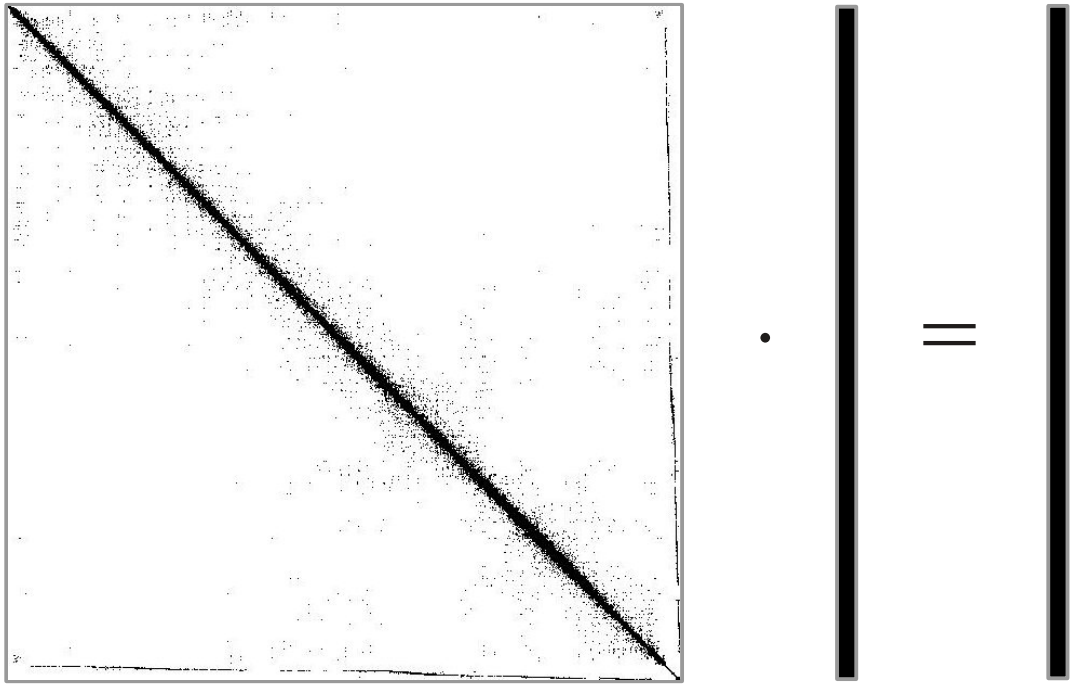
SPARSITY: Optimization Framework For Sparse Matrix Kernels

Eun-Jin Im, Katherine Yelick, Richard Vuduc

International Journal of High Performance Computing Applications 2004 18: 135

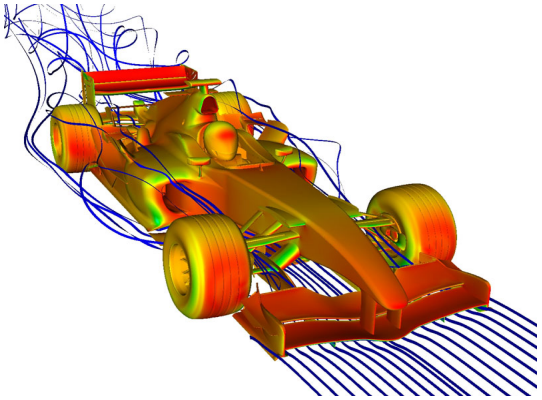
The online version of this article can be found at:
http://hpc.sagepub.com/content/18/1/135
Published by:
http://www.sagepublications.com

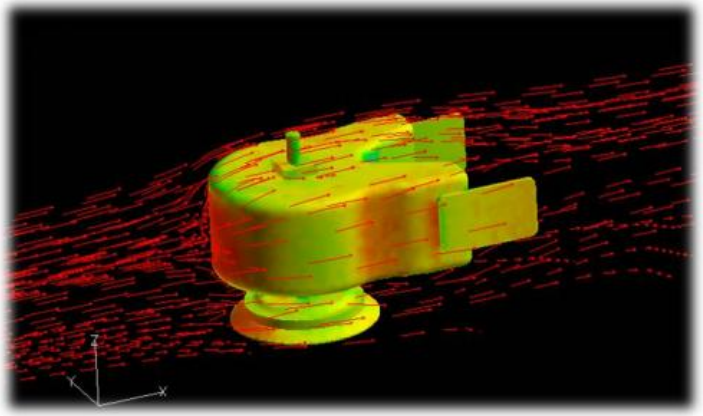
One Operation

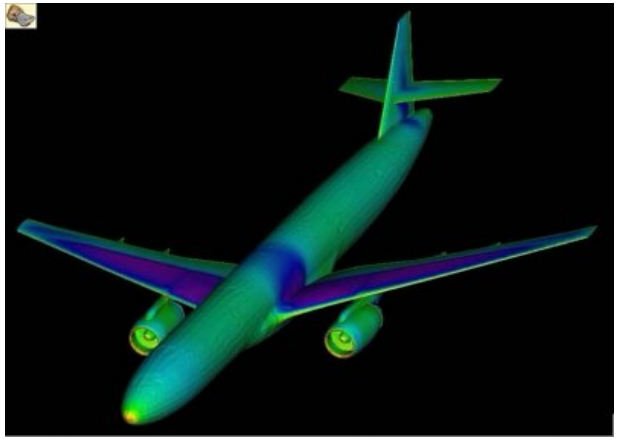


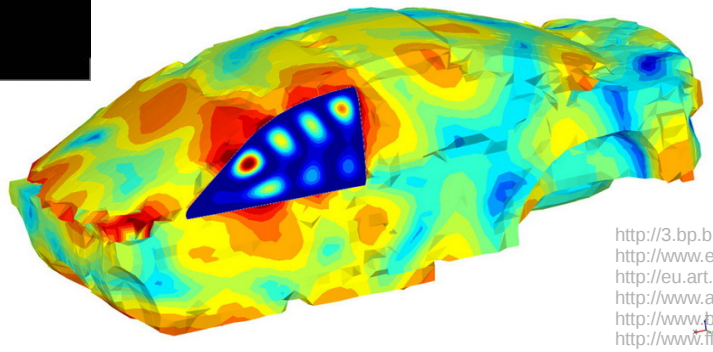
MATLAB, file from http://www.cise.ufl.edu/research/sparse/matrices/Simon/venkat01.html

Motivation









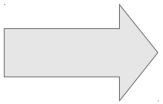
http://3.bp.blogspot.com/-jwj51xaDhsk/Thk3KtjWwsI/AAAAAAAAAOA/P8eNt0_MJUQ/s1600/Challenger2.gif
http://www.erneuerbareenergiequellen.com/pictures/other/oil_some_questions/oil_rig.jpg
http://eu.art.com/products/p14342284-sa-i2886553/posters.htm?ui=BFBAB751660645AA8C02F859E5BAD142
http://www.aspsys.com/userfiles/image/fluent3.jpg
http://www.bloodhoundssc.com/_db/_images/airliner_resized.jpg
http://www.ifit.be/images/documents/219.jpg
http://www.onu.edu/files/images/alumni/Flow_around_object.jpg
http://t0.gstatic.com/images?q=tbn:ANd9GcQDP4JEXQNigtR04rNdj2gBvl8QpO1Sf1k2hcOMF9yXWqP_PCQb

Machines

Processor	Clock (MHz)	Data Cache sizes	DGEMV (MFLOPS)	DGEMM (MFLOPS)
Sun Ultra Sparc IIi	333	L1: 16 KB L2: 2 MB	58	425
Intel Pentium III-Mobile	800	L1: 16 KB L2: 256 MB	147	590
IBM Power 4	1300	L1: 64 KB L2: 1.5 MB L3: 32 MB	915	3500
Intel Itanium 2	900	L1: 16 KB L2: 256 KB L3: 3 MB	1330	3500

CSR: Compressed Sparse Row Format

3	0	0	5
0	1	7	0
0	0	2	0
0	0	0	4



Values:

3	5	1	7	2	4

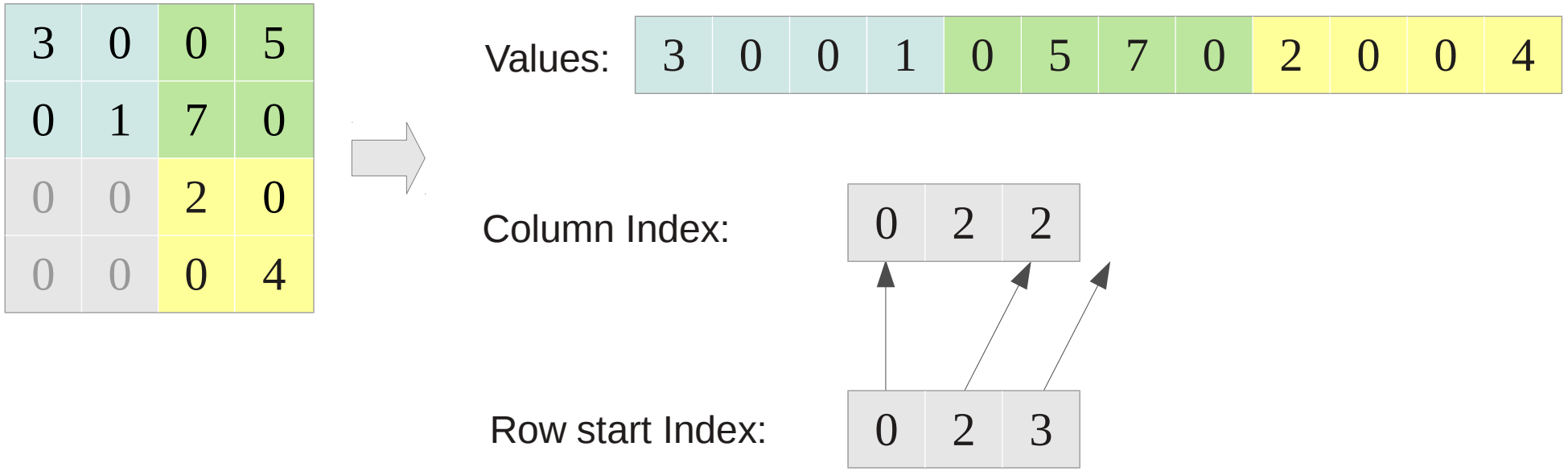
Column Index:

0	3	1	2	2	4

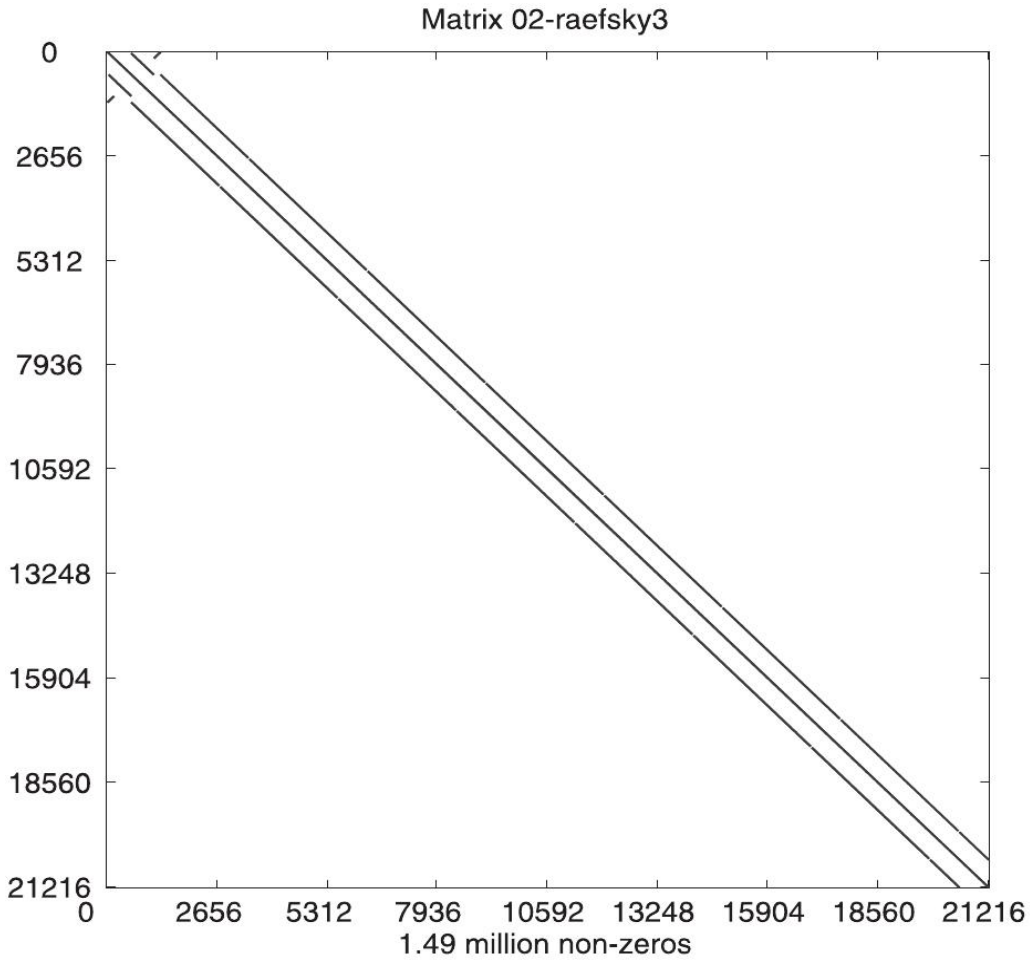
Row start Index:

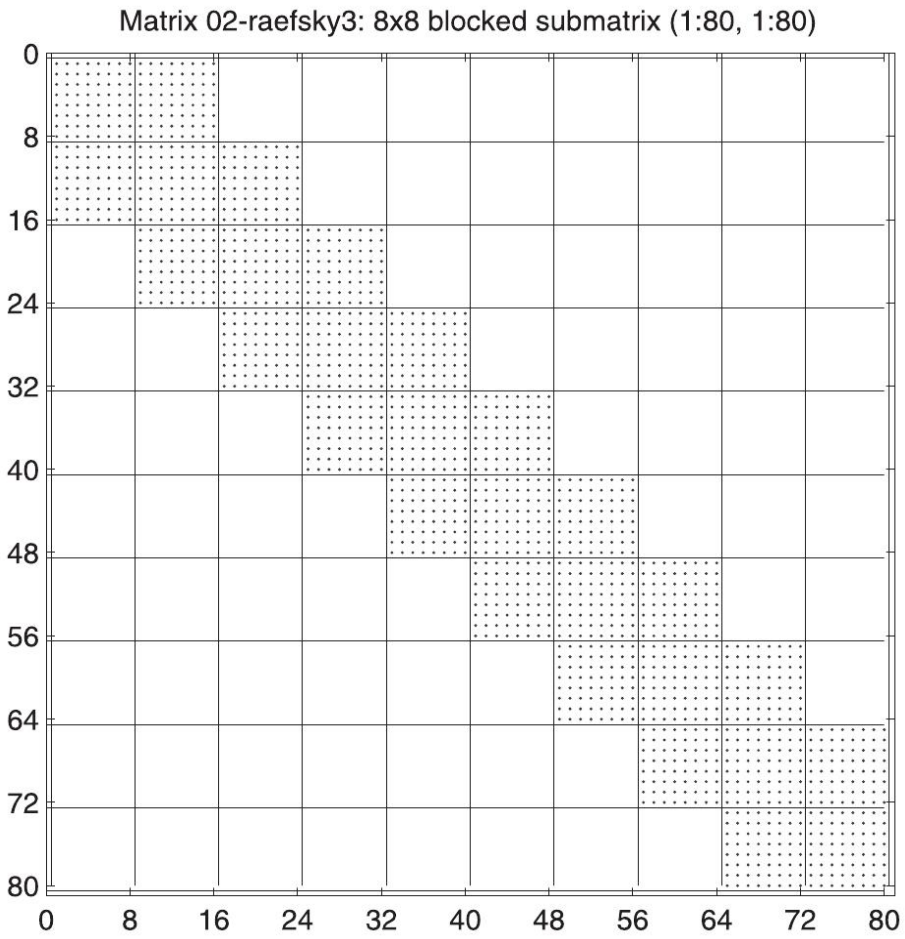
0	2	3	5	6

Register-Blocking

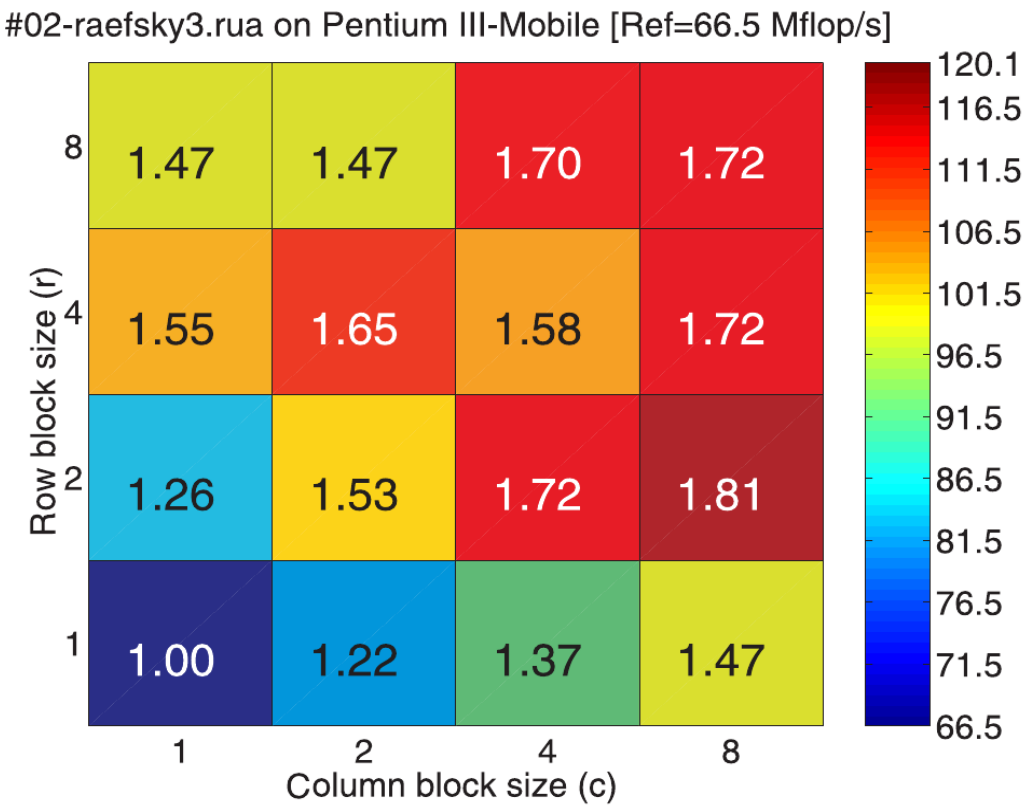


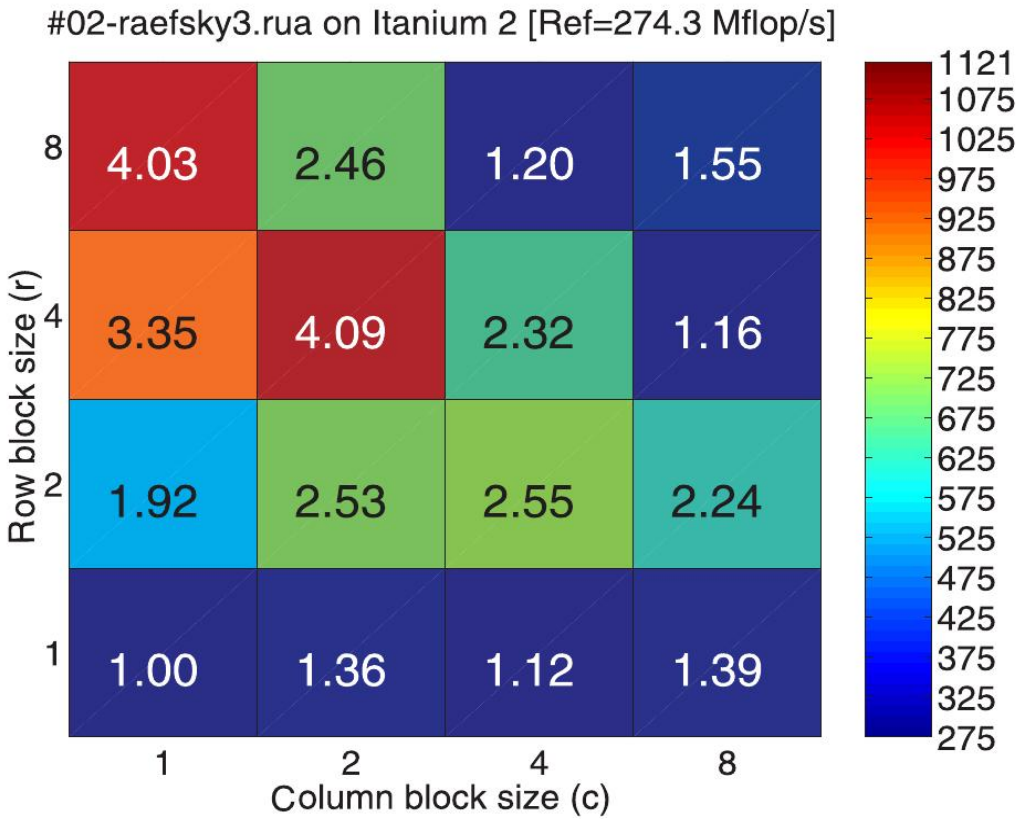
Example for Register-Blocking



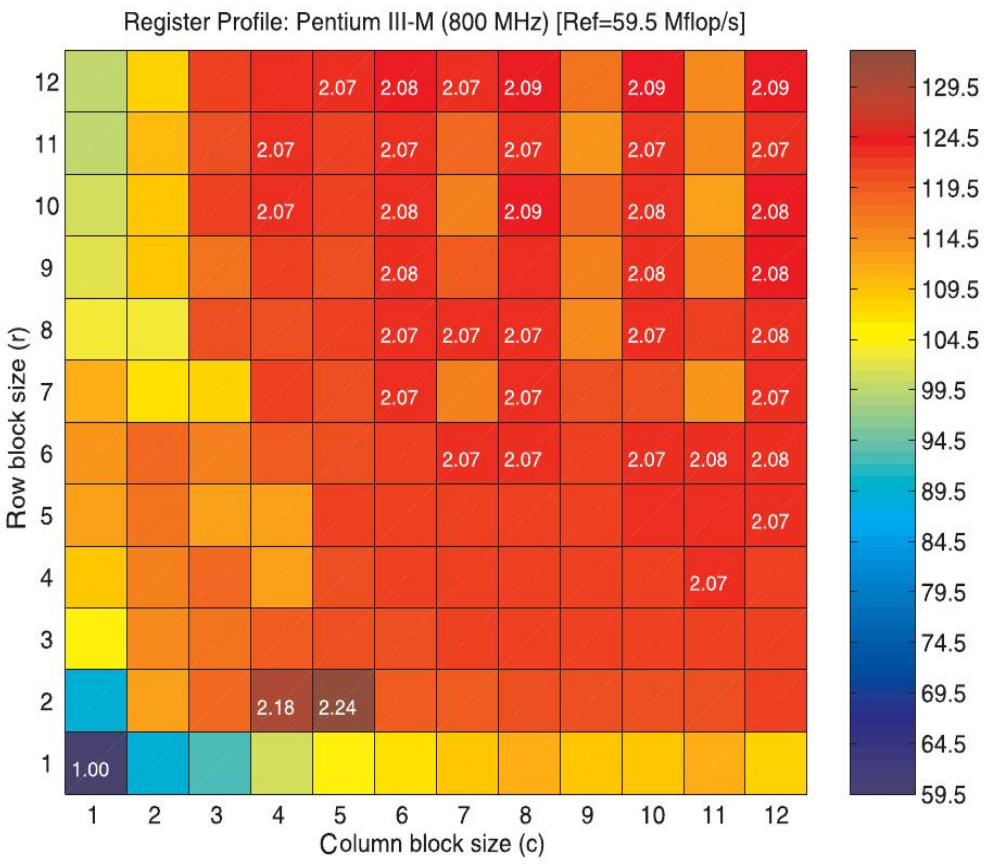


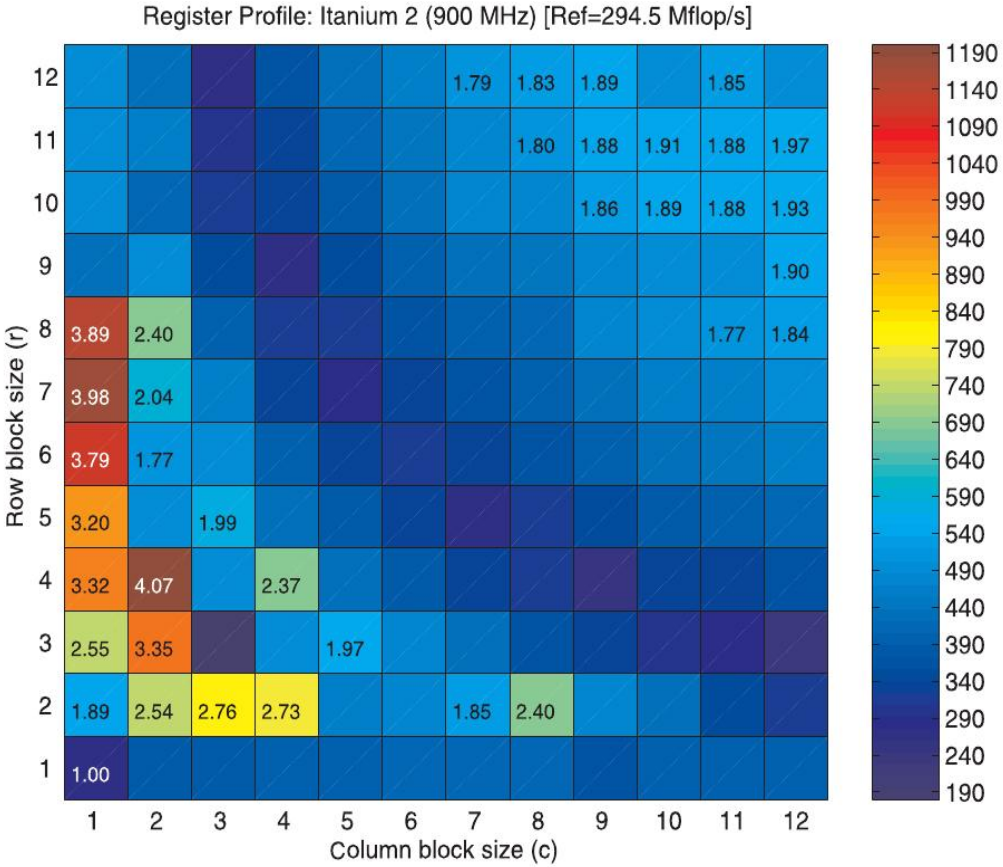
Example Results





