Improving Performance of Hypermatrix Cholesky Factorization

José R. Herrero, Juan J. Navarro

{josepr,juanjo}@ac.upc.es

Dept. Arquitectura Computadors

Universitat Politècnica de Catalunya

Barcelona, Spain

Talk Outline

- Introduction
- Goal
- Building an efficient Small Matrix Library
- Impact on Sparse Hypermatrix Cholesky
- Conclusions

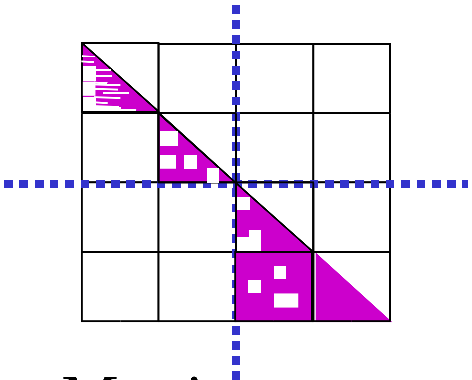
Introduction

- Cholesky factorization
  - Symmetric Positive Definite (SPD) matrix
  - $A = L * L^T$

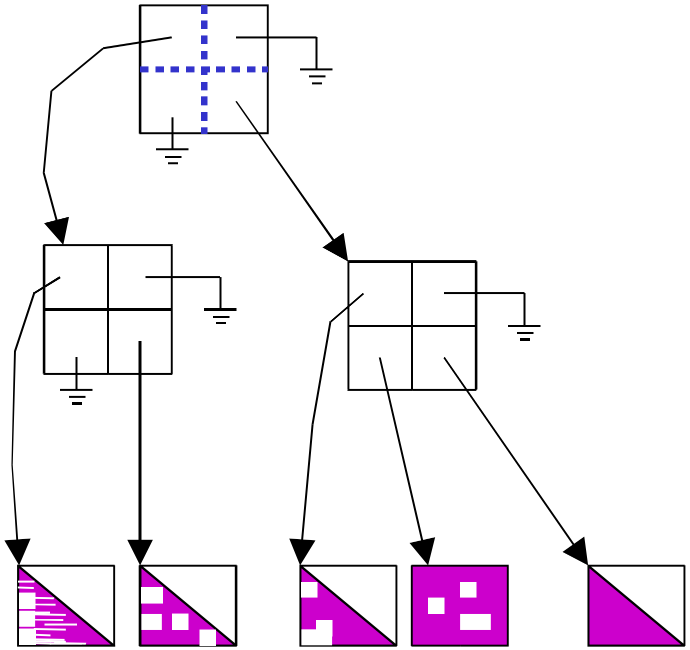
- Sparse matrices
  - Plenty of 0's
    - Avoid storage and computation on 0's

Introduction

- Cholesky Factorization using Hypermatrices



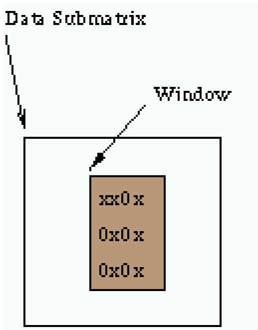
Matrix



HyperMatrix

Introduction

- Potential advantage
  - 2D matrix partition
- Disadvantage
  - Large overhead due to 0's within data submatrices



- Trade - off
  - Block size

Talk Outline

- Introduction
- Goal
- Building an efficient Small Matrix Library
- Impact on Sparse Hypermatrix Cholesky
- Conclusions

Goal

- Improve the sparse hypermatrix Cholesky factorization:

- Get routines which
  - Work on small matrices (which fit in cache)
  - Are efficient
  - Can be created for any target system

  $\Rightarrow$ Create a Small Matrix Library (SML)

