# Optimal Block Size for Matrix Multiplication Using Blocking

**Sasko Ristov, Marjan Gushev – Ss. Cyril and Methodius University, FSCE**
**{marjan.gushev, sashko.ristov}@finki.ukim.mk**

**Goran Velkoski – Innovation LLC**
**goran.velkoski@innovation.com.mk**

"Ss. Cyril and Methodius" University in Skopje
FACULTY OF COMPUTER
SCIENCE AND ENGINEERING

# Abstract

- **Speeding up** the multiplication of huge matrices **is imperative**
- **Blocking reduces the cache misses**
  - **choosing the block size is not** the **only optimization**
- This paper analyzes the impact of various block size (*M* x *K* and *K* x *N*) on the performance.
  - **Different** parameter values for ***K***
  - **predefined** values of the parameters ***M*** and ***N***,
  - test the algorithm behavior in **different cache regions**.

# Abstract

- The results of the experiments show **three phenomena**.
  - if **M** > **N**, then choosing the block *M* x *N* of the first matrix will achieve a **significantly greater speed**.
  - if the second parameter ***N*** is increased for constant ***M*** has no significant influence on the performance.
  - the **speed decreases significantly** if ***N*** is **increasing** for constant ***M***.

# Outline

- **Background & Motivation**
- Testing Methodology
- The Results of the Experiments
- Discussion
- Conclusion & Future Work

# Background – Cache Associativity vs. Blocking

- Blocking algorithm improves the MM
  - Only L1 cache size
  - Does not takes into account the cache set associativity problem !!!

S. Ristov and M. Gusev, "**Achieving Maximum Performance for Matrix Multiplication using Set Associative Cache**", in 8th Int. Conf Computing and Information Management ICCM, IEEE Conference proceedings, vol. ICNIT 2012, Seoul, Korea, 2012, pp.542-547.

M. Gusev and S. Ristov, "**Performance Gains and Drawbacks using Set Associative Cache**", JNIT, Journal of Next generation Information Technology, ISSN: 2233-9388, 2012, Volume 3, Number 3, pp.87-98.

# Background – Improved Blocking

- Recently we proposed 1D/2D blocking MM
  - Better for AMD (low associativity)

M. Gusev, S. Ristov, and G. Velkoski, "**Hybrid 2D/1D Blocking as Optimal Matrix Multiplication**", in ICT Innovations 2012, Advances in Intelligent and Soft Computing, (ed. S. Markovski and M. Gusev), Springer Verlag, Berlin Heidelberg, 2013, volume AISC 257, pp.13-22.

- In this paper, **determine the optimal block dimensions** *M* x *K* and *K* x *N*
  - the **same number of operations** is executed
  - **Improve memory access time**

# Outline

- Background & Motivation
- **Testing Methodology**
- The Results of the Experiments
- Discussion
- Conclusion & Future Work

# Testing Algorithms

- Matrix elements are **double**
- Choosed values to **mitigate associativity problem**
  - **56 = 64-8**
  - **224 = 56 x 4**
  - **896 = 56 x 16**

| Test Case | Notation | $BA_{M \cdot K}$ | $BB_{K \cdot N}$ |
|---|---|---|---|
| 1 | 56x56 | $56 \cdot K$ | $K \cdot 56$ |
| 2 | 56x224 | $56 \cdot K$ | $K \cdot 224$ |
| 3 | 56x896 | $56 \cdot K$ | $K \cdot 896$ |
| 4 | 224x56 | $224 \cdot K$ | $K \cdot 56$ |
| 5 | 224x224 | $224 \cdot K$ | $K \cdot 224$ |
| 6 | 224x896 | $224 \cdot K$ | $K \cdot 896$ |
| 7 | 896x56 | $896 \cdot K$ | $K \cdot 56$ |
| 8 | 896x224 | $896 \cdot K$ | $K \cdot 224$ |
| 9 | 896x896 | $896 \cdot K$ | $K \cdot 896$ |

  - **Cover all cache regions**

# Testing Environment

- Intel Xeon CPU X5647 @ 2.93GHz with 8GB RAM memory.
  - quad core,
  - each core has its own private L1 (32KB) and L2 (256KB) caches.
  - All 4 cores share L3 cache of 12MB.

