The MUMPS library: work done during the SOLSTICE project

MUMPS team, Lyon-Grenoble, Toulouse, Bordeaux

Sparse Days and ANR SOLSTICE
Final Workshop
June 2010
Permanent members in 2007

Patrick Amestoy (N7-IRIT, Toulouse)

Jean-Yves L’Excellent (INRIA-LIP, Lyon)

Abdou Guermouche (LABRI, Bordeaux)

Bora Uçar (CNRS-LIP, Lyon)

Alfredo Buttari (CNRS-IRIT, Toulouse)
Permanent members in 2010

Patrick Amestoy (N7-IRIT, Toulouse)

Jean-Yves L’Excellent (INRIA-LIP, Lyon)

Abdou Guermouche (LABRI, Bordeaux)

Bora Uçar (CNRS-LIP, Lyon)

Alfredo Buttari (CNRS-IRIT, Toulouse)
MUMPS Team since beg. of SOLSTICE (2007)

- Post-docs funded by the ANR SOLSTICE project:
  - Indranil Chowdhury (May 2009–March 2010)

- PhD. Students Emmanuel Agullo (ENS Lyon, 2005-2008)
  - Mila Slavova (CERFACS, Toulouse, 2005-2009)
  - François-Henry Rouet (INPT-IRIT, Toulouse, 2009-)
  - new!

- Master Student Clément Weisbecker (INPT-IRIT, Toulouse)
  - new!

- Engineers Aurélie Fève (INRIA, 2005-2007)
  - new!
  - new!
What is MUMPS

MUMPS (MUltifrontal Massively Parallel sparse direct Solver) solves sparse systems of linear equations $Ax = b$


- Initially funded by European project PARASOL (1996-1999)
- Co-developed in Lyon-Toulouse-Bordeaux
  $\approx$ 250 000 lines of C and Fortran code
- 1000+ downloads per year from our website

Main originalities
- Wide range of numerical features
- Numerical stability based on threshold partial pivoting
- Mainly MPI-based, with a dynamic and asynchronous approach to parallelism
News since beg. of SOLSTICE (2007)

Download requests forms filled on the MUMPS website

Number of requests

Month


Colors:
- 4.9
- 4.8
- 4.7
- 4.6
SOLSTICE project http://solstice.gforge.inria.fr

- Supported by ANR (project number ANR-06-CIS6-010)
- Academics: CERFACS, INRIA (Lyon and Bordeaux), INPT-IRIT
  Industrials: CEA-CESTA, EADS CCR, EDF, CNRS-CNRM-LA

SOLSTICE tasks

- **T1**: Algorithmic tasks
  - T1.1: Rank-revealing and null space
  - T1.2: Out-of-core
  - T1.3: Parallel graph partitioners – PT-SCOTCH
  - T1.4: Parallel analysis and preprocessings
  - T1.5: Hybrid direct/iterative solvers

- **T2**: Applications related to industrial activities (Meteo, Electromagnetism, Structural mechanics)

- **T3**: TLSE platform (http://www.gridtlse.org/)
SOLSTICE project http://solstice.gforge.inria.fr

- Supported by ANR (project number ANR-06-CIS6-010)
- Academics: CERFACS, INRIA (Lyon and Bordeaux), INPT-IRIT
  Industrials: CEA-CESTA, EADS CCR, EDF, CNRS-CNRM-LA

SOLSTICE tasks involving MUMPS team

- T1: Algorithmic tasks
  - T1.1: Rank-revealing and null space $\rightarrow$ collab. with X. Vasseur (CERFACS) and S. Gratton (INPT-IRIT)
  - T1.2: Out-of-core
  - T1.3: Parallel graph partitioners – PT-SCOTCH
  - T1.4: Parallel analysis and preprocessings $\rightarrow$ see talk by Alfredo Buttari and Bora Uçar
  - T1.5: Hybrid direct/iterative solvers

- T2: Applications related to industrial activities (Meteo, Electromagnetism, Structural mechanics)

- T3: TLSE platform (http://www.gridtlse.org/) $\rightarrow$ see talk by TLSE team
CONTENTS OF THE TALK

• Illustration of out-of-core work (T1.2)

• Illustration of additional work (concerning all tasks including industrial requirements)

• Multicores: some recent work to mix OpenMP and MPI

• Concluding remarks and future work
Out-of-core related work

Out-of-core storage: 2 PhD completed

- Emmanuel AGULLO (ENS Lyon, 2005-2008) *On the Out-of-core Factorization of Large Sparse Matrices*
- Mila Slavova (CERFACS, Toulouse, 2005-2009) *Parallel triangular solution in the out-of-core multifrontal approach*
- Tree traversals, low-level I/O mechanisms, task scheduling, . . .

