COMP3432 – OPERATING SYSTEMS
PROJECT REPORT FILE
SPRING 2025

IŞIK UNIVERSITY
COMPUTER SCIENCE AND ENGINEERING

Student ID	**Name**	**Lastname**
20COMP1019	Arda	AKDUYGU
19SOFT1023	Mehmet	SERİN

Paragraph 1

The performance of matrix multiplication improved significantly as the thread count increased. At first, adding threads led to dramatic reductions in execution time because multiple rows of matrix A were processed simultaneously. However, after a certain point, adding more threads did not continue to provide performance benefits. The “sweet spot” for this hardware setup was found to be around 8 to 20 threads, where the execution time was minimized. Beyond this range, the benefits of adding threads plateaued or even started to degrade.

Paragraph 2

The performance degraded with too many threads because of excessive context switching and CPU resource contention. This is directly related to the number of CPU cores available on the system. While the CPU can execute a number of threads in parallel equal to its physical core count, adding more threads than cores leads to overhead from constantly switching between threads, reducing the overall performance.

Paragraph 3

Matrix B is loaded entirely into memory because it is needed in full for every row of matrix A during the multiplication. Loading Matrix A fully into memory would be memory-inefficient given its large size. By reading only the rows of matrix A that each thread needs, the program reduces memory consumption while ensuring that all necessary data for multiplication is accessible.

Paragraph 4

The performance measurements using System.currentTimeMillis() were found to be fairly consistent when running the script multiple times. However, there are small fluctuations due to background processes and system load. To ensure reliability, the script was run several times, and the results were averaged to minimize the impact of these fluctuations