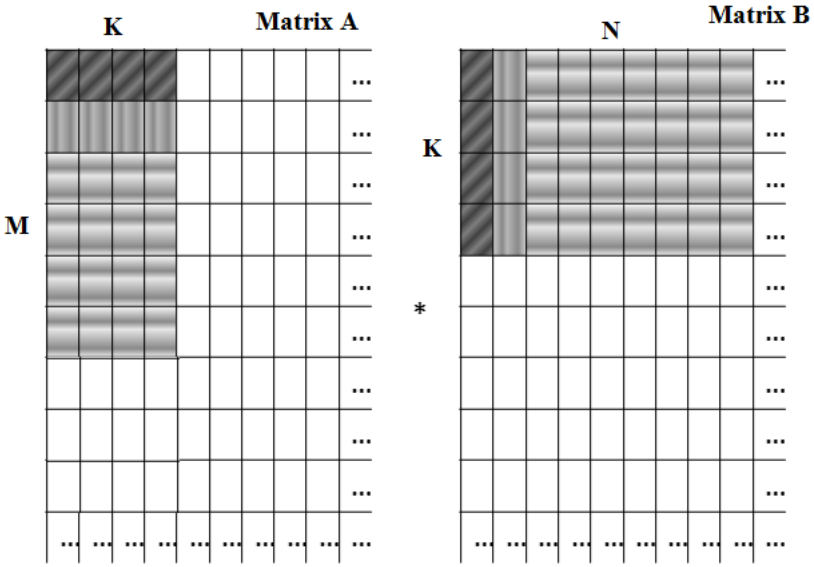
# Test Data

- The execution time ***T****(M, N, K)* changing *K*
- Calculate ***Speed V****(M, N, K)*

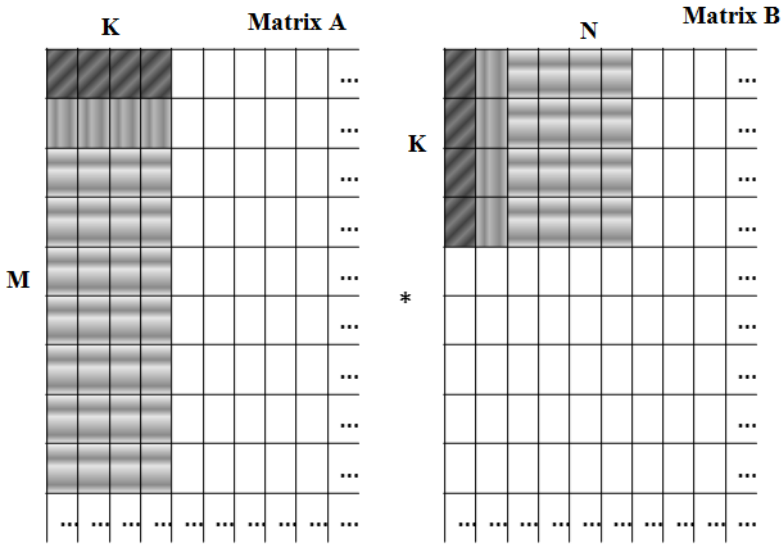
$$V(M,N,K) = \frac{2 \cdot M \cdot N \cdot K}{T(M,N,K)}$$

# Experiments

- M<N




M > N



# Three Test Goals

- The First Goal
  - determine **which pair of block sizes** provides **better performance** if the **blocks** have the **same number of elements**, but **exchanged parameters**.
  - For example
    - $M=56 < N=224$
    - $M=6 < N=9$.
  - The parameter $K$ is variable.

