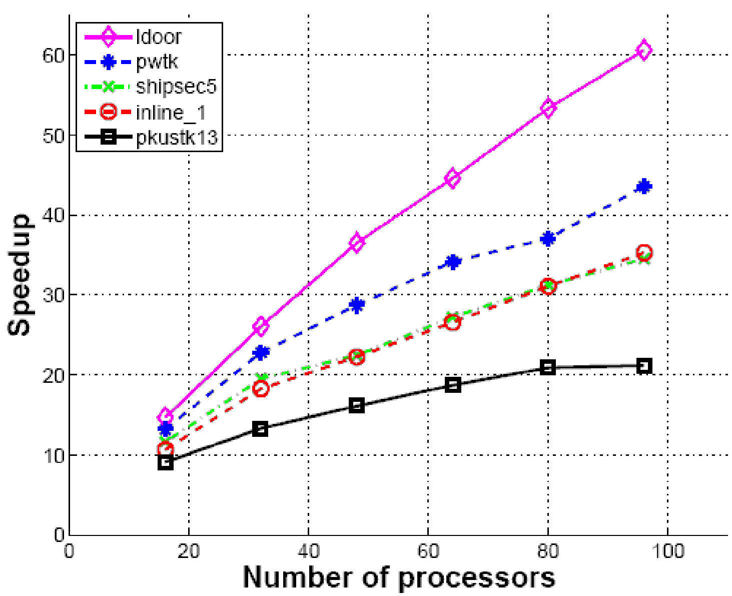
```
Epetra_CrsMatrix A;
   ...
Isorropia::Epetra::Colorer colorer(A,paramlist);
colorer.color(); /* Performs coloring */

 /* Parallel loop */
For (int c=1; c <= colorer.numColors() ; ++c){
   int length = colorer.numElemsWithColor(c);
   int *columns = new int[length];
   colorer.elemsWithColor(c, columns, length);
   /* Do some computations of columns of A of
     color c */
   ...
}
```

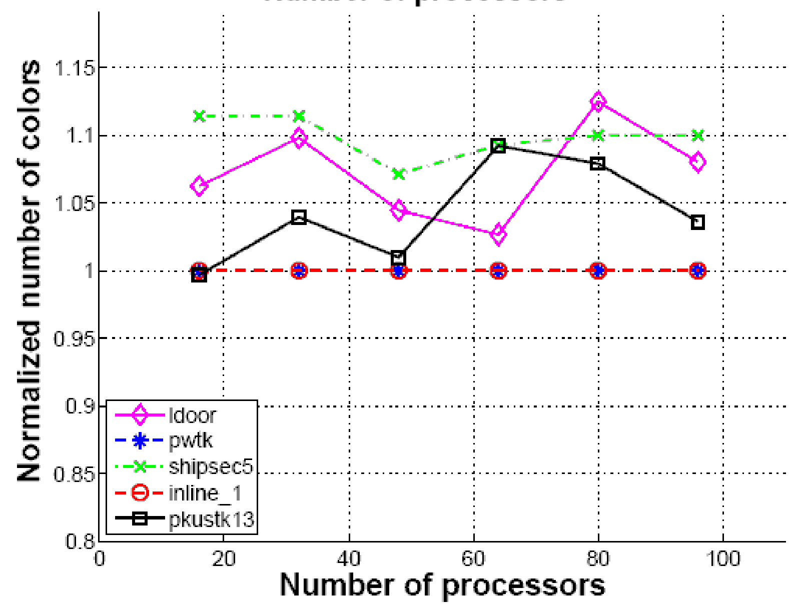
Sandia National Laboratories

Experimental Results









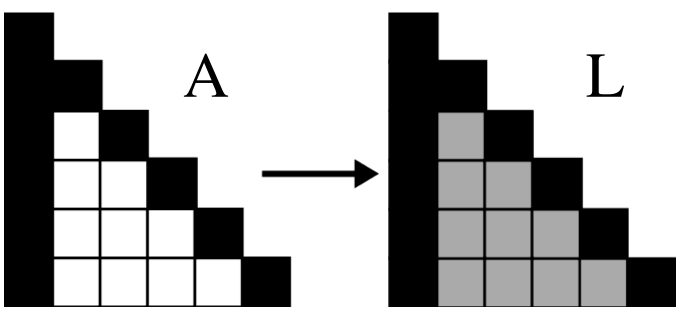
Future work

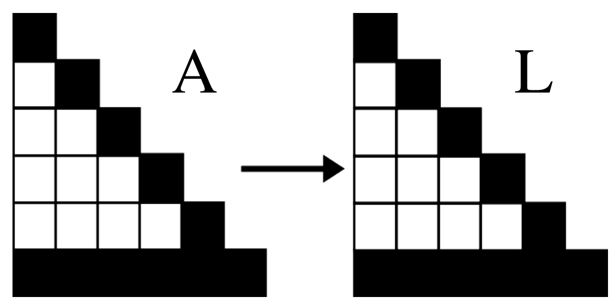
- May be extended to other colorings to be suitable for:
  - Automatic differenciation
  - Finite differences computations
- Possible interaction with other coloring software like ColPack (CSCAPES)

Sparse Matrix Ordering

Sparse Matrix Ordering problem

- When solving sparse linear systems with direct methods, non-zero terms are created during the factorization process ($A \rightarrow LL^t$, $A \rightarrow LDL^t$ or $A \rightarrow LU$).
- Fill-in depends on the order of the unknowns.
  - Need to provide fill-reducing orderings.



