Towards a parallel analysis phase for a multifrontal sparse solver

Alfredo Buttari

INRIA *Rhône-Alpes*

5th International Workshop on
Parallel Matrix Algorithms and Applications (PMAA'08)
20-22 June 2008, Neuchtel, Switzerland

Sparse direct solvers: the three phases

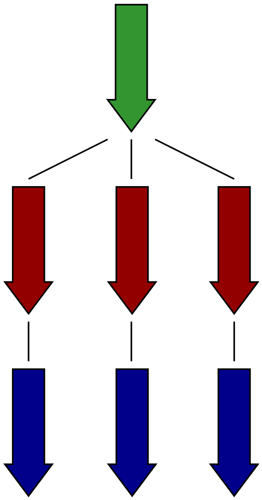
The solution of a sparse system with the MUMPS solver is achieved in three phases:

1. The **Analysis** phase
   - Scaling and Max-Trans
   - Fill-reducing pivot order
   - Symbolic factorization
2. The **Factorization** phase
   - $LU = PA$
3. The **Solve** phase
   - Forward/backward substitutions

Sparse direct solvers: the three phases

The solution of a sparse system with the MUMPS solver is achieved in three phases:

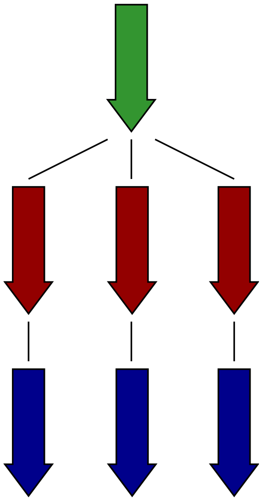
1. The **Analysis** phase
   - Scaling and Max-Trans
   - Fill-reducing pivot order
   - Symbolic factorization
2. The **Factorization** phase
   - $LU = PA$
3. The **Solve** phase
   - Forward/backward substitutions



Sparse direct solvers: the three phases

The solution of a sparse system with the MUMPS solver is achieved in three phases:

1. The **Analysis** phase
   - Scaling and Max-Trans
   - Fill-reducing pivot order
   - Symbolic factorization
2. The **Factorization** phase
   - $LU = PA$
3. The **Solve** phase
   - Forward/backward substitutions

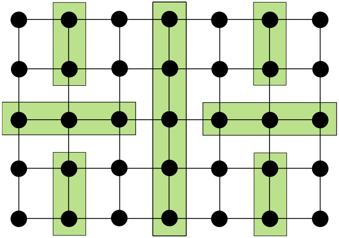


An approach to parallelization of the analysis

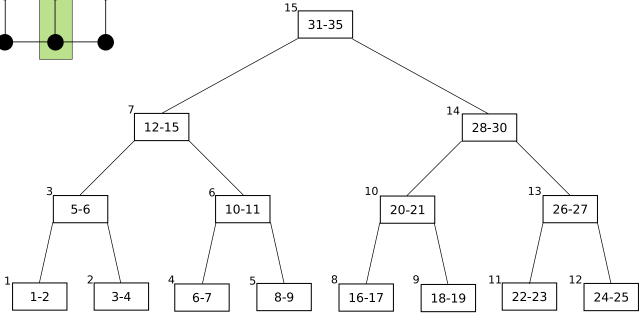
The parallelization is based on the coupling of a parallel graph ordering tool and a parallel symbolic factorization algorithm [Grigori et al., 2007] :

- PT-SCOTCH: "A tool for efficient parallel graph ordering" [Chevalier and Pellegrini, 2006]
- Quotient graph based symbolic factorization with restarting [George and Liu, 1980, Amestoy et al., 1996, Amestoy, 1997]

The PT-SCOTCH parallel ordering tool



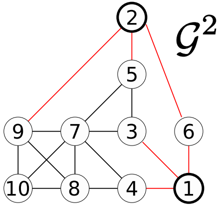
- Runs on any number of processors
- Nested Dissection does not stop at NP subdomains
- Quality of the ordering is virtually independent from NP



The symbolic factorization: quotient graphs

$A \in \Re^{nxn} \Rightarrow G = (V, E) \quad V = \{1, ..., n\} \quad E = \{(i, j) | a_{ij} \neq 0\}$

