

Comparison of Matrix Multiplication

Petr Kucera

Monday 18h

Open Informatics, Computer Engineering

kucerp28@fel.cvut.cz

I. METHODOLOGY

A. Warm-up

- I conducted the warm-up measurement using the methodology outlined in the assignment. Following that, I ran a benchmark test involving 30 warm-up iterations, 100 measured interactions, and 10 forks.

B. Measurements

- For measuring time performance, I employed the methods described in the task assignment. I tested benchmarks for each function with 10 forks, 30 warm-up iterations, and 100 measuring iterations. I determined the accuracy of the given algorithm using a 95% confidence interval.

C. Comparison

- I compared all implemented algorithms among themselves again using 95% confidence intervals of the ratios, as requested in the task assignment. The script for compute confidence intervals and ration can be reviewed in the project repository: <https://github.com/petrkucerak/ESW/tree/main/hw/hw04/assets>.