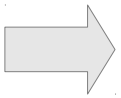
Performance Model: Machine Profile





Performance Model: Fill-Overhead

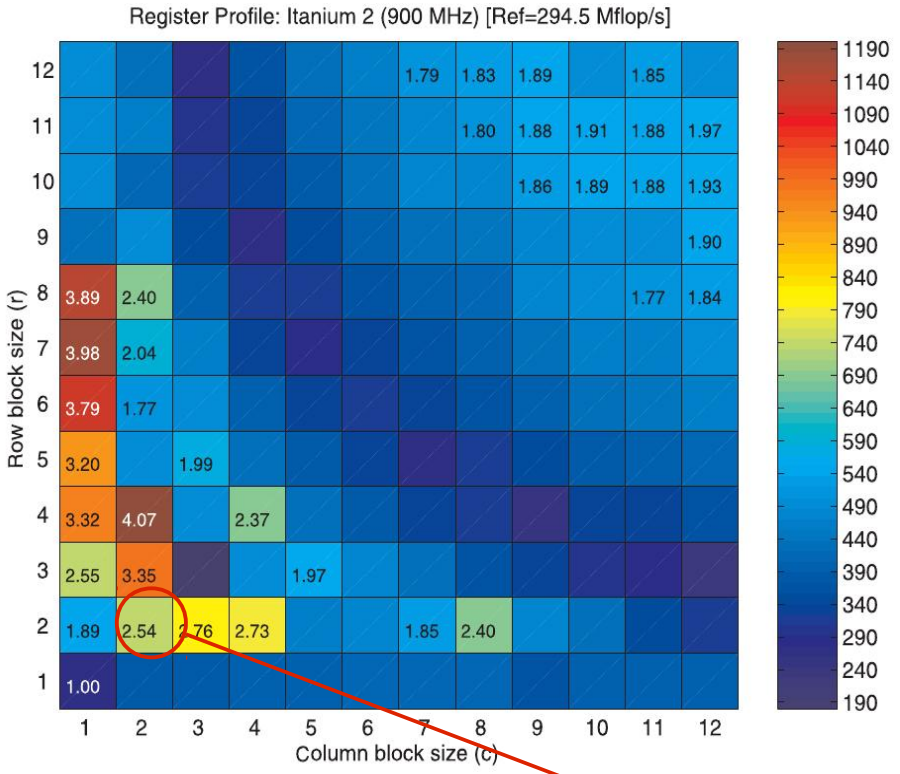
3	0	0	5
0	1	7	0
0	0	2	0
0	0	0	4



$$\frac{12}{6}=2$$

Performance Model

Example on Intel Itanium 2 with 2×2 block-size:

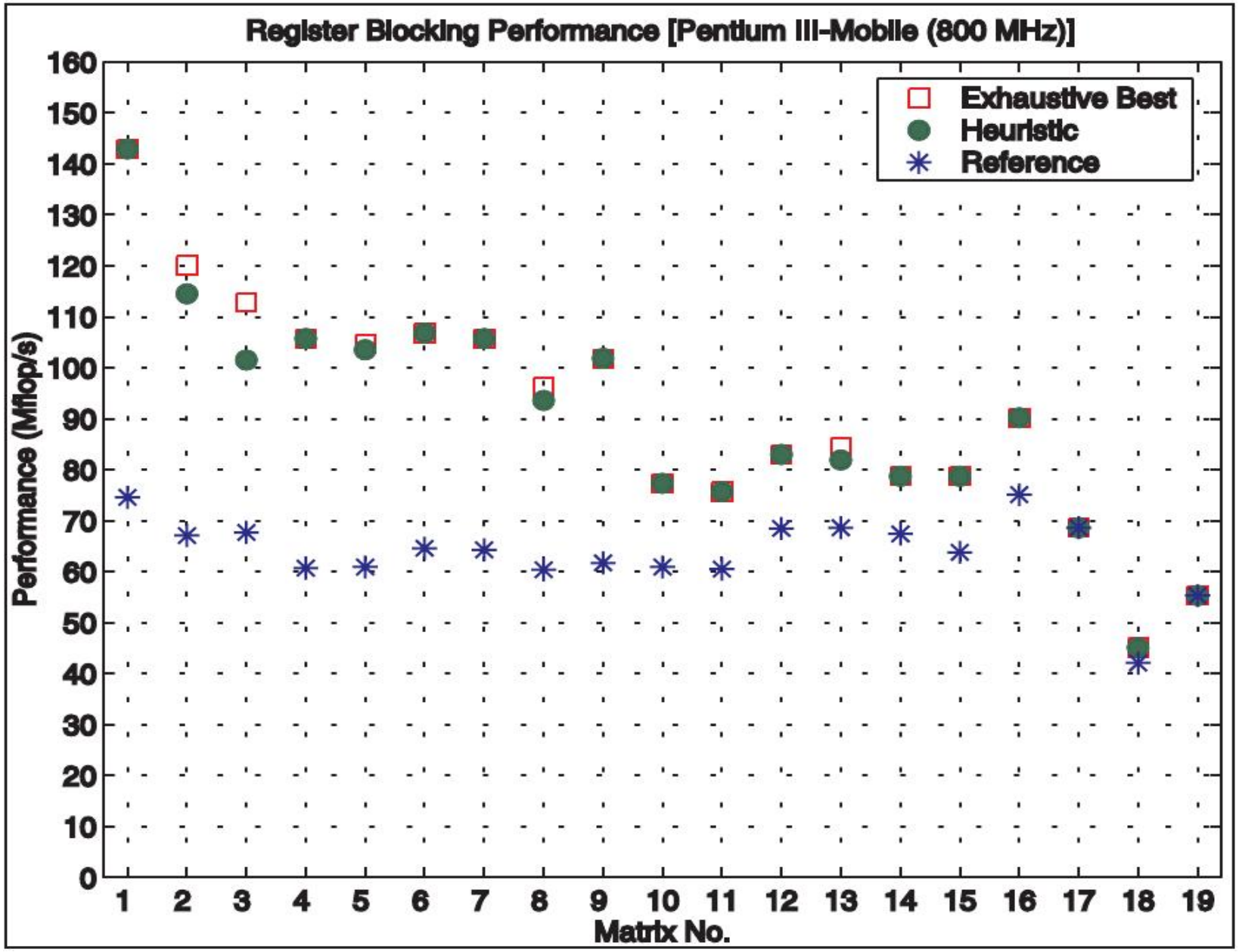


3	0	0	5
0	1	7	0
0	0	2	0
0	0	0	4

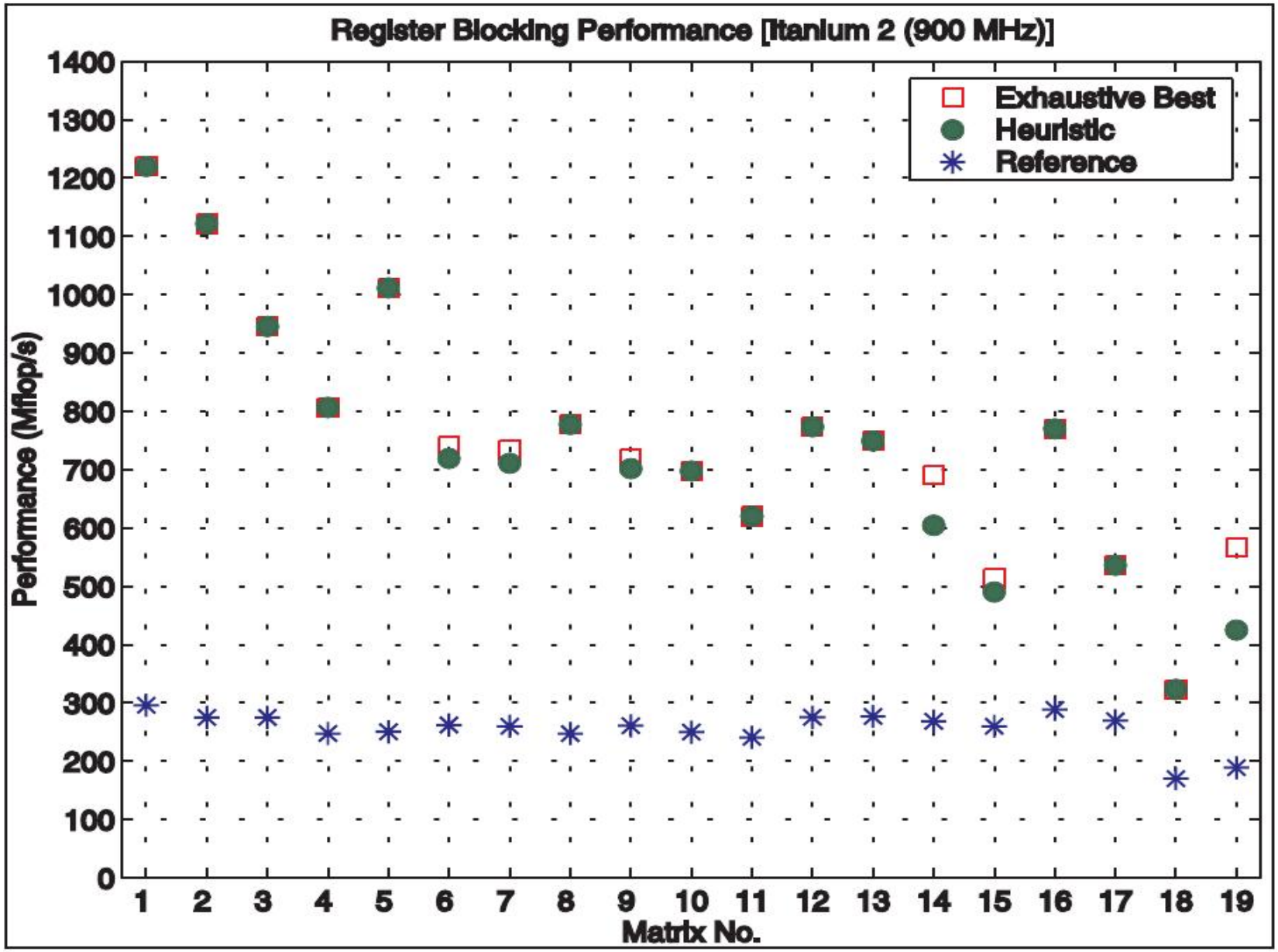
$$\frac{12}{6} = 2$$

$$\frac{2.54}{2} = \underline{\underline{1.27}}$$

Register-Blocking Speedup: Intel Pentium III-M

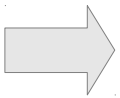


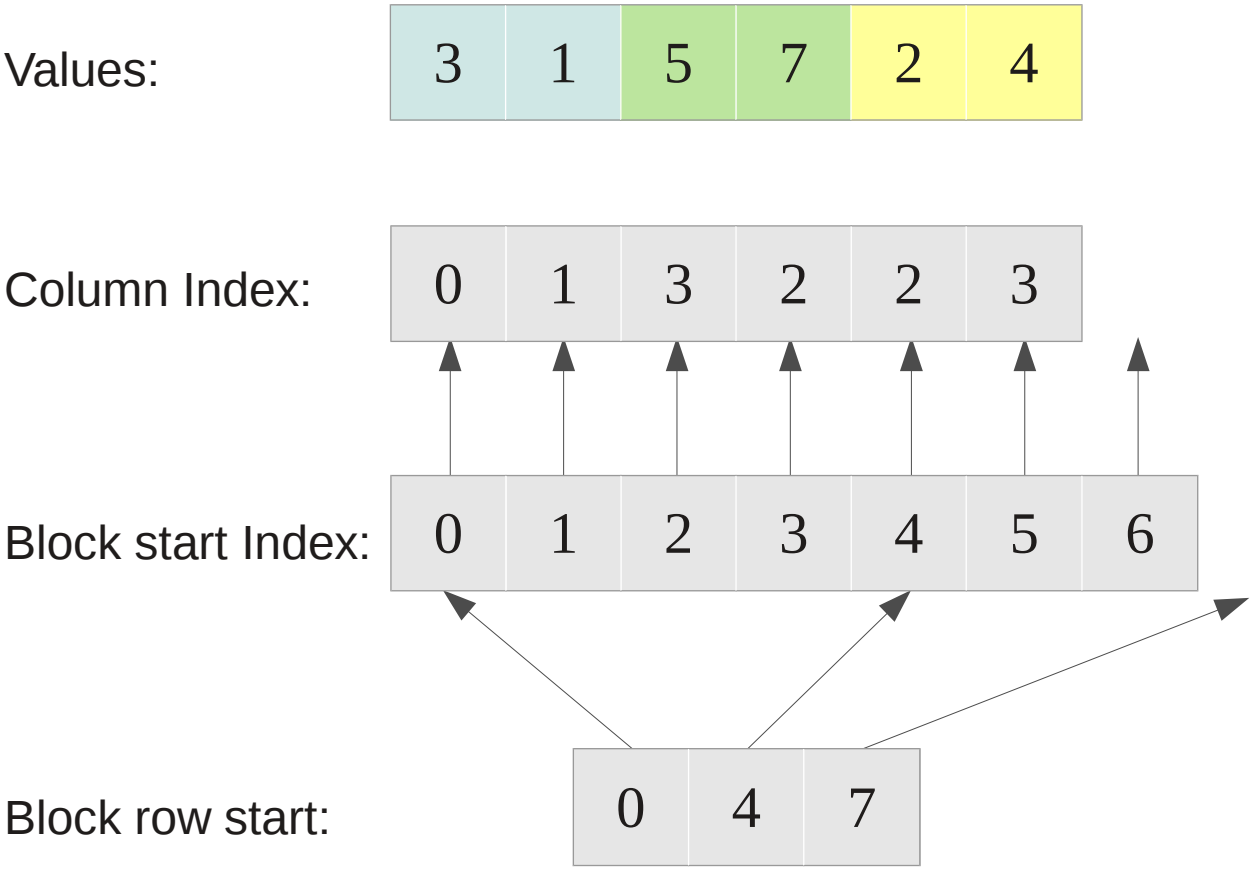
Register-Blocking Speedup: Intel Itanium 2