- **variables**:non-eliminated nodes
- **elements**: eliminated nodes



- **Quotient graphs**: elimination graphs G^k can be implicitly represented by quotient graphs $\mathcal{G}^k = (V^k, \bar{V}^k, E^k, \bar{E}^k)$
  - V^k: set of variables
  - $\bar{V}^k$: set of elements
  - $E^k \subseteq V^k \times V^k$: set of edges between variables
  - $\bar{E}^k \subseteq V^k \times \bar{V}^k$: set of edges between variables and elements

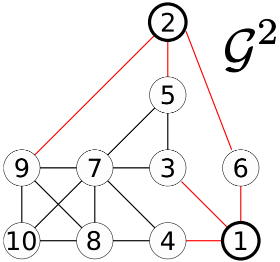
The symbolic factorization: quotient graphs

The symbolic factorization computes:

$$d_k = \left(\mathcal{A}_k^{k-1} \bigcup_{e \in \mathcal{E}_i^{k-1}} \mathcal{L}_e^{k-1} \right) \setminus \{k\}$$

where

- $\mathcal{A}_i^k = \{j : (i,j) \in E^k\} \subseteq V$
- $\mathcal{E}_i^k = \{e : (i,e) \in \bar{E}^k\} \subseteq \bar{V}^k$
- $\mathcal{L}_e^k = \{i : (i,e) \in \bar{E}^k\} \subseteq V^k$



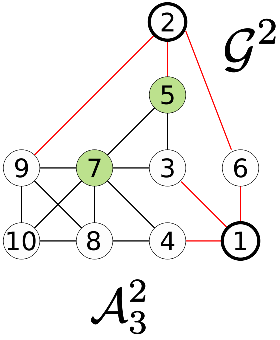
The symbolic factorization: quotient graphs

The symbolic factorization computes:

$$d_k = \left(\mathcal{A}_k^{k-1} \bigcup_{e \in \mathcal{E}_i^{k-1}} \mathcal{L}_e^{k-1} \right) \setminus \{k\}$$

where

- $\mathcal{A}_i^k = \{j : (i,j) \in E^k\} \subseteq V$
- $\mathcal{E}_i^k = \{e : (i,e) \in \bar{E}^k\} \subseteq \bar{V}^k$
- $\mathcal{L}_e^k = \{i : (i,e) \in \bar{E}^k\} \subseteq V^k$



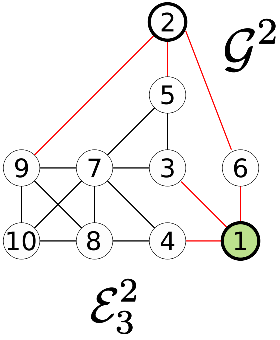
The symbolic factorization: quotient graphs

The symbolic factorization computes:

$$d_k = \left(\mathcal{A}_k^{k-1} \bigcup_{e \in \mathcal{E}_i^{k-1}} \mathcal{L}_e^{k-1} \right) \setminus \{k\}$$

where

- $\mathcal{A}_i^k = \{j : (i,j) \in E^k\} \subseteq V$
- $\mathcal{E}_i^k = \{e : (i,e) \in \bar{E}^k\} \subseteq \bar{V}^k$
- $\mathcal{L}_e^k = \{i : (i,e) \in \bar{E}^k\} \subseteq V^k$



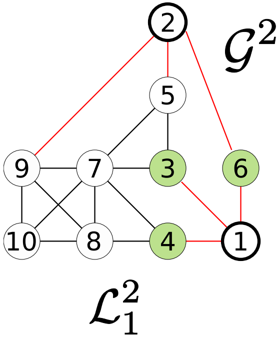
The symbolic factorization: quotient graphs

The symbolic factorization computes:

$$d_k = \left(\mathcal{A}_k^{k-1} \bigcup_{e \in \mathcal{E}_i^{k-1}} \mathcal{L}_e^{k-1} \right) \setminus \{k\}$$

where

- $\mathcal{A}_i^k = \{j : (i,j) \in E^k\} \subseteq V$
- $\mathcal{E}_i^k = \{e : (i,e) \in \bar{E}^k\} \subseteq \bar{V}^k$
- $\mathcal{L}_e^k = \{i : (i,e) \in \bar{E}^k\} \subseteq V^k$