- Out-of-core factorization using a panel-oriented scheme
  (Needed to validate research; numerical pivoting \(\rightarrow\) more complex)

<table>
<thead>
<tr>
<th>Matrix</th>
<th>#procs</th>
<th>I/O granularity for Factors</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Written by fronts</td>
</tr>
<tr>
<td>AUDIKW_1</td>
<td>1</td>
<td>1067.1</td>
</tr>
<tr>
<td>AUDIKW_1</td>
<td>32</td>
<td>155.5</td>
</tr>
<tr>
<td>CONV3D64</td>
<td>1</td>
<td>3341.5</td>
</tr>
<tr>
<td>CONV3D64</td>
<td>32</td>
<td>757.6</td>
</tr>
</tbody>
</table>

Size of I/O Buffers (MB) with asynchronous I/O’s
Out-of-core related work (cont’)

Computing entries of the inverse of a sparse matrix

- **PhD François-Henry Rouet**, 2009-
  (continuation/extension of PhD work of M. Slavova)
  - **Exploit sparsity** of multiple right-hand sides (RHS) during the solution
  - Combinatorial problem to decide **how to partition the right-hand-sides**.
  - Illustration on an application from astrophysics

<table>
<thead>
<tr>
<th>Exploit sparsity of RHS</th>
<th>43 396 sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>Natural ordering</td>
<td>721 sec</td>
</tr>
<tr>
<td>PostOrdering</td>
<td>199 sec</td>
</tr>
</tbody>
</table>

time to compute **ALL** diagonal of $A^{-1}$, OOC, $N=148k$

- **On going work** : use of hypergraph partitionning, in-core and flop reduction, parallelism.

<table>
<thead>
<tr>
<th>Exploit sparsity of RHS</th>
<th>4708 sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exploit sparsity</td>
<td>188 sec</td>
</tr>
</tbody>
</table>

time to compute **ALL** the diagonal of $A^{-1}$, in core, $N=148k$
• 64-bit integers to address large internal arrays (requested by users but also needed to experiment out-of-core and parallel analysis on large challenging problems)

• Redesign parts of the mapping algorithm ("Epicure" matrix from EDF)

<table>
<thead>
<tr>
<th>Factors Min/Max</th>
<th>InCore Memory (in MB) Avg</th>
<th>InCore Memory (in MB) Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Before</td>
<td>0.06</td>
<td>1,753</td>
</tr>
<tr>
<td>After</td>
<td>0.70</td>
<td>1,634</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Nprocs MPI</th>
<th>Factor. time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>337</td>
</tr>
<tr>
<td>4</td>
<td>316</td>
</tr>
<tr>
<td>8</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td></td>
</tr>
</tbody>
</table>
Work started with Indranil Chowdhury, Sosltice postdoc (May 2009–March 2010)

Main goal: understand the limits of MPI + threaded BLAS + OpenMP

Work based on the use of TAU and Intel Vtune profiles:
• Insert OpenMP directives (assembly, stack, initialization,...)
• Experiment with various matrices on various architectures

Difficulties:
• Understand speed-down problems (use of OMP IF directives)
• Side effects of threaded BLAS libraries (MKL ok)
• Dependence on OpenMP implementations
• Threaded BLAS within OpenMP parallel regions
Initial results:

- best current compromise: use 4 threads per MPI process
typical speed-up of 6 on 8 cores
- unsymmetric and Cholesky versions more efficient than $LDL^t$

Recent illustrative results:

- 96-core machine, INRIA Bordeaux Sud-Ouest
- Runs by A. Guermouche on a medium-size problem (ULTRASOUND80)
  - Order: 531 441, non-zeroes: $330 \times 10^6$
  - Factors: $981 \times 10^9$ entries, $3.9 \times 10^{12}$ flops
Execution time (in seconds)

- MPI only
- MPI+Thread MKL (4 threads per process)
- MPI+Thread MKL + OMP (4 threads per process)
Increasing the number of threads per MPI process is also critical for memory:

1 MPI 8 threads → 8.97 GB
2 MPI 4 threads → 11.0 GB
4 MPI 2 threads → 11.3 GB
8 MPI 1 thread → 13.3 GB
Locality issues with $LDL^t$ factorization (row-major stor.)

- Fully summed variables
- L factor (in pivot block)
- L factor (in pivot block) (BLAS 3 update of fully summed columns)
- 1. DCOPY
- 2. DSYR
- 3. DSCAL
- 4. DGER

Pivot block

: Pivot
Rewriting factorization of pivot block

Old algorithm (threaded BLAS)

