Intra-Block Amalgamation in Sparse Hypermatrix Cholesky Factorization

José R. Herrero, Juan J. Navarro

{josepr,juanjo}@ac.upc.edu

Computer Architecture Department
Universitat Politècnica de Catalunya
Barcelona
Spain
Outline

- Introduction
  - Hypermatrices
  - Small Matrix Library (SML)
  - Windows within data submatrices
- Intra-Block Amalgamation
- Results
- Conclusions
Introduction

Cholesky factorization

- Symmetric Positive Definite (SPD) matrix

Sparse matrices

- Plenty of 0’s
- Avoid storage and computation on 0’s
A sparse matrix and an equivalent Hypermatrix
Overhead

Can store 0’s within data submatrices

- Storage
- Computation

Trade-off in data submatrix size

- BLAS3 efficiency
- (Useless) operation on 0’s
Efficient Implementation of a sparse Cholesky factorization

...using a hypermatrix

Important fact:

Matrix multiplicacion takes around 90% of the total factorization time.
Reducing Overhead & Increasing Performance

Summary of (effective) previous work:

- Efficient kernels which operate on small data submatrices (SML)
- Windows within data submatrices
Efficient operation on small data submatrices

Goal:

- Reduce data submatrix size while keeping good BLAS3 performance

Idea [Euro-Par’03]:

- Fix parameters at compilation time
- Choose best algorithm
  - Loop unrolling factors
  - Loop orders
Matrix multiplication performance on small matrices

\[ C = C - A \cdot B^t \]

R10000 250 MHz (500 Mflops peak)
Hypermatrix Cholesky on problem PDS40

HM Cholesky: pds40

LP problem: Patient Distribution System (40 days)
Window within a data submatrix

Data Submatrix

left column

right column

top row

bottom row

window

See [HPCSE’04, PARA’04]
Using windows to avoid (some) useless operations
HM performance with and without windows
Matrix multiplication using windows in 2 dimensions
Matrix multiplication using windows in rows
Matrix multiplication using windows in columns
Matrix multiplication of full data submatrices
Matrix multiplication: efficiency of codes

Less efficient

Most efficient
Intra-Block Amalgamation: Original window
Intra-Block Amalgamation: column-wise

Data Submatrix

top row

bottom row

left column

right column

window

Data Submatrix
Intra-Block Amalgamation: row-wise

Data Submatrix

left column

right column

top row

bottom row

window

x

x

x

x
Intra-Block Amalgamation: row and column-wise

Data Submatrix

left column

right column

top row

bottom row

window
Results: Context information

- MIPS R10000 @ 250 MHz (500 Mflops peak)
- Sequential code
- Data submatrices of fixed size
- Large problems solved In-Core
- Ordered using METIS
- Linear Programming problems
  - NetLib
  - Multicommodity Network Flow generators
### Results: Matrix Characteristics