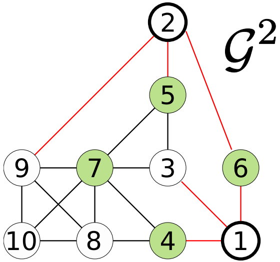
The symbolic factorization: quotient graphs

The symbolic factorization computes:

$$d_k = \left(\mathcal{A}_k^{k-1} \bigcup_{e \in \mathcal{E}_i^{k-1}} \mathcal{L}_e^{k-1} \right) \setminus \{k\}$$

where

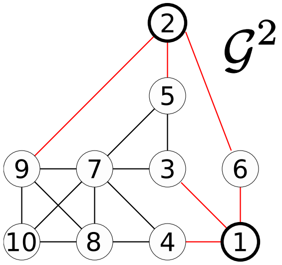
- $\mathcal{A}_i^k = \{j : (i,j) \in E^k\} \subseteq V$
- $\mathcal{E}_i^k = \{e : (i,e) \in \bar{E}^k\} \subseteq \bar{V}^k$
- $\mathcal{L}_e^k = \{i : (i,e) \in \bar{E}^k\} \subseteq V^k$



The symbolic factorization: quotient graphs

The usage of quotient graphs can benefit from a number of simplifications

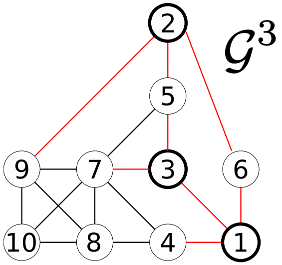
- **Nodes Absorption** All the elements in $\mathcal{E}_k^{k-1}$ will be removed from $\mathcal{G}^k$ and all the variables in $\mathcal{L}_e^{k-1}$ for each $e \in \mathcal{E}_k^{k-1}$ will be included in $\mathcal{L}_k^k$
- **Redundant Edges Elimination** Any edge (i, j) where $i, j \in \mathcal{L}_k^k$ will be suppressed



The symbolic factorization: quotient graphs

The usage of quotient graphs can benefit from a number of simplifications

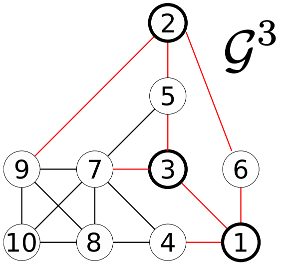
- **Nodes Absorption** All the elements in $\mathcal{E}_k^{k-1}$ will be removed from $\mathcal{G}^k$ and all the variables in $\mathcal{L}_e^{k-1}$ for each $e \in \mathcal{E}_k^{k-1}$ will be included in $\mathcal{L}_k^k$
- **Redundant Edges Elimination** Any edge (i, j) where $i, j \in \mathcal{L}_k^k$ will be suppressed



The symbolic factorization: quotient graphs

The usage of quotient graphs can benefit from a number of simplifications

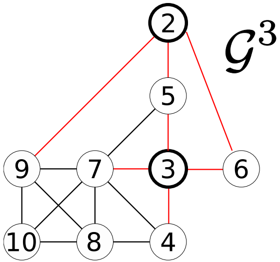
- **Nodes Absorption** All the elements in $\mathcal{E}_k^{k-1}$ will be removed from $\mathcal{G}^k$ and all the variables in $\mathcal{L}_e^{k-1}$ for each $e \in \mathcal{E}_k^{k-1}$ will be included in $\mathcal{L}_k^k$
- **Redundant Edges Elimination** Any edge (i, j) where $i, j \in \mathcal{L}_k^k$ will be suppressed



The symbolic factorization: quotient graphs

The usage of quotient graphs can benefit from a number of simplifications

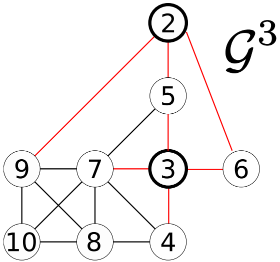
- **Nodes Absorption** All the elements in $\mathcal{E}_k^{k-1}$ will be removed from $\mathcal{G}^k$ and all the variables in $\mathcal{L}_e^{k-1}$ for each $e \in \mathcal{E}_k^{k-1}$ will be included in $\mathcal{L}_k^k$
- **Redundant Edges Elimination** Any edge (i, j) where $i, j \in \mathcal{L}_k^k$ will be suppressed