1. DCOPY (column to row in upper part)
2. DSCAL (column)
3. DSYR (triangular block)
4. DGER (rectangular block)

New algorithm

Hand-written
No BLAS calls
OpenMP based
Locality better exploited

Results on matrix “Haltere” from CEA-CESTA, CINES SGI

<table>
<thead>
<tr>
<th># threads</th>
<th>1</th>
<th>2</th>
<th>4</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Before (4.9.2)</td>
<td>16.7 (115)</td>
<td>16.7 (75)</td>
<td>19.9 (64)</td>
<td>29.0 (66)</td>
</tr>
<tr>
<td>Intermediate</td>
<td>15.8 (110.7)</td>
<td>10.1 (65.4)</td>
<td>8.5 (47.5)</td>
<td>7.6 (39.6)</td>
</tr>
<tr>
<td>After</td>
<td>15.7 (110.4)</td>
<td>9.8 (64.6)</td>
<td>6.0 (44.1)</td>
<td>4.7 (34.9)</td>
</tr>
</tbody>
</table>

Legend: Time factorization of pivot block (time for whole factorization)

Times in seconds Intermediate=without “OMP IF”
Rewriting factorization of pivot block

Old algorithm (threaded BLAS)

1. DCOPY (column to row in upper part)
2. DSCAL (column)
3. DSYR (triangular block)
4. DGER (rectangular block)

New algorithm

- Hand-written
- No BLAS calls
- OpenMP based
- Locality better exploited

Results on matrix “Haltere” from CEA-CESTA, CINES SGI

<table>
<thead>
<tr>
<th># threads</th>
<th>1</th>
<th>2</th>
<th>4</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Before (4.9.2)</td>
<td>16.7 (115)</td>
<td>16.7 (75)</td>
<td>19.9 (64)</td>
<td>29.0 (66)</td>
</tr>
<tr>
<td>Intermediate</td>
<td>15.8 (110.7)</td>
<td>10.1 (65.4)</td>
<td>8.5 (47.5)</td>
<td>7.6 (39.6)</td>
</tr>
<tr>
<td>After</td>
<td>15.7 (110.4)</td>
<td>9.8 (64.6)</td>
<td>6.0 (44.1)</td>
<td>4.7 (34.9)</td>
</tr>
</tbody>
</table>

Legend: Time factorization of pivot block (time for whole factorization)

Times in seconds Intermediate=without “OMP IF”
Further improvements to $LDL^t$ version

During update of pivot block with new algorithm, check for stability of pivot candidate from next column (while in cache!)

Pivot block

L factor (in pivot block)

Fully summed variables

1. DCOPY
2. DSYR
3. DSCAL
4. DGER

MUMPS team, Sparse Days and ANR SOLSTICE, Final Workshop, June 2010
Further improvements to $LDL^t$ version

During update of pivot block with new algorithm, check for stability of pivot candidate from next column (while in cache!)

<table>
<thead>
<tr>
<th># threads</th>
<th>algorithm</th>
<th>Pivot search</th>
<th>Pivot block update</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>old</td>
<td>2.77</td>
<td>15.6</td>
</tr>
<tr>
<td></td>
<td>new</td>
<td>0.91</td>
<td>15.3</td>
</tr>
<tr>
<td>2</td>
<td>old</td>
<td>1.95</td>
<td>9.8</td>
</tr>
<tr>
<td></td>
<td>new</td>
<td>0.72</td>
<td>9.7</td>
</tr>
<tr>
<td>4</td>
<td>old</td>
<td>1.56</td>
<td>6.0</td>
</tr>
<tr>
<td></td>
<td>new</td>
<td>0.64</td>
<td>6.0</td>
</tr>
<tr>
<td>8</td>
<td>old</td>
<td>1.58</td>
<td>4.7</td>
</tr>
<tr>
<td></td>
<td>new</td>
<td>0.64</td>
<td>4.8</td>
</tr>
</tbody>
</table>

Time reduction on pivot search without affecting pivot block update
\( LDL^t \) : Summary

- Huge gains obtained by studying more closely locality aspects:
  On 8 OpenMP threads: 66 → 34 seconds (1 thread: 111 seconds)
  (4 threads with 2 MPI processes would be better – see previous results)

**On-going work**
- Similar work needed for 2x2 pivots updates and distributed fronts
- Other kernels to be studied in more details
- For example on this matrix: 7 seconds out of 34 (8 threads) and the same 7 seconds out of 111 (1 thread) spent in assembly operations
- Clearly much scope for further improvements

18/20 MUMPS team, Sparse Days and ANR SOLSTICE, Final Workshop, June 2010
Main new functionalities since beginning of SOLSTICE project

- T1.1: null pivots and null space basis
- T1.2: out-of-core factorization and solve phases
- T1.4: parallel scalings and parallel symbolic factorization, use of PT-scotch and Parmetis
- 64-bit integers to address large arrays
- Performance and memory improvements on SOLSTICE applications
- Determinant (factor-less factorization)
Future work

- Improve multithreaded parallelism:
  - locality, affinity: continue on-going work
  - threaded tree parallelism at bottom of tree
  - serial BLAS and more OpenMP? GPU?

- Improve MPI parallelism, memory scalability and quality of memory estimates in a parallel context (PhD thesis of F.H. Rouet)

- Performance of solution phase (MPI/OpenMP/sparse right-hand sides)

- Reduce memory consumption and improve performance