Idea

- Fix parameters at compilation time
  - Leading dimensions
  - Loop limits

- Example: $C=C+A*B^T$

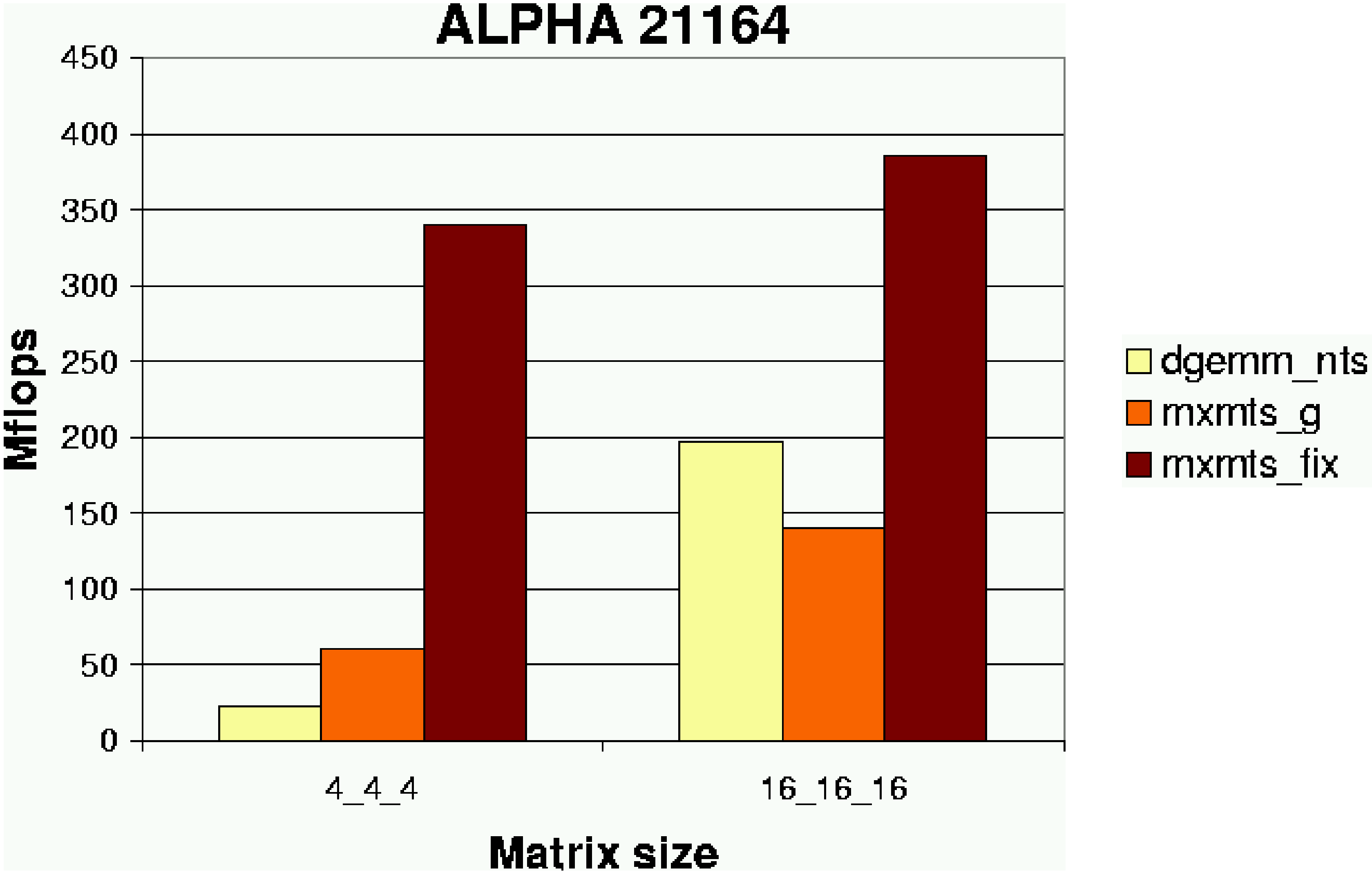
```
subroutine mxmt(A,B,C,
   lda,ldb,ldc, ui, uj, uk)
integer A(lda,*), ...
do I=1, ui
    ...
```

→

```
subroutine mxmt_fix(A,B,C)


integer A(8,*)
do I=1, 8
    ...
```

Motivation: mxmt_fix results



Talk Outline

- Introduction
- Goal
- Building an efficient Small Matrix Library
- Impact on Sparse Hypermatrix Cholesky
- Conclusions