The symbolic factorization: quotient graphs

The usage of quotient graphs can benefit from a number of simplifications

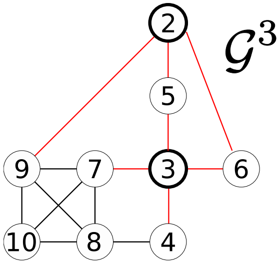
- **Nodes Absorption** All the elements in $\mathcal{E}_k^{k-1}$ will be removed from $\mathcal{G}^k$ and all the variables in $\mathcal{L}_e^{k-1}$ for each $e \in \mathcal{E}_k^{k-1}$ will be included in $\mathcal{L}_k^k$
- **Redundant Edges Elimination** Any edge (i, j) where $i, j \in \mathcal{L}_k^k$ will be suppressed



The symbolic factorization: quotient graphs

The usage of quotient graphs can benefit from a number of simplifications

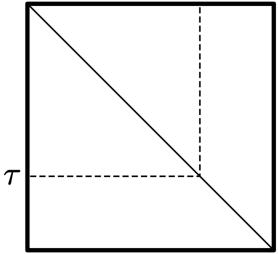
- **Nodes Absorption** All the elements in $\mathcal{E}_k^{k-1}$ will be removed from $\mathcal{G}^k$ and all the variables in $\mathcal{L}_e^{k-1}$ for each $e \in \mathcal{E}_k^{k-1}$ will be included in $\mathcal{L}_k^k$
- **Redundant Edges Elimination** Any edge (i, j) where $i, j \in \mathcal{L}_k^k$ will be suppressed



The symbolic factorization: restarting

The technique of restarting is based on a combination of left- and right-looking updates of the quotient graph:

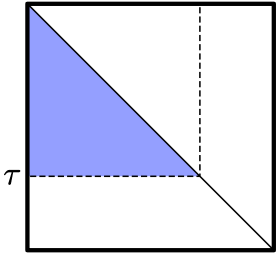
1. in pivotal steps $1, \ldots, \tau$ are processed and only the adjacency information for variables $1 - \tau$ is updated in a right-looking way
2. **restart**: the adjacency information of variables $\tau - n$ is updated with respect to elements $1 - \tau$ in a left-looking way
3. apply steps 1 and 2 recursively on variables $\tau - n$



The symbolic factorization: restarting

The technique of restarting is based on a combination of left- and right-looking updates of the quotient graph:

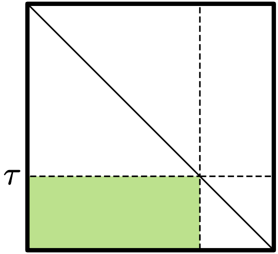
1. in pivotal steps $1, \ldots, \tau$ are processed and only the adjacency information for variables $1 - \tau$ is updated in a right-looking way
2. **restart**: the adjacency information of variables $\tau - n$ is updated with respect to elements $1 - \tau$ in a left-looking way
3. apply steps 1 and 2 recursively on variables $\tau - n$



The symbolic factorization: restarting

The technique of restarting is based on a combination of left- and right-looking updates of the quotient graph:

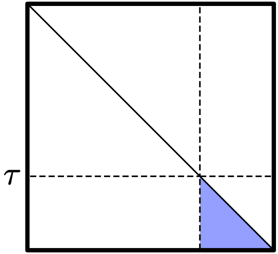
1. in pivotal steps $1, ..., \tau$ are processed and only the adjacency information for variables $1 - \tau$ is updated in a right-looking way
2. **restart**: the adjacency information of variables $\tau - n$ is updated with respect to elements $1 - \tau$ in a left-looking way
3. apply steps 1 and 2 recursively on variables $\tau - n$



The symbolic factorization: restarting

The technique of restarting is based on a combination of left- and right-looking updates of the quotient graph:

1. in pivotal steps $1, \ldots, \tau$ are processed and only the adjacency information for variables $1 - \tau$ is updated in a right-looking way
2. **restart**: the adjacency information of variables $\tau - n$ is updated with respect to elements $1 - \tau$ in a left-looking way
3. apply steps 1 and 2 recursively on variables $\tau - n$



