January 01, 2006

# Characterizing the relationship between sparse matrix preconditioners and the storage hierarchy

Diego Rivera
*Northeastern University*

David Kaeli
*Northeastern University*

# "Characterizing the Relationship between Sparse Matrix Preconditioners and the Storage Hierarchy"


Diego Rivera and David Kaeli
The Center for Subsurface Sensing and Imaging Systems (CenSSIS)
Dept. of Electrical and Computer Engineering
Northeastern University, Boston, MA 02115
{drivera, kaeli}@ece.neu.edu, www.ece.neu.edu/students/drivera/tlg/tunlib.htm


### Objective

- To improve the performance of preconditioning techniques codes
- Accelerate the memory accesses associated with these codes

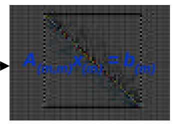
### Motivation

- Prior work targeted Krylov subspace methods
- However, little has been done in the case of preconditioners

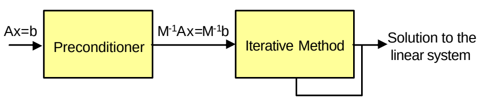
*"Nothing will be more central to computational science in the next century than the art of transforming a problem that appears intractable into another whose solution can be approximated rapidly. For Krylov subspace matrix iterations, this is preconditioning"* from book Numerical Linear Algebra by Trefethen and Bau (1997).

### Common target applications

- *Diffuse Optimal Tomography (CenSSIS)*
- *Chemical engineering applications*
- *Turbulence problems in airplanes*
- *DNA models*
- *Circuit simulation*




- Computational time is a barrier in these applications
- Parallel processing can be used to lower this barrier
- The sparsity of the data reduces the effectiveness of direct parallel computation
- Preconditioning techniques can be used to accelerate the convergence of Krylov subspace methods

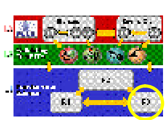



- A drawback of these approaches is that it's difficult to choose good values for their tuning-parameters
- Choosing good values depends heavily on the structure of non-zero elements of the coefficient matrix.
- In our work we have found that it depends also on the memory hierarchy machine used to compute the solution

### State of the art

- Development of new preconditioner techniques (2003-06)
- Introduction of suitable parallel computing tools for Krylov methods
- Also, several heuristic solutions for mapping sparse matrices (2002-04) ® SpMxV
- Improvement of Kernel optimization techniques (2004) ® SpMxV, $AA^T$ and $A^k$
- Introduction of parallel versions of ILU, approximate inverse and AMG(2001-06).
- Development of several algorithms based on data mining techniques for predicting the solution of a system for a certain precondioner (2004-05)
- What about tuning memory access patterns of preconditioner techniques?


### Acknowledgement



This work is affiliated with CenSSIS, the Center for Subsurface Sensing and Imaging Systems. Support is provided by National Science Foundation under Grant No. NFS ACR:0342555. This project is also supported by the National Science Foundation's Computing and Communication Foundations Division, grant number CCF-0342555 and the Institute of Complex Scientific Software.


### Experiments and results

- Incomplete LU factorization preconditioner with threshold (ILUT)
- Similar behavior has been observed for other variations of ILU preconditioners
- The behavior depends on the dropping strategies adopted

ILUT(ρ,τ) algorithm

```
for i = 1,…, n
  w = a_i,all
  for k = 1,…, i - 1
    if w_k ? 0
      w_k = w_k / a_k,k
      Apply 1st dropping rule to w_k
      if w_k ? 0
        w = w - w_k*u_k,all
      endif
    endif
  endfor
  Apply 2nd dropping rule to row w
  l_i,j = w_j for j = 1,…, i - 1
  u_i,j = w_j for j = l,…, n
  w = 0
endfor
```

- *This part is the main source of conflict misses*
- *The accuracy of the preconditioner and conflict misses are directly related*

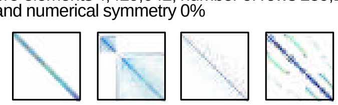
- τ minimizes the number of operations required (reduces computation cost)
- *r* controls the number of elements per row (controls memory usage)