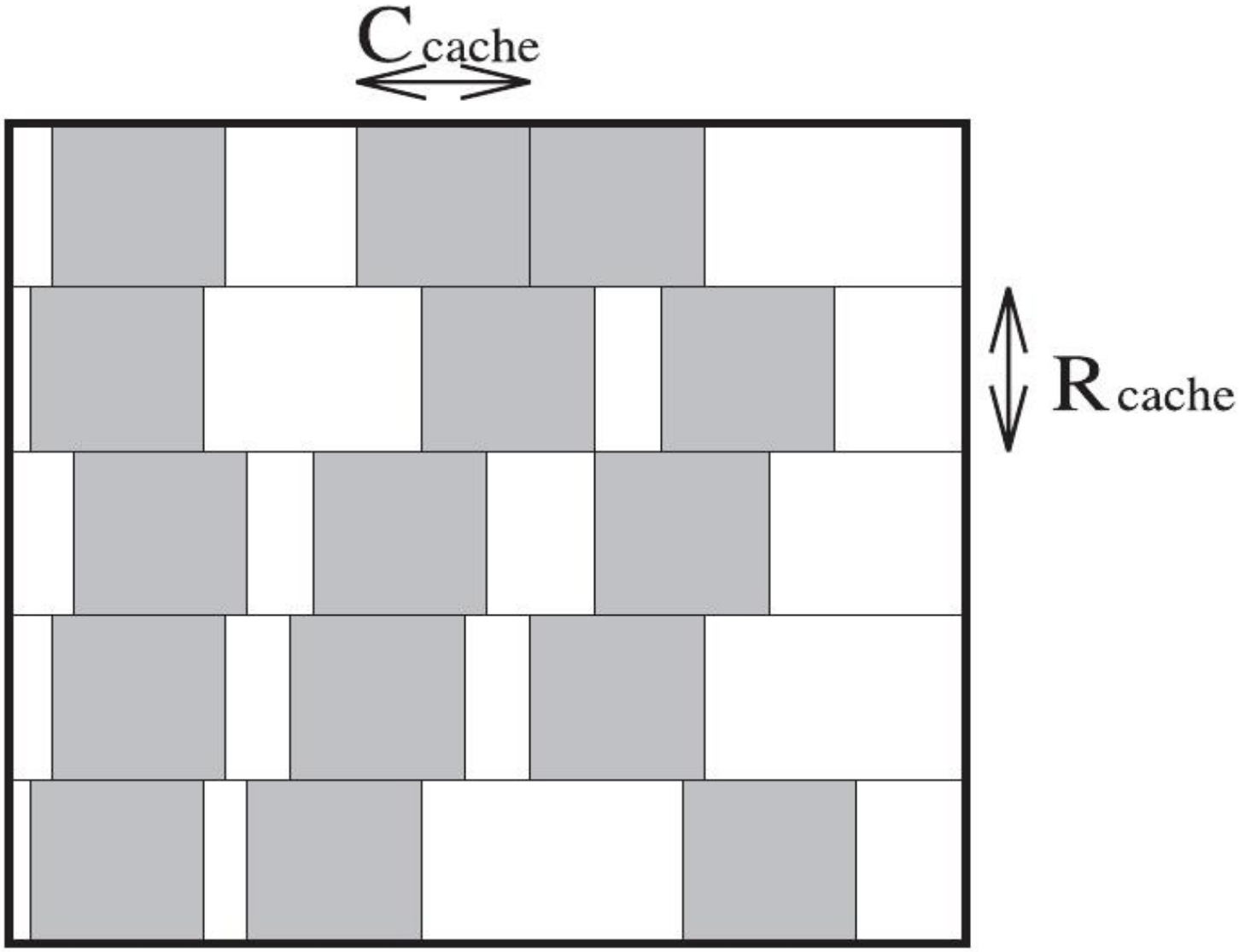
Cache-Blocking

3	0	0	5
0	1	7	0
0	0	2	0
0	0	0	4

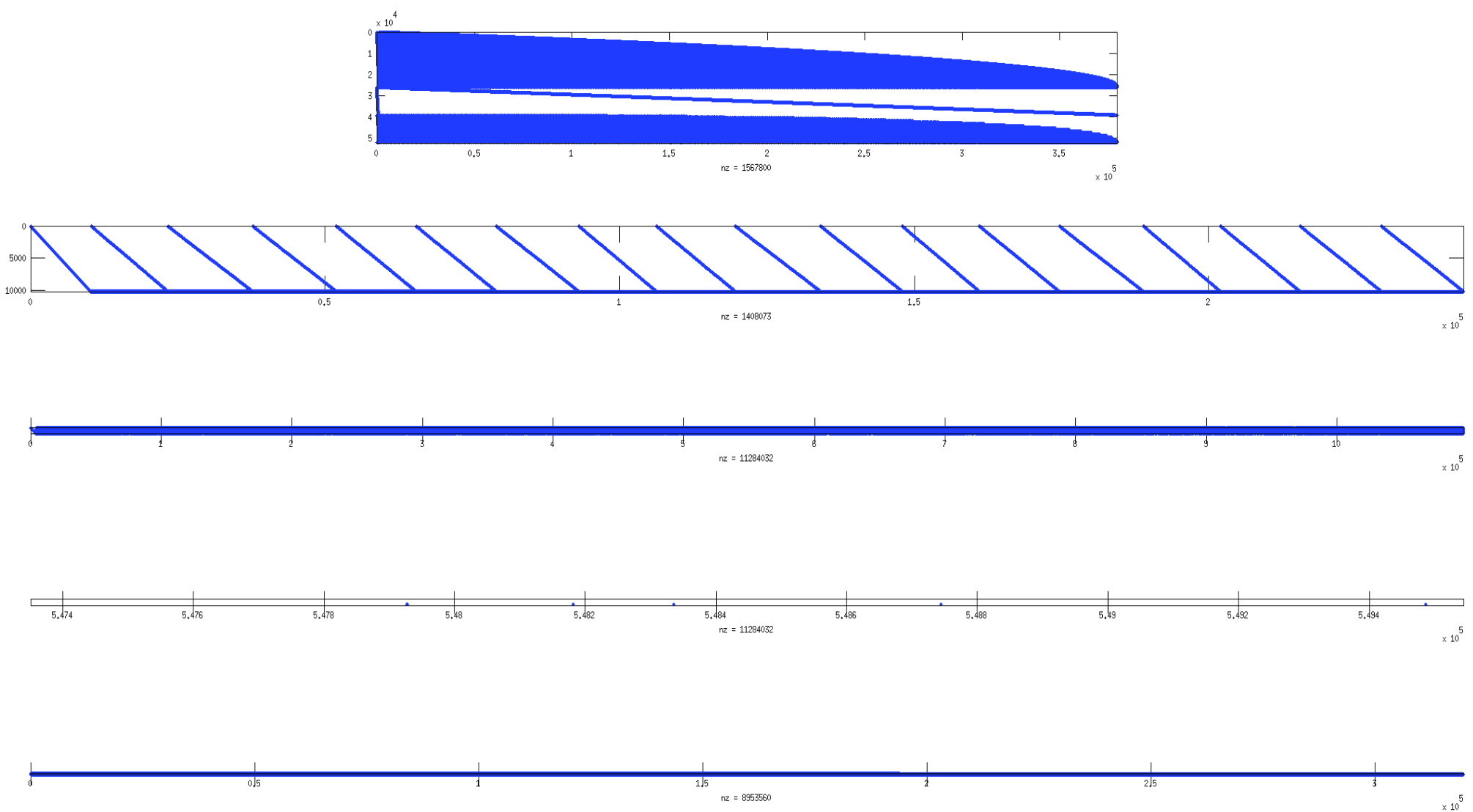




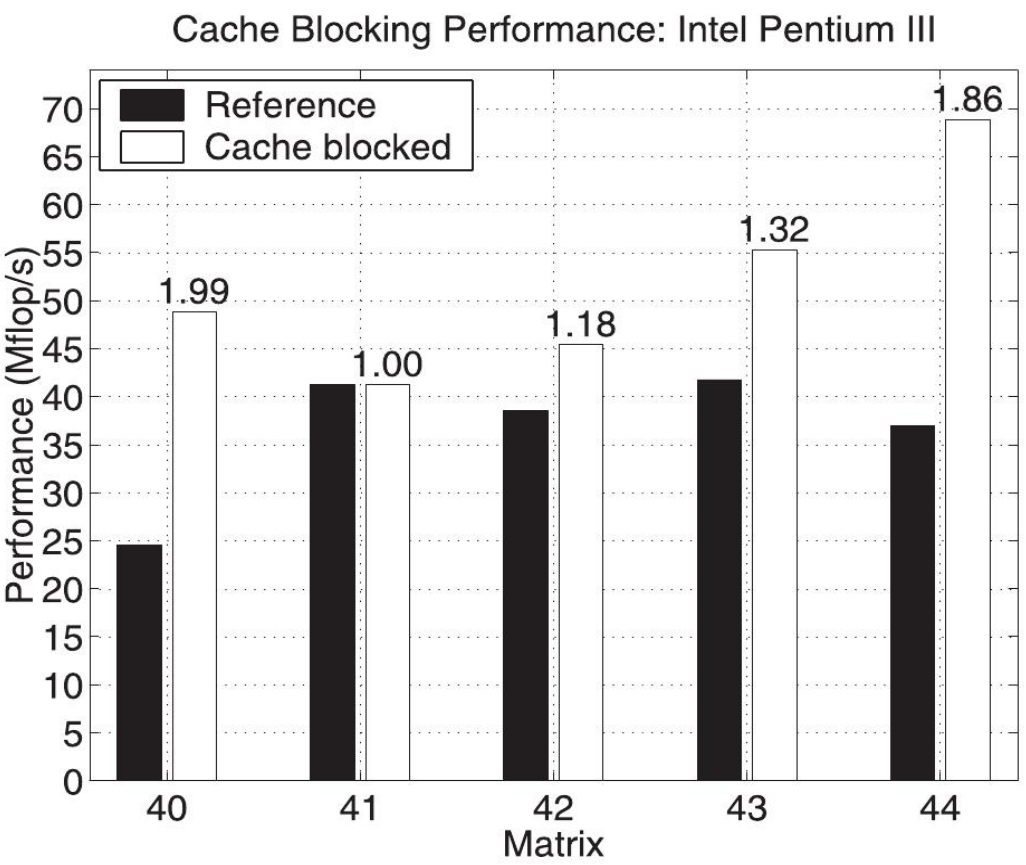
Cache-Blocking

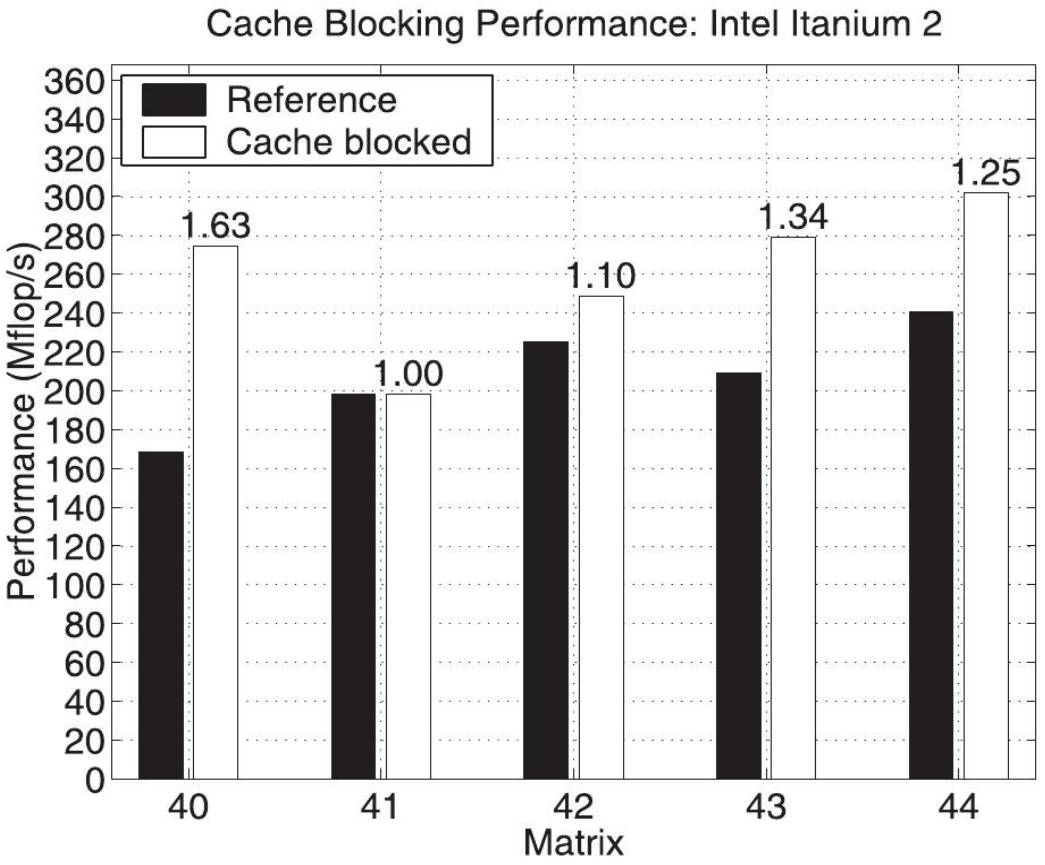


Benchmark Cache-Blocking

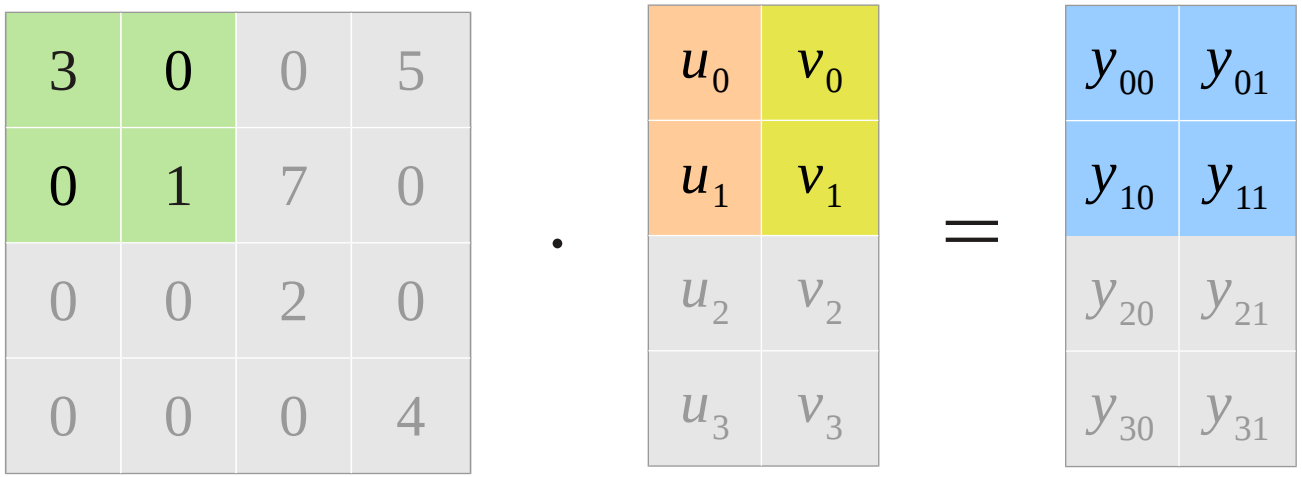