Coupling the two steps

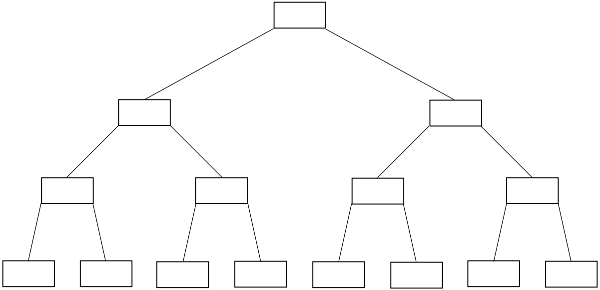
The ordering and the symbolic factorization are performed on the graph built from $|A| + |A^T|$

- run PT-SCOTCH on the graph and get a separator's tree
- leaf subtrees are processed independently by processors
- the top of the tree is processed sequentially by a "root" node

Coupling the two steps

The ordering and the symbolic factorization are performed on the graph built from $|A| + |A^T|$

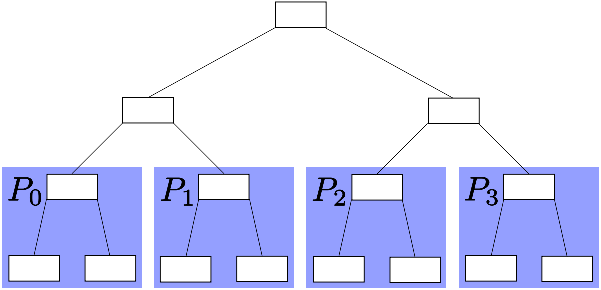
- run PT-SCOTCH on the graph and get a separator's tree
- leaf subtrees are processed independently by processors
- the top of the tree is processed sequentially by a "root" node



Coupling the two steps

The ordering and the symbolic factorization are performed on the graph built from $|A| + |A^T|$

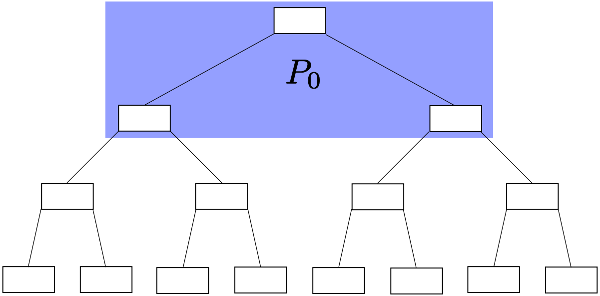
- run PT-SCOTCH on the graph and get a separator's tree
- leaf subtrees are processed independently by processors
- the top of the tree is processed sequentially by a "root" node



Coupling the two steps

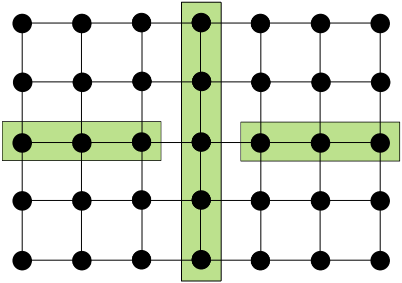
The ordering and the symbolic factorization are performed on the graph built from $|A| + |A^T|$

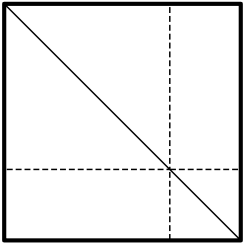
- run PT-SCOTCH on the graph and get a separator's tree
- leaf subtrees are processed independently by processors
- the top of the tree is processed sequentially by a "root" node



Coupling the two steps

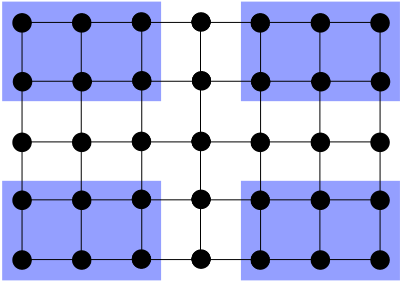
The symbolic factorization of the top of the tree is modeled as a restarting step