# Three Test Goals

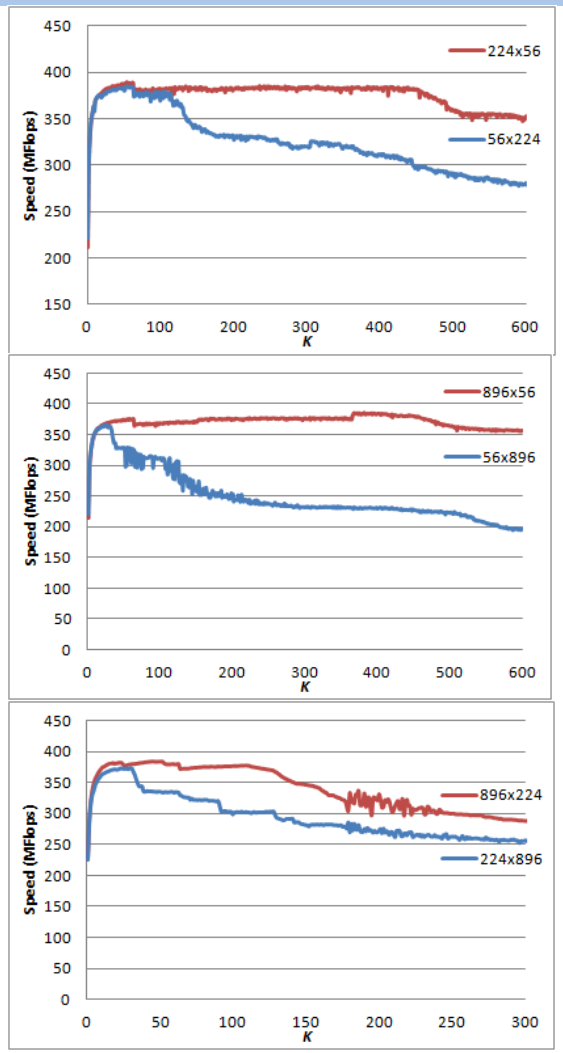
- The **second** goal
  - determine the **impact of** the parameter ***M***
- The **third** goal
  - determine the **impact of** the parameter ***N***
- **Varying** the parameter ***K***

# Outline

- Background & Motivation
- Testing Methodology
- **The Results of the Experiments**
- Discussion
- Conclusion & Future Work

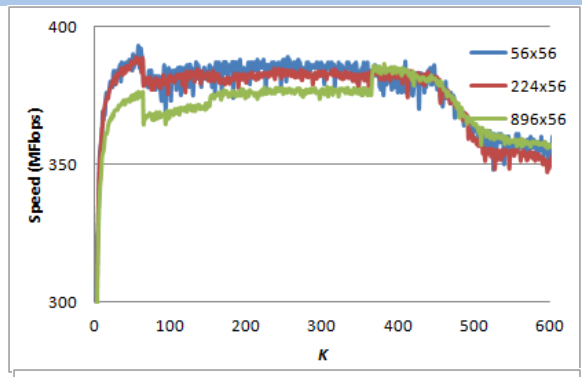
# Horizontal or Vertical Rectangle?!

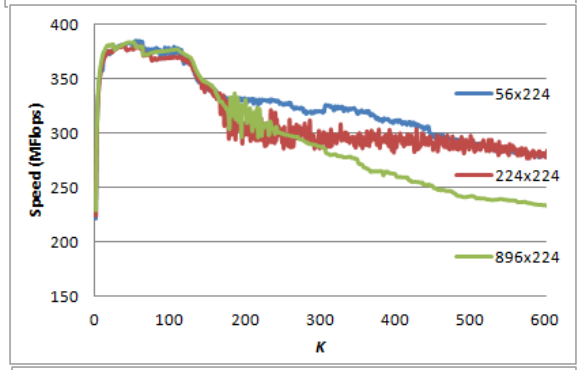
- Similar results for all cases
- Both curves are identical until some ***K***
  - the point when both matrices together exceed L2 cache (Private per core)
- **Better when *M* > *N***

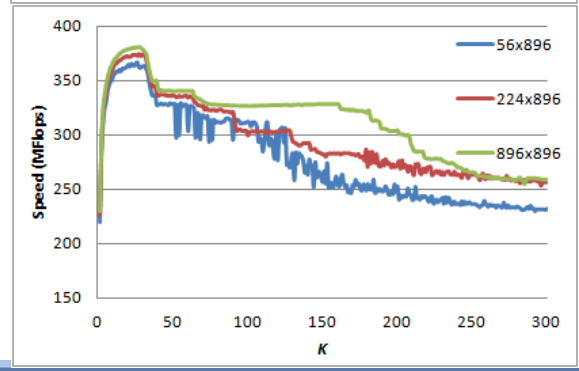


# The *M*'s Impact

- Increasing ***M* does not impact** the performance when matrices can be **placed in cache**
- **Similar** results for all three values of ***N***.