Cache-Blocking Speedup





Multiple Vectors



$$(1) \quad 3 \cdot u_0 + 0 \cdot u_1 = y_{00}$$
$$(2) \quad 0 \cdot u_0 + 1 \cdot u_1 = y_{10}$$

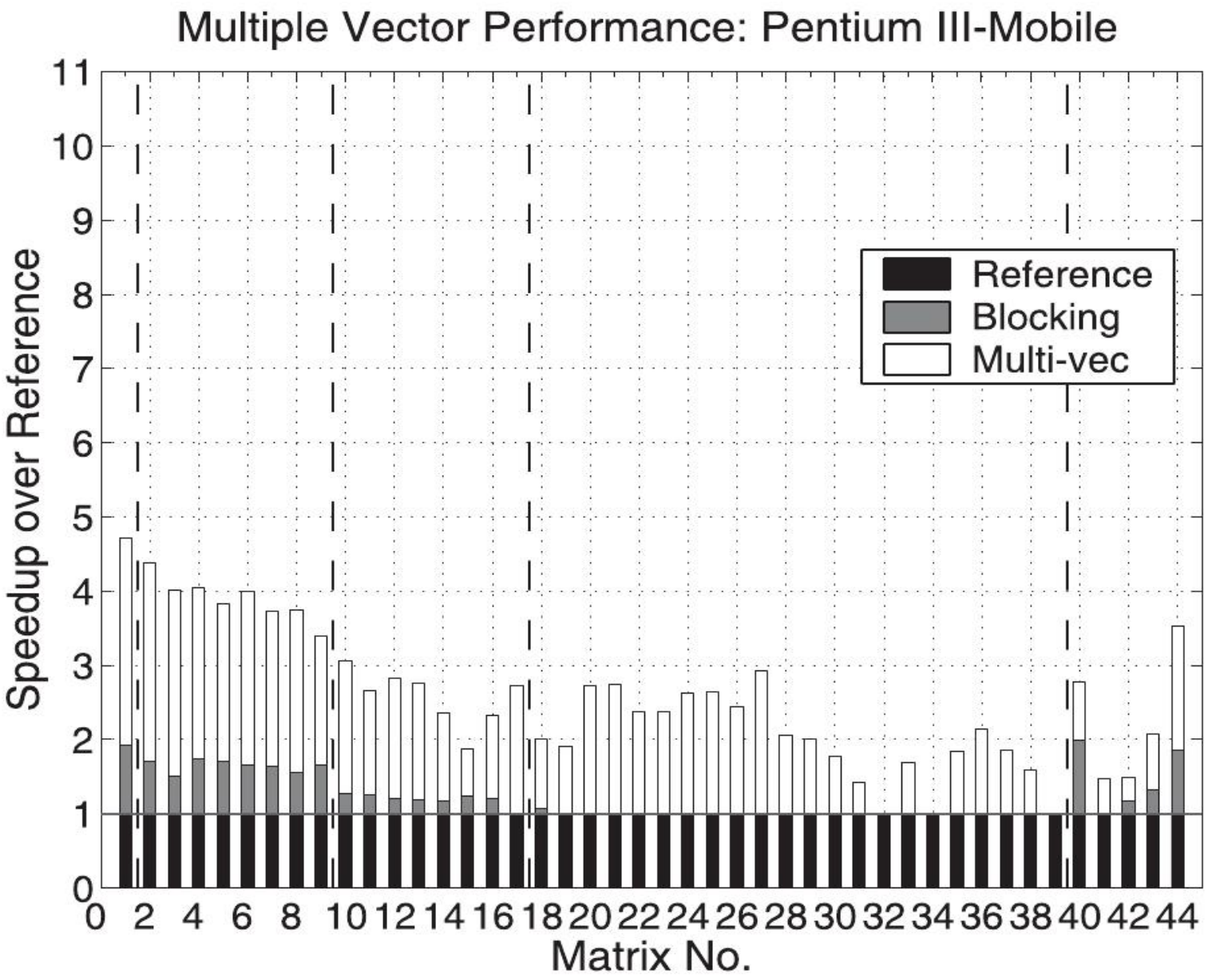
$$\ldots$$

$$(nz+1) \quad 3 \cdot v_0 + 0 \cdot v_1 = y_{01}$$
$$(nz+2) \quad 0 \cdot v_0 + 1 \cdot v_1 = y_{11}$$