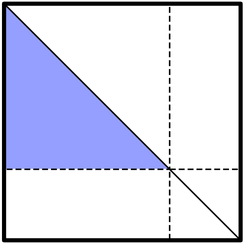




Coupling the two steps

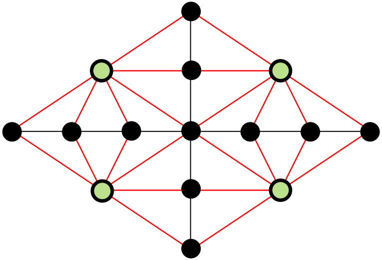
The symbolic factorization of the top of the tree is modeled as a restarting step

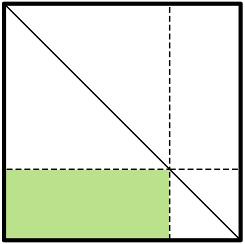




Coupling the two steps

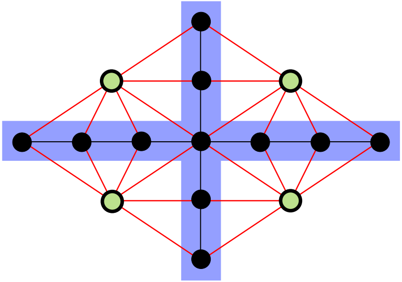
The symbolic factorization of the top of the tree is modeled as a restarting step

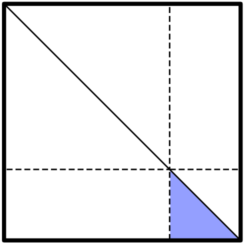




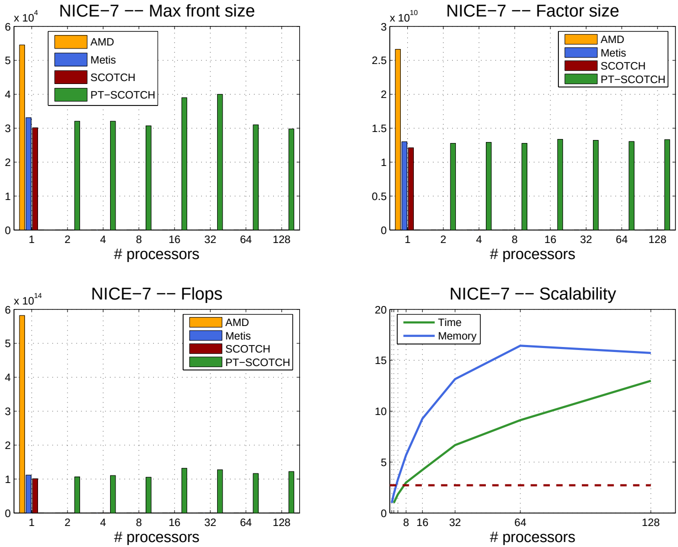
Coupling the two steps

The symbolic factorization of the top of the tree is modeled as a restarting step