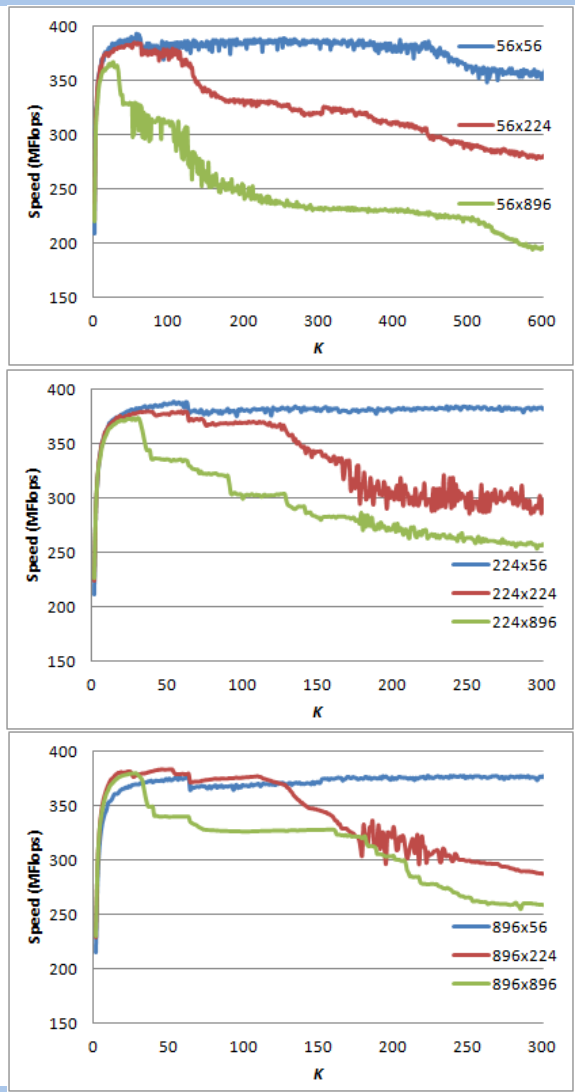


# The *N*'s Impact

- **Increasing *N* negatively impacts** the algorithm performance,
  - emphasized for greater ***K***.



# Outline

- Background & Motivation
- Testing Methodology
- The Results of the Experiments
- **Discussion**
- Conclusion & Future Work

# Discussion

- **Choosing** a rectangle **depends** only **when** matrices **cannot be placed in L2** cache,
  - **favoring** the rectangle with **greater *M* - vertical rectangle**.

- **VERY IMPORTANT**:
  - Speed is increased until L2 (Not until L1) **for horizontal rectangle**, while the speed **keeps its value for vertical rectangle**.

- This leads to conclusion:
  - the **blocking can be choosed with greater blocks (Not in L1)** since **the number of operations will be smaller**, and thus **the overall execution will be faster** than traditional blocking.

# Discussion

- The impact of M and N is totally different
  - ***M*** has a small impact to the performance, especially for smaller values of the parameter $N$.
    - **Means** that **blocks can be even > L1** cache size.
  - **Increasing** the ***N*** **significantly reduces** the algorithm performance

- The **common**
  - **increased impact** for **greater *K***

# Outline

- Background & Motivation
- Testing Methodology
- The Results of the Experiments
- Discussion
- **Conclusion & Future Work**

# Conclusion & Future Work

- Three phenomena
- Discussed results
  - **Greater than L1**

- Other values of parameters ***M*** and ***N***
- Parallelization of these experiments using
  - multi-core CPUs,
  - GPUs (also have set associative caches)

# THANK YOU FOR YOUR ATTENTION

- QUESTIONS?