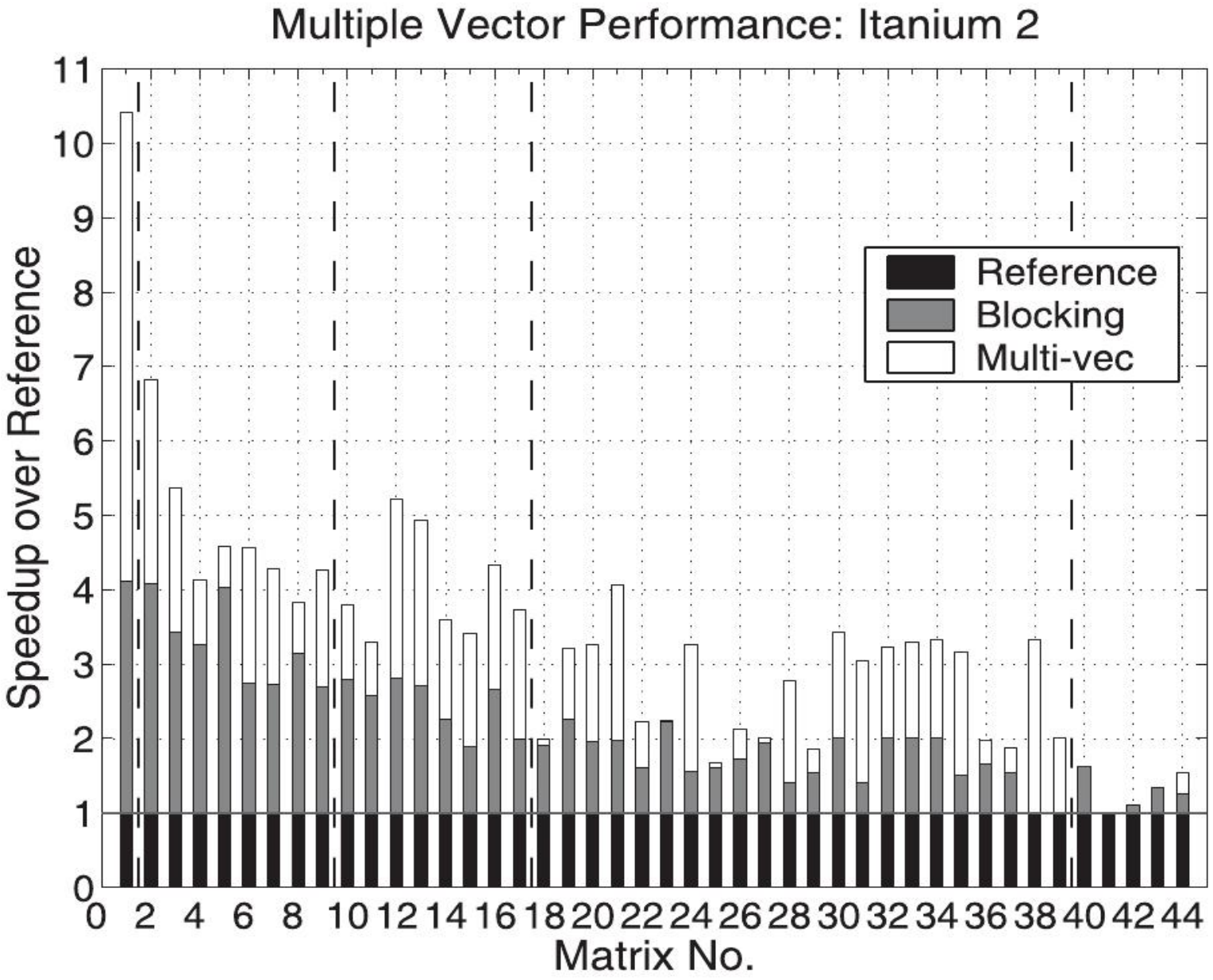
$$(1) \quad 3 \cdot u_0 + 0 \cdot u_1 = y_{00}$$
$$(2) \quad 0 \cdot u_0 + 1 \cdot u_1 = y_{10}$$
$$(3) \quad 3 \cdot v_0 + 0 \cdot v_1 = y_{01}$$
$$(4) \quad 0 \cdot v_0 + 1 \cdot v_1 = y_{11}$$

nz = number of non-zero elements in A

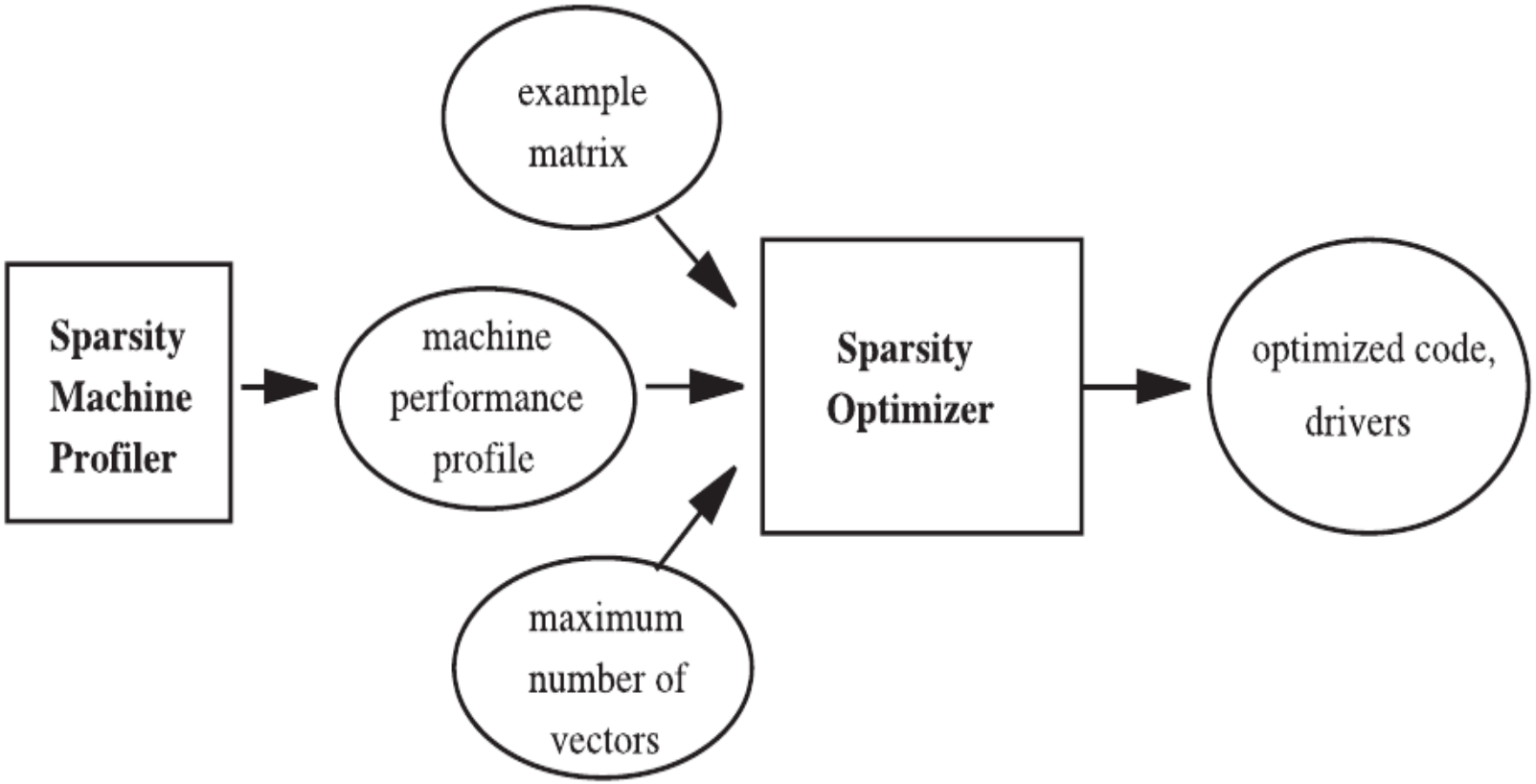
Multiple Vectors Speedup: Intel Pentium III-M



Multiple Vectors Speedup: Intel Itanium 2



SPARSITY System



Graph: Paper

Conclusion

- 4x improvement for register-blocking
- 2x for cache-blocking
- 10x for register-blocking combined with multiple vectors
- Lot of publications in reference to SPARSITY