Procedure

- Write several variants of code
  - Loop order
  - Loop unrolling
- Use the *best* compiler available
  - Try several compilation options

Example: MxM^t codes

Code of Algorithm	form
1	jik_Creg
2	jik
3	kji_Breg
4	$i(jk4\tilde{i}\_BCreg)$
5	$i(jk4\tilde{i}\_Breg)$
6	$i(jk4\tilde{i})$
7	ijk
8	kji
9	jki
10	$jik4\tilde{k}\_Creg$
11	$jik8\tilde{k}\_Creg$

Example: MxM^t optimum

Code of Algorithm	form
1	*jik_Creg*
2	*jik*
3	*kji_Breg*
4	*i(jk4ĩ_BCreg)*
5	*i(jk4ĩ_Breg)*
6	*i(jk4ĩ)*
7	*ijk*
8	*kji*
9	*jki*
10	*jik4k̃_Creg*
11	*jik8k̃_Creg*

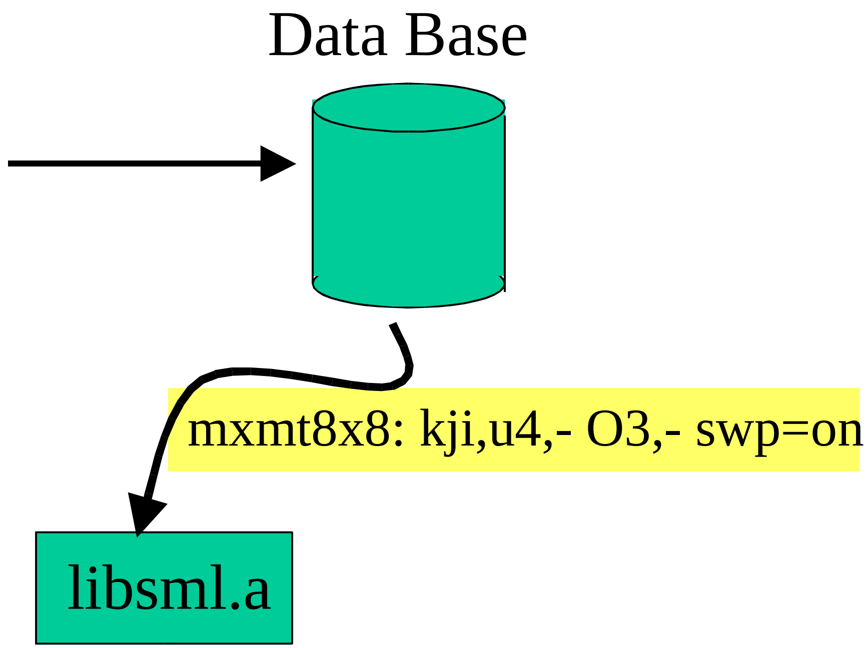
Matrix sizes	*Alpha 21164*	*R10000*
4_4_4	2	8
4_4_8	3	5
4_4_16	3	3
4_4_32	3	3
8_8_8	10	6
8_8_16	11	8
8_8_32	11	5
16_16_16	11	2
16_16_32	11	5
32_32_32	11	6

Procedure

- Many combinations:
  - Leading dimensions
  - Loop limits
  - Loop orders
  - Loop unrolling factors
  - Compiler flags
  - Target machines
- We need to automate the tests!
  - Use a benchmarking tool