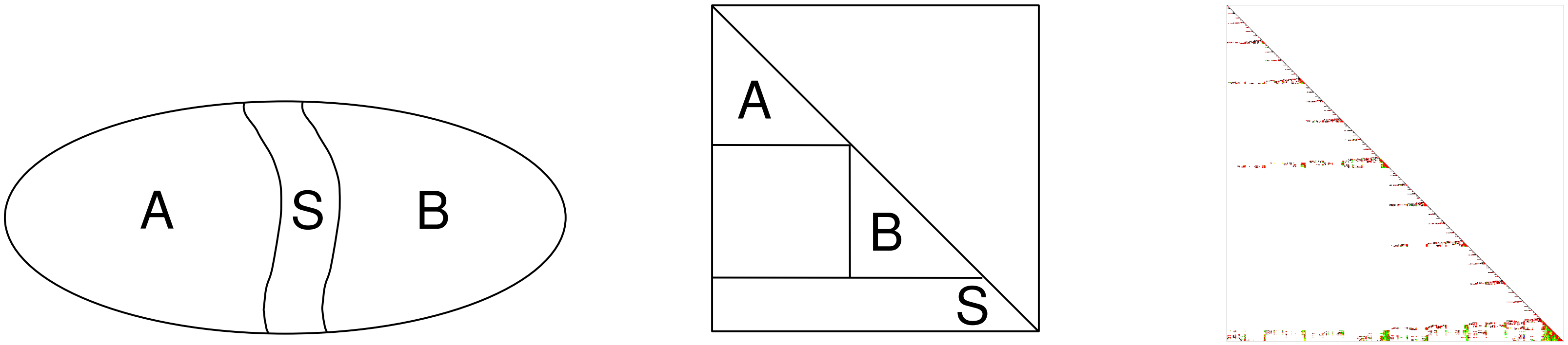


Fill Reducing ordering

- Combinatorial problem, depending on only the structure of the matrix *A*:
  - We can work on the graph associated with *A*.
- NP-Complete, thus we deal only with heuristics.
- Most popular heuristics:
  - Minimum Degree algorithms (AMD, MMD, AMF …)
  - Nested Dissection

Nested dissection (1)

- Principle [George 1973]
  - Find a vertex separator *S* in graph.
  - Order vertices of *S* with highest available indices.
  - Recursively apply the algorithm to the two separated subgraphs *A* and *B.*



Nested dissection (2)

- Advantages:
  - Induces high quality block decompositions.
    - Suitable for block BLAS 3 computations.
  - Increases the concurrency of computations.
    - Compared to minimum degree algorithms.
    - Very suitable for parallel factorization.
      - It's the scope here: parallel ordering is for parallel factorization.

Isorropia interface

- One abstract class: `Orderer`
- The coloring can be perform by the method `order()`
- Input : `Epetra_RowMatrix` or `Epetra_CrsGraph`
- Object `Orderer` provides different accessors:
  - `Operator[]`: associates to a Local ID the permuted Global ID
  - Only the permutation vector is available (for more advanced uses, Zoltan provides more informations)

How do we compute ordering ?

- Computations are done via Zoltan, but in third party libraries:
  - Metis
  - ParMetis
  - Scotch (PT-Scotch)
  - Easy to add another

Usages

- Focused on Cholesky factorization:
  - Limited to symmetric matrices
  - Can be use for symmetrised matrices (AA^t or $A+A^t$), but not automatically converted
- In the future, can deal with unsymmetric LU factorization:
  - Will be available also directly in Zoltan by using Hypergraph model

Example

```
Epetra_CrsMatrix A;              /* Square Matrix */
   ...
Isorropia::Epetra::Orderer orderer(A,paramlist);
orderer.order();          /* Performs ordering */

/* orderer[LID] is the new global position of LID
 according to the ordering */

/* Call direct solver:
   SuperLU/Mumps/UMFPack/Pastix ... */
```