### Evaluation environment

- Intel XEON 3.06 GHz, L1 8KB 4-way for data, 12KB 4-way for instructions, L2 512 KB 8-way, L3 1 MB 8-way, 2 GB RAM. All cache levels use a pseudo-LRU replacement algorithm
- Ultra Sparc-III 750 MHz, L1 64KB 4-way for data, 32 KB 4-way for instructions, L2 8MB 2-way, 1 GB RAM . All the cache levels use a pseudo-random replacement algorithm
- Pin - A dynamic binary instrumentation tool used to capture cache events. LRU and random replacement policies are modeled

### Matrices

- Several matrices were evaluated. Results from four representative matrices are shown
- Raefsky3: non-zero elements 1,488,768, number of rows 21,200, structural symmetry 100% and numerical symmetry 48%
- Ldoor: non-zero elements 42,493,817, number of rows 952,203, structural symmetry 100% and numerical symmetry 100%
- Cage14: non-zero elements 27,130,349, number of rows 1,505,785, structural symmetry 100% and numerical symmetry 21%
- Torso3: non-zero elements 4,429,042, number of rows 259,156, structural symmetry 95% and numerical symmetry 0%

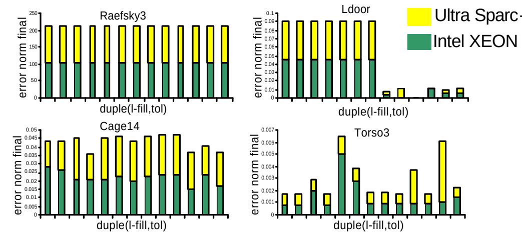



*Relation numerical-symmetry/matrix-bandwidth decreases in this direction*

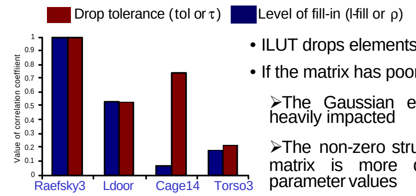
### Technology transfer

This research addresses an important problem in accelerating the execution of applications that involve the solution of large sparse linear systems, including applications in the area of Subsurface Sensing and Imaging System. This has a high value for the industrial and CenSSIS community.

*Error norm vs. 13 first duple sorted in increasing order for execution time of preconditioner and Krylov method*



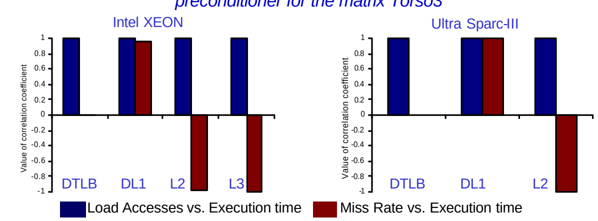

*Correlation of tuning-parameters between both machines*




- ILUT drops elements by their magnitude
- If the matrix has poor numerical symmetry:
  - The Gaussian elimination process is more heavily impacted
  - The non-zero structure of the new coefficient matrix is more dependent on the tuning-parameter values

- This dependence on the tuning-parameter values is translated to the our ability to exploit the locality present in the new matrix, given a particular memory hierarchy

*Correlation of memory events (load accesses and miss rate) vs. Execution time of preconditioner for the matrix Torso3*




- The same trend was observed in other matrices
- The execution time is highly correlated to load accesses: the amount of instructions for the inner statement "if" and the accuracy of the preconditioner are directly related
- In levels 2 and 3, the miss rate and the execution time have a strong inverse correlation: reducing cache misses will speed up the construction phase of a preconditioner whose accuracy is acceptable
- The overall execution time may be reduced, which occurs in memory hierarchies that have the ability to exploit the locality in the new matrix

*We are working to define an algorithm for selecting the best preconditioner for a given memory hierarchy keeping, while considering the convergence rate and final error. Also we are exploring compiler optimization techniques to reduce conflict misses.*

The general steps of that algorithm are:

1. Determine the relationship between the preconditioner's parameters, the system's memory hierarchy and the problem.
2. Find a suitable range for the preconditioner parameter values using a semi-automatic procedure.
3. Construct the preconditioner.