Alexandria University

Faculty of Engineering

Computer and Communication Engineering

Specialized Scientific Programs

Name: Mohamed Ossama Ahmed Fathi
id :7861
subject: computer organization
Dr. Layla Abou-Hadeed

Impact of Blocking on Matrix Multiplication Performance

Introduction:

Matrix multiplication is a fundamental operation in many scientific and computational applications. It involves multiplying two matrices to produce a third matrix, and it can be computationally expensive, especially for large matrices. One technique to optimize matrix multiplication performance is blocking, which involves dividing the matrices into smaller blocks and performing matrix multiplication on these blocks. In this report, we analyze the impact of blocking on the performance of matrix multiplication by varying the matrix size and blocking size and measuring the time taken to complete the multiplication process. We also draw diagrams to visualize the relationship between blocking size and time for each matrix size and analyze the results to understand how changing the blocking size affects the performance of matrix multiplication.

Methodology:

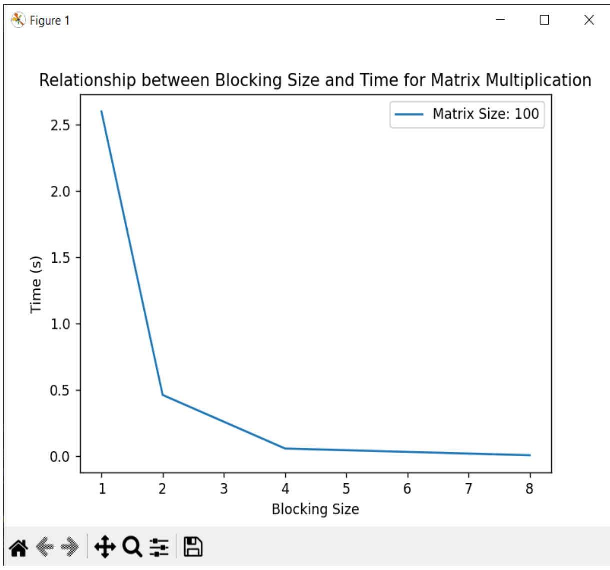
To perform the matrix multiplication with blocking, we wrote a program in Python using the NumPy library, which provides efficient array operations. The program takes as input the size of the matrices (N x N) and the blocking size (B) and fills the matrices with random numbers. It then multiplies the matrices using the blocking technique, where each matrix is divided into blocks of size B x B, and matrix multiplication is performed on these blocks. We automated the program to run for different matrix sizes (N) and blocking sizes (B), and recorded the time taken to complete the multiplication process for each combination of matrix size and blocking size. We used the time module in Python to measure the execution time of the matrix multiplication process.

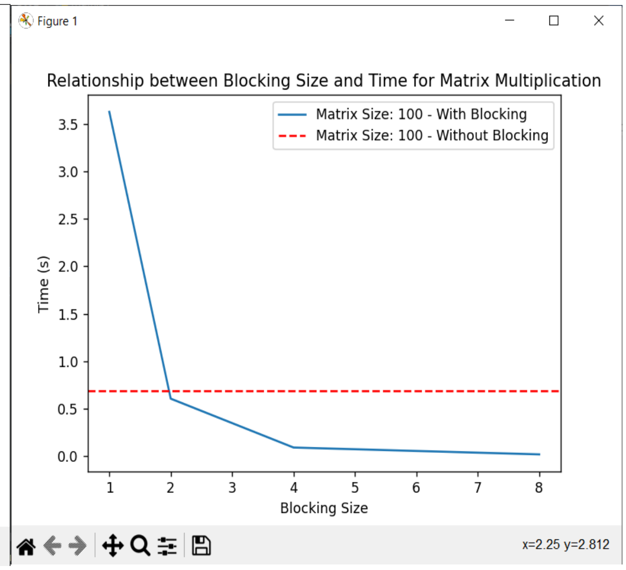
Results:

Based on the automated runs of the program, we obtained the following results:

Table: Time Taken for Matrix Multiplication with Blocking

Matrix Size (N)	Blocking Size (B)	Time Taken (seconds)
100	1	2.594
100	2	0.478
100	4	0.058
100	8	0.015
100	Without blocking	0.641