<table>
<thead>
<tr>
<th>Matrix</th>
<th>Dimension</th>
<th>NZs</th>
<th>NZs in $L^a$</th>
<th>Density</th>
<th>Flops to factor $^b$</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRIDGEN1</td>
<td>330430</td>
<td>316257</td>
<td>130586943</td>
<td>0.002</td>
<td>278891</td>
</tr>
<tr>
<td>QAP8</td>
<td>912</td>
<td>14864</td>
<td>193228</td>
<td>0.463</td>
<td>63</td>
</tr>
<tr>
<td>QAP12</td>
<td>3192</td>
<td>77784</td>
<td>2091706</td>
<td>0.410</td>
<td>2228</td>
</tr>
<tr>
<td>QAP15</td>
<td>6330</td>
<td>192405</td>
<td>8755465</td>
<td>0.436</td>
<td>20454</td>
</tr>
<tr>
<td>RMFGEN1</td>
<td>28077</td>
<td>151557</td>
<td>6469394</td>
<td>0.016</td>
<td>6323</td>
</tr>
<tr>
<td>TRIPART1</td>
<td>4238</td>
<td>80846</td>
<td>1147857</td>
<td>0.127</td>
<td>511</td>
</tr>
<tr>
<td>TRIPART2</td>
<td>19781</td>
<td>400229</td>
<td>5917820</td>
<td>0.030</td>
<td>2926</td>
</tr>
<tr>
<td>TRIPART3</td>
<td>38881</td>
<td>973881</td>
<td>17806642</td>
<td>0.023</td>
<td>14058</td>
</tr>
<tr>
<td>TRIPART4</td>
<td>56869</td>
<td>2407504</td>
<td>76805463</td>
<td>0.047</td>
<td>187168</td>
</tr>
<tr>
<td>pds1</td>
<td>1561</td>
<td>12165</td>
<td>37339</td>
<td>0.030</td>
<td>1</td>
</tr>
<tr>
<td>pds10</td>
<td>18612</td>
<td>148038</td>
<td>3384640</td>
<td>0.019</td>
<td>2519</td>
</tr>
<tr>
<td>pds20</td>
<td>38726</td>
<td>319041</td>
<td>10739539</td>
<td>0.014</td>
<td>13128</td>
</tr>
<tr>
<td>pds30</td>
<td>57193</td>
<td>463732</td>
<td>18216426</td>
<td>0.011</td>
<td>26262</td>
</tr>
<tr>
<td>pds40</td>
<td>76771</td>
<td>629851</td>
<td>27672127</td>
<td>0.009</td>
<td>43807</td>
</tr>
<tr>
<td>pds50</td>
<td>95936</td>
<td>791087</td>
<td>36321636</td>
<td>0.007</td>
<td>61180</td>
</tr>
<tr>
<td>pds60</td>
<td>115312</td>
<td>956906</td>
<td>46377926</td>
<td>0.006</td>
<td>81447</td>
</tr>
<tr>
<td>pds70</td>
<td>133326</td>
<td>1100254</td>
<td>54795729</td>
<td>0.006</td>
<td>100023</td>
</tr>
<tr>
<td>pds80</td>
<td>149558</td>
<td>1216223</td>
<td>64148298</td>
<td>0.005</td>
<td>125002</td>
</tr>
<tr>
<td>pds90</td>
<td>164944</td>
<td>1320298</td>
<td>70140993</td>
<td>0.005</td>
<td>138765</td>
</tr>
</tbody>
</table>

- $^a$ Number of non-zeros in factor $L$ (matrix ordered using METIS).
- $^b$ Number of floating point operations (in Millions) necessary to obtain $L$ from the original matrix (ordered with METIS).
Results: QAP12

Intra-Block Amalgamation in Sparse Hypermatrix Cholesky Factorization

Jose R. Herrero, Comp. Arch. Dept., UPC
Results: TRIPART1

Intra-Block Amalgamation in Sparse Hypermatrix Cholesky Factorization

Jose R. Herrero, Comp. Arch. Dept., UPC
Results: TRIPART2

Intra-Block Amalgamation in Sparse Hypermatrix Cholesky Factorization

Jose R. Herrero, Comp. Arch. Dept., UPC
Results: pds10

![Graph showing Effective Mflops vs Columns amalgamated]

Legend:
- amr=0
- amr=1
- amr=2
- amr=3
Results: pds20

![Graph showing effective Mflops vs columns amalgamated. The graph has different lines for amr=0, amr=1, amr=2, and amr=3, with the x-axis representing columns amalgamated and the y-axis representing effective Mflops.](image-url)
Results: Original (without amalgamation) vs Intra-block amalgamation

ICCSE’05
Intra-Block Amalgamation in Sparse Hypermatrix Cholesky Factorization
Istanbul, Turkey, June 27, 2005
José R. Herrero, Comp. Arch. Dept., UPC
Results: Performance of several sparse Cholesky codes
Conclusions and future work

- R10000: row amalgamation 1 + column amalgamation 5
  - 5.3% Average improvement on matrix test suite
- Current work (Paper in progress)
  - Use variable partitioning of hypermatrix using the Elimination Tree
  - Hypermatrix oriented supernode amalgamation
- Future work
  - Store data submatrices as supernodes
Overhead in number of operations in sparse HM Cholesky (4x32 + windows).