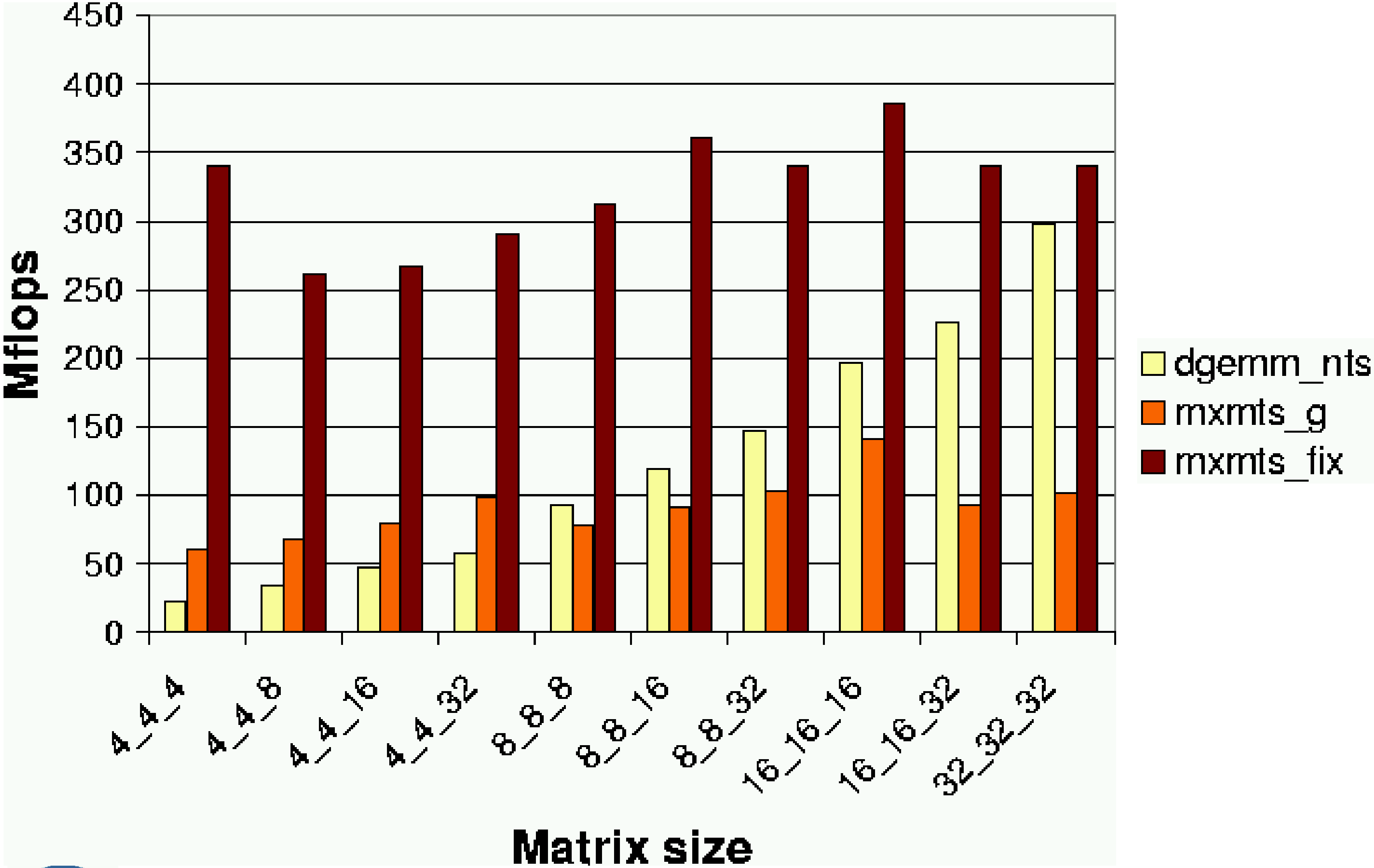
Benchmarking Tool

- foreach parameter combination
  - compile
  - execute
  - store results (Mflops)