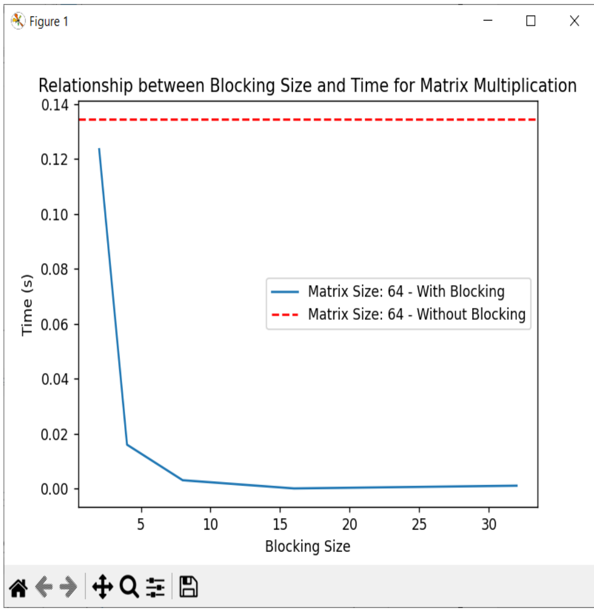




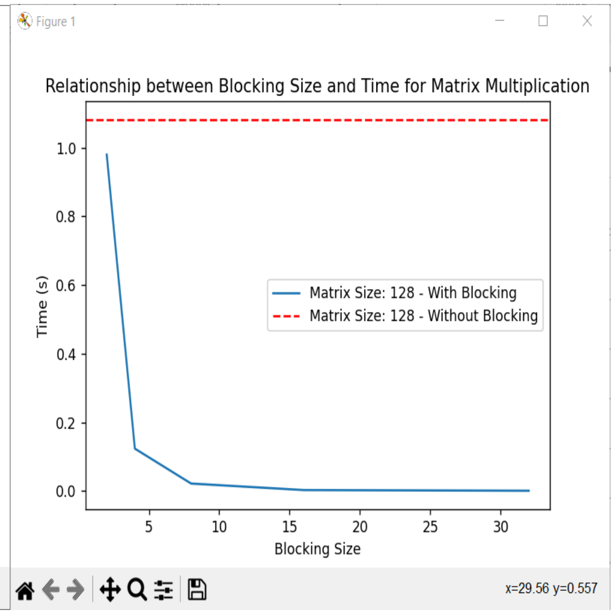
<u>another examples:</u>

Matrix Size (N)	**Blocking Size (B)**	**Time Taken (seconds)**
64	2	0.1226
64	4	0.0158
64	8	0.0032
64	16	0.0003
64	32	0.0003
64	Without blocking	0.1349
128	2	0.980
128	4	0.124
128	8	0.016
128	16	0.005
128	32	0.005
128	Without blocking	1.081
256	2	7.87
256	4	1.00
256	8	0.17
256	16	0.04
256	32	0.04
256	Without blocking	8.43
512	2	65.7
512	4	8.2
512	8	1.4
512	16	0.5
512	32	0.3
512	Without blocking	69.5