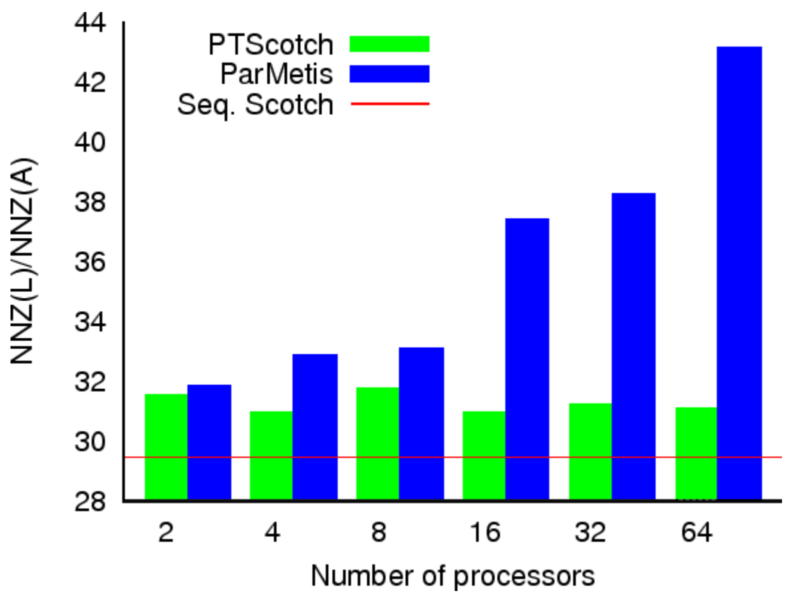
Experimental results (1)

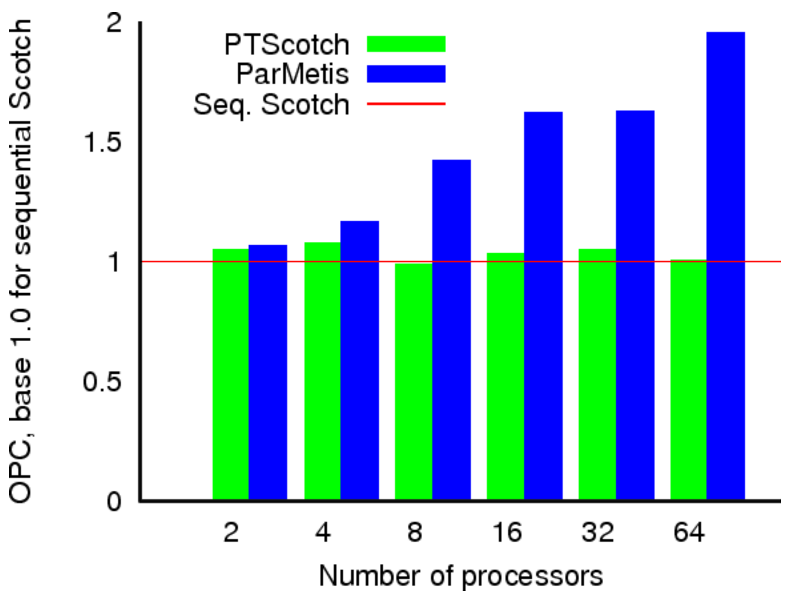
- Metric is OPC, the operation count of Cholesky factorization.
- Largest matrix ordered by PT-Scotch: 83 millions of unknowns on 256 processors (CEA/CESTA).
- Some of our largest test graphs.

Graph	Size (x1000)		Average degree	O_{SS}	Description
	|V|	|E|			
audikw1	944	38354	81.28	5.48E+12	3D mechanics mesh, Parasol
qimonda07	8613	29143	6.76	8.92E+10	Circuit simulation, Qimonda
23millions	23114	175686	7.6	1.29E+14	CEA/CESTA

Experimental results (2)

Test case	Number of processes					
	2	4	8	16	32	64
audikw1						
O_{PTS}	5.73E+12	5.65E+12	5.54E+12	5.45E+12	5.45E+12	5.45E+12
O_{PM}	5.82E+12	6.37E+12	7.78E+12	8.88E+12	8.91E+12	1.07E+13
t_{PTS}	73.11	53.19	45.19	33.83	24.74	18.16
t_{PM}	32.69	23.09	17.15	9.80	5.65	3.82





Experimental results (3)

- ParMETIS crashes for all other graphs.

Test case	Number of processes					
	2	4	8	16	32	64
Qimonda07						
O_{PTS}	-	-	5.80E+10	6.38E+10	6.94E+10	7.70E+10
t_{PTS}	-	-	34.68	22.23	17.30	16.62
23millions						
O_{PTS}	1.45E+14	2.91E+14	3.99E+14	2.71E+14	1.94E+14	2.45E+14
t_{PTS}	671.60	416.45	295.38	211.68	147.35	103.73

Future directions

- Add unsymmetric LU ordering (not available elsewhere)
- Direct integration in Amesos ?
  - Current interface is enough for SuperLU (even with the next LU ordering)
  - How it works for other solvers ?
  - Do users call directly their solver ?
- Adaptation to provide other matrix ordering ?
  - Bandwith reduction by RCM or GPS
- Add more powerful interface ? Like in Zoltan ?

General Summary

- Isorropia is ready to be in production
- Isorropia offers some interesting tools for helping/improving parallelisation in solvers
- Isorropia is wider than Partitioning:
  - Access to (at least a big part of) the power of Zoltan with minimal effort
  - See Erik Boman's talk about partitioning

The End

Thank You !