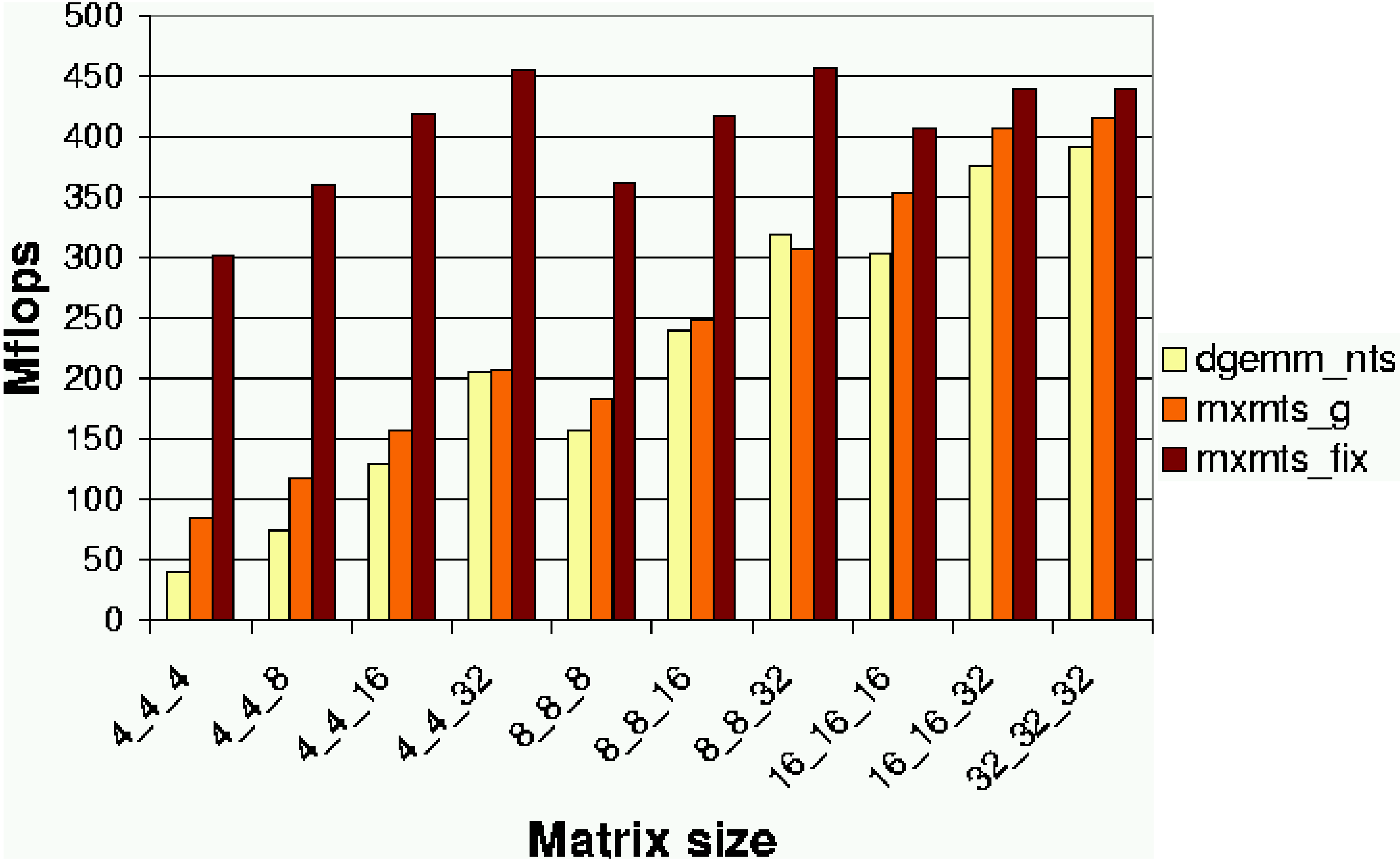


- select best combination
- add object to library

Results: Alpha-21164



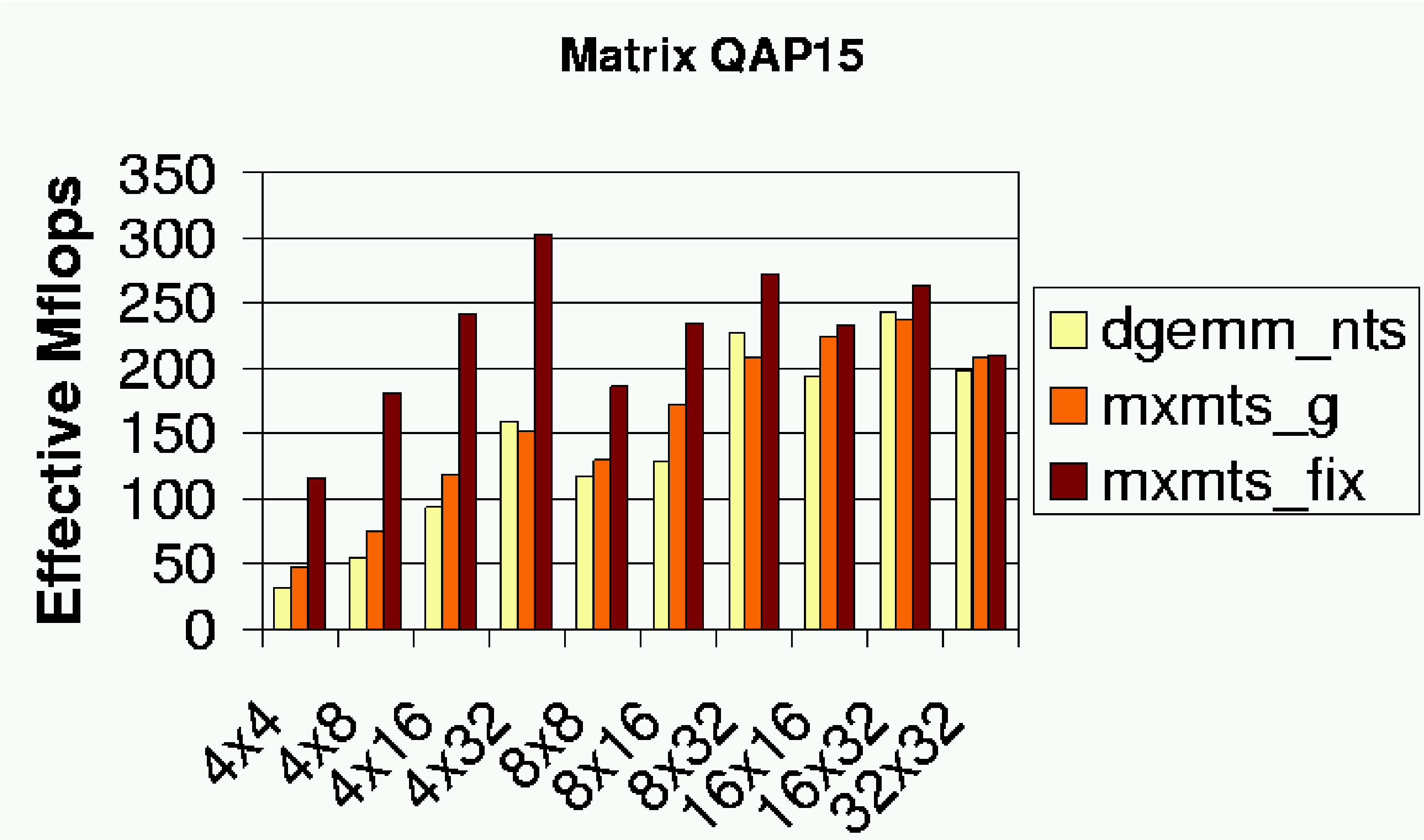
Results: R10000



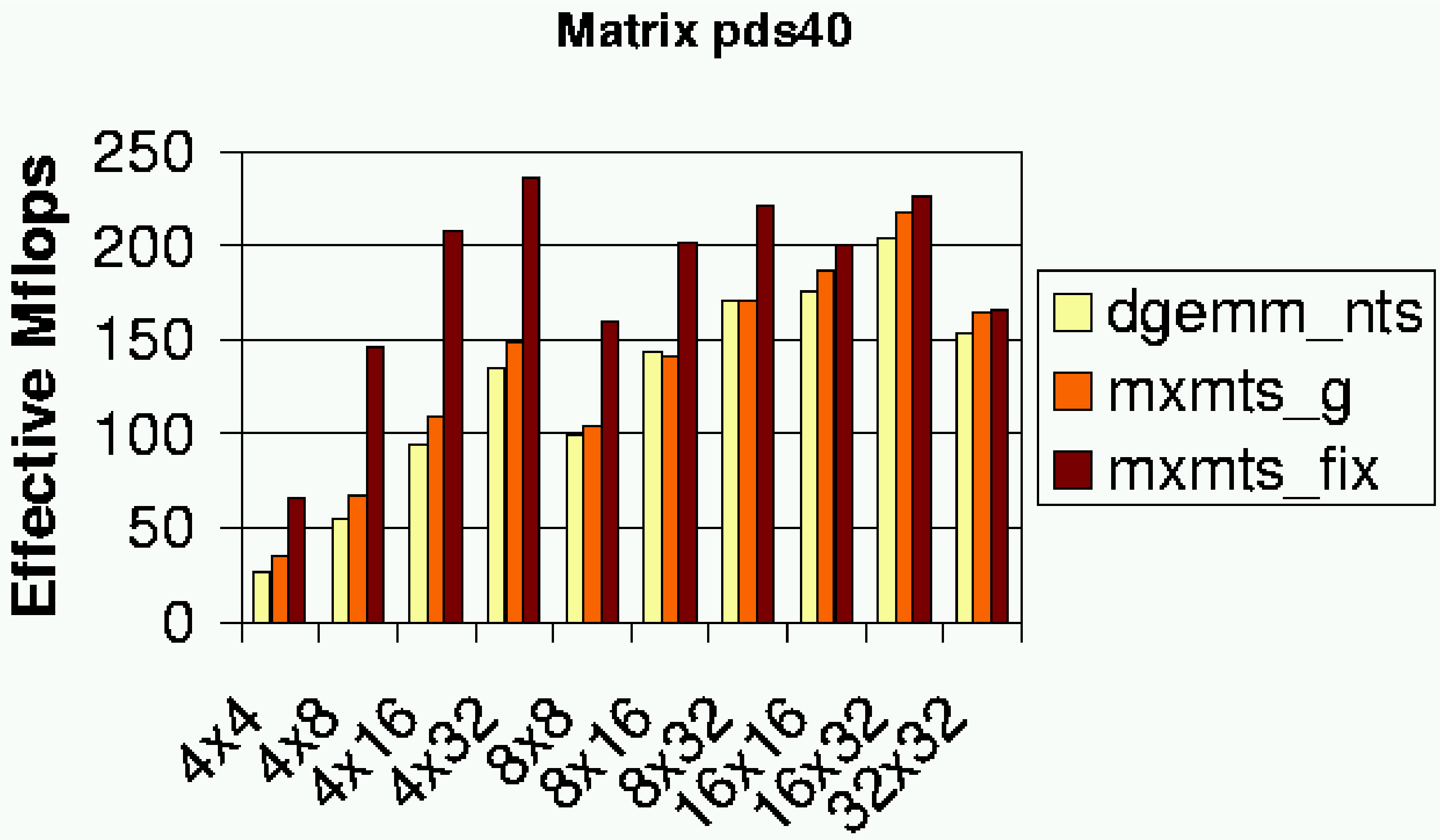
Talk Outline

- Introduction
- Goal
- Building an efficient Small Matrix Library
- Impact on Sparse Hypermatrix Cholesky
- Conclusions

HM Cholesky: Results



HM Cholesky: Results



HM Cholesky: Results

Matrix	*Dimension* (In thousands)	*Factor NZs* (In millions)	*Density*	*Mflops*
TRIPART1	4.2	1.1	0.127	223.0
TRIPART2	19.7	5.9	0.030	226.9
TRIPART3	38.8	17.8	0.023	237.4
TRIPART4	56.8	76.8	0.047	278.2
pds40	76.7	27.6	0.009	236.6
pds50	95.9	36.3	0.007	249.9
pds80	149.5	64.1	0.005	254.4
pds90	164.9	70.1	0.005	263.3
QAP15	6.3	8.7	0.436	303.1

Talk Outline

- Introduction
- Goal
- Building an efficient Small Matrix Library
- Impact on Sparse Hypermatrix Cholesky
- Conclusions

Conclusions

- We can get high performance codes for very small matrix kernels
  - Fixing leading dimensions and loop limits
  - Choosing the algorithm that produces the best result for each case
- Using SML routines we improved the sparse hypermatrix Cholesky
- Creation of ad-hoc codes can yield great performance but have a great creation cost

$\Rightarrow$ We need to do it automatically

Talk Outline

- Introduction
- Goal
- Building an efficient Small Matrix Library
- Impact on Sparse Hypermatrix Cholesky
- Conclusions
- Questions?