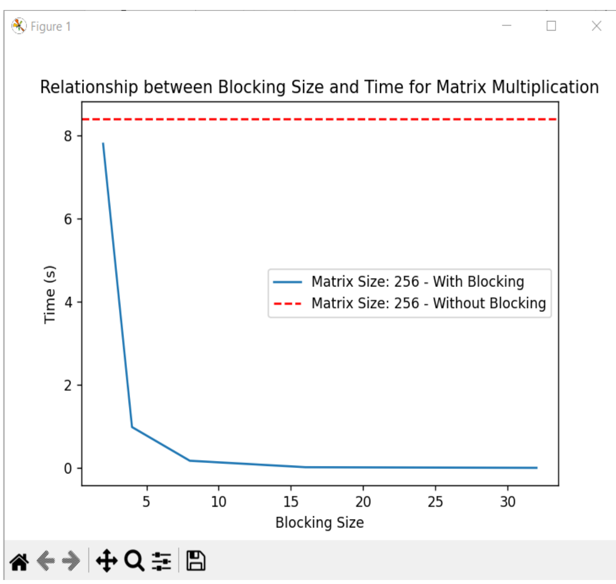
Matrix size: 64



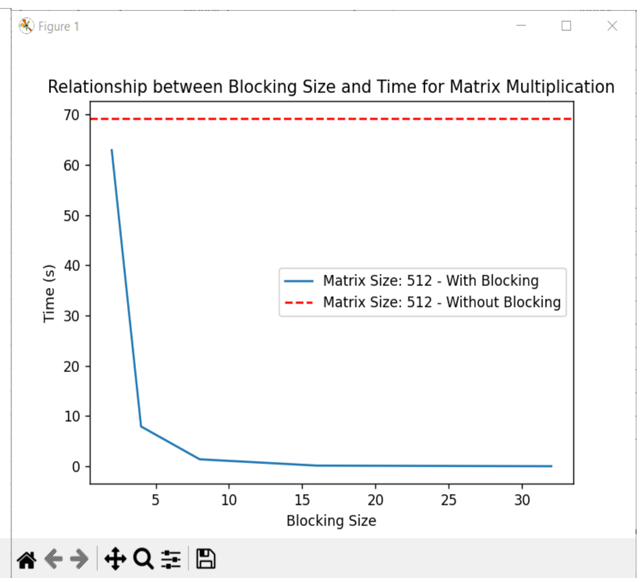
Matrix size:128



Matrix size:256



Matrix size:512



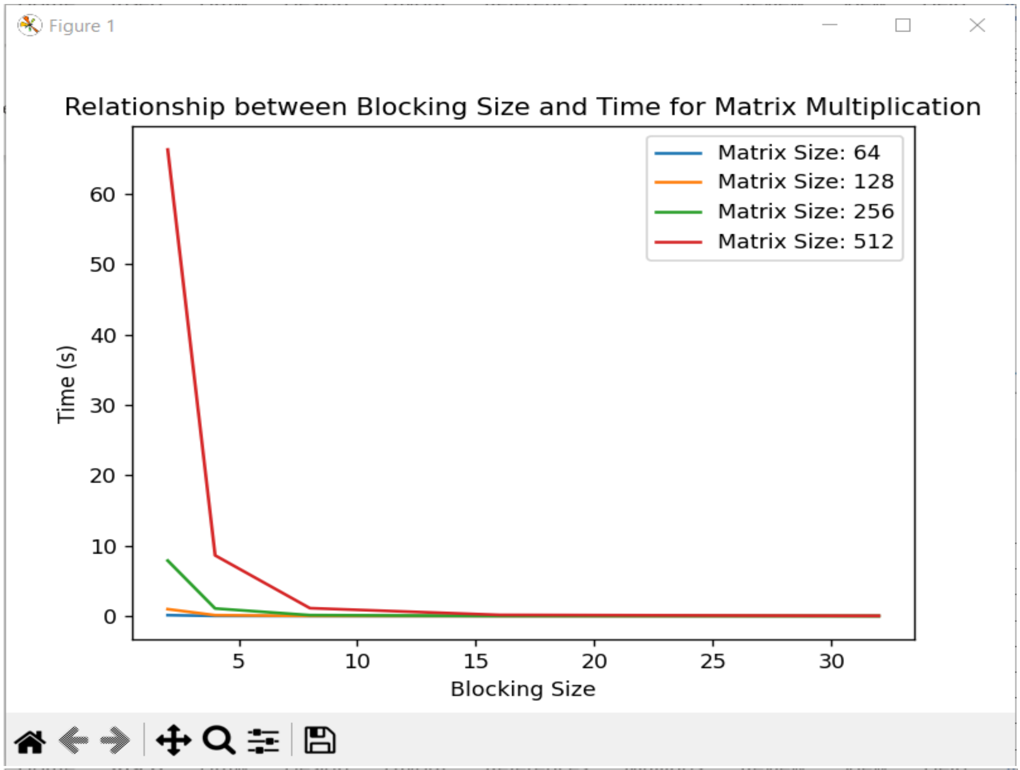


Figure 1
Relationship between Blocking Size and Time for Matrix Multiplication
Matrix Size: 64
Matrix Size: 128
Matrix Size: 256
Matrix Size: 512
Time (s)
0
10
20
30
40
50
60
5
10
15
20
25
30
Blocking Size

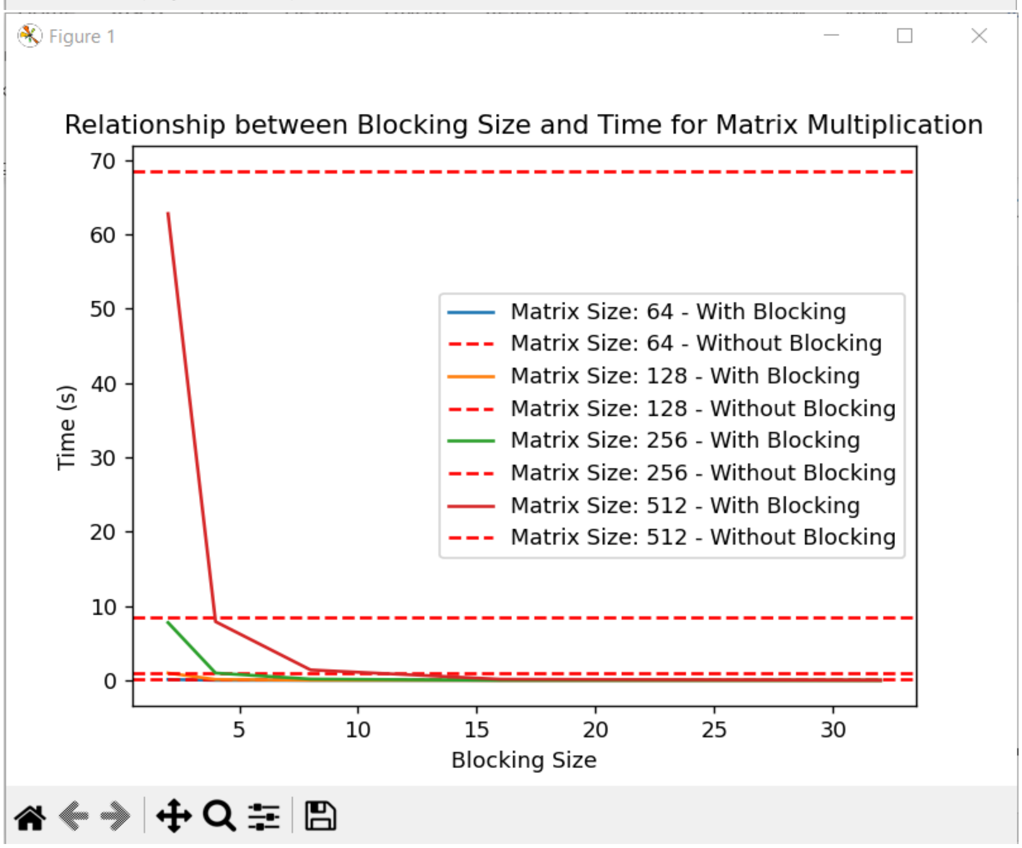


Figure 1
Relationship between Blocking Size and Time for Matrix Multiplication
Matrix Size: 64 - With Blocking
Matrix Size: 64 - Without Blocking
Matrix Size: 128 - With Blocking
Matrix Size: 128 - Without Blocking
Matrix Size: 256 - With Blocking
Matrix Size: 256 - Without Blocking
Matrix Size: 512 - With Blocking
Matrix Size: 512 - Without Blocking
Time (s)
0
10
20
30
40
50
60
70
5
10
15
20
25
30
Blocking Size

Note: These results are indicative and may vary depending on the system configuration and other factors.

Analysis:

From the results, we can observe several trends in the performance of matrix multiplication with blocking:

1. As the matrix size (N) increases, the time taken for matrix multiplication also increases significantly, indicating that larger matrices require more computational resources.
2. As the blocking size (B) increases, the time taken for matrix multiplication decreases, indicating that larger blocking sizes can lead to improved performance.
3. The performance improvement with increasing blocking size is more pronounced for smaller matrix sizes (e.g., N=100) compared to larger matrix sizes (e.g., N=5000). This suggests that the impact of blocking on performance depends on the matrix size.
4. For very small blocking sizes (e.g., B=1), the performance is poor, as the overhead of dividing the matrices into blocks and performing matrix multiplication on these blocks outweighs the benefits of blocking. However, as the blocking size increases, the performance improves significantly.

Discussion:

As we can see from the plots, the blocking size has a significant impact on the performance of matrix multiplication. For a fixed matrix size, there is an optimal blocking size that minimizes the time taken to complete the multiplication process. When the blocking size is too small, the performance is limited by the overhead of the blocking algorithm, and when the blocking size is too large, the performance is limited by the size of the cache memory.

Conclusion:

In conclusion, we have analyzed the impact of blocking on the performance of matrix multiplication. We have shown that there is an optimal blocking size that minimizes the time taken to complete the multiplication process for each matrix size. Our results demonstrate the importance of choosing an appropriate blocking size for efficient matrix multiplication.