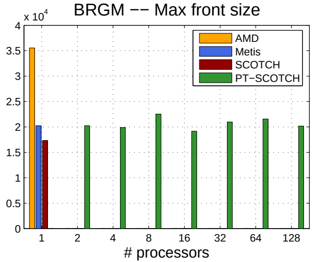


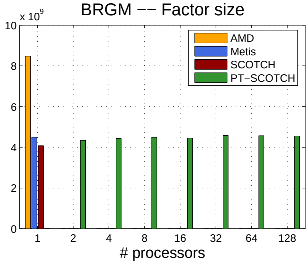


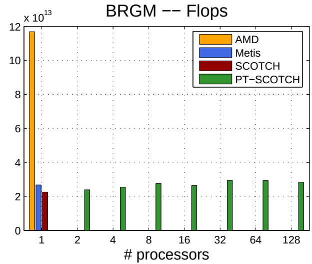
NICE-7: N=8159758, NNZ=669172552

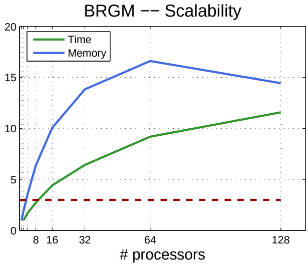


BRGM: N=3699643, NNZ=307580395

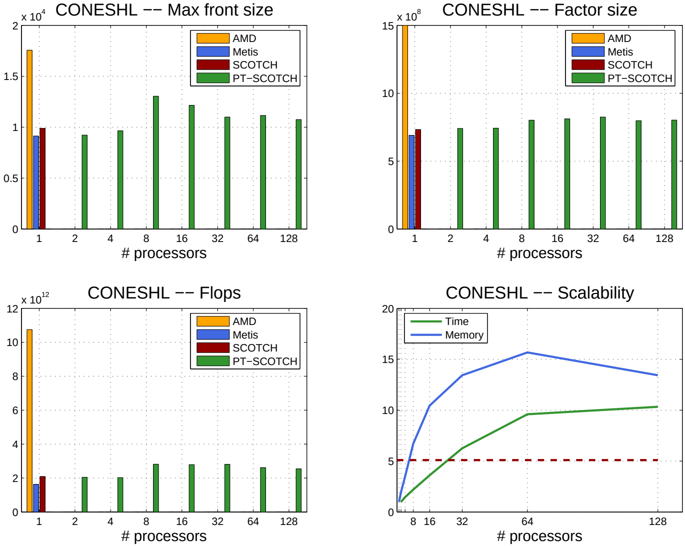




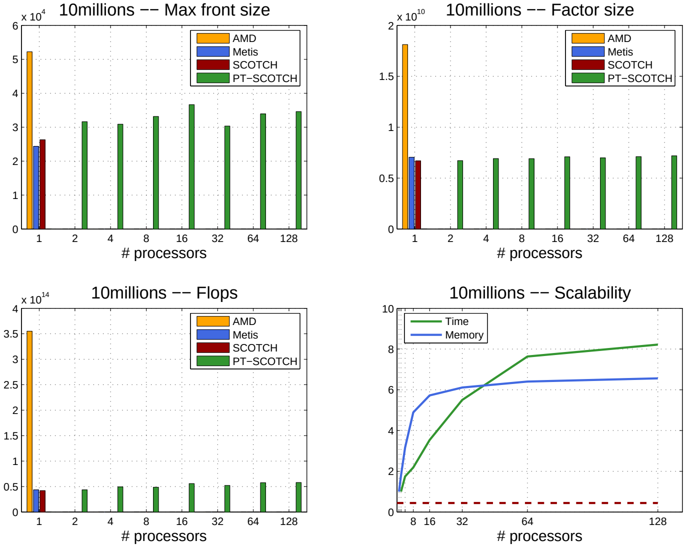




CONESHL: N=1262212, NNZ=84753352



10millions: N=10423737, NNZ=167722005



Future work

- Parallelize top-of-the-tree symbolic factorization
- Experiment with *multisector ordering* schemes [Ashcraft and Liu, 1998]
- Parallelize *amalgamation*
- Parallelize *scaling* [Amestoy et al., 2008]
- Parallelize *maximum transversal*

Thanks

[Amestoy, 1997] Amestoy, P. R. (1997).
Recent progress in parallel multifrontal solvers for unsymmetric sparse matrices.
In *Proceedings of the 15th World Congress on Scientific Computation, Modelling and Applied Mathematics, IMACS 97, Berlin.*

[Amestoy et al., 1996] Amestoy, P. R., Davis, T. A., and Duff, I. S. (1996).
An approximate minimum degree ordering algorithm.
17:886–905.

[Amestoy et al., 2008] Amestoy, P. R., Duff, I. S., Ruiz, D., and car, B. U. (2008).
A parallel scaling algorithm.
In *VECPAR 08*, Lecture Notes in Computer Science. Springer.
To appear.

[Ashcraft and Liu, 1998] Ashcraft, C. and Liu, J. W. H. (1998).
Robust ordering of sparse matrices using multisection.
SIAM J. Matrix Anal. Appl., 19(3):816–832.

[Chevalier and Pellegrini, 2006] Chevalier, C. and Pellegrini, F. (2006).
PT-Scotch: A tool for efficient parallel graph ordering.
In *Proceedings of PMAA2006, Rennes, France.*

[George and Liu, 1980] George, A. and Liu, J. W. H. (1980).
An optimal agorithm for symbolic factorization of symmetric matrices.
SIAM Journal on Computing, 9(3):583–593.

[Grigori et al., 2007] Grigori, L., Demmel, J. W., and Li, X. S. (2007).
Parallel symbolic factorization for sparse lu with static pivoting.
SIAM J. Sci. Comput., 29(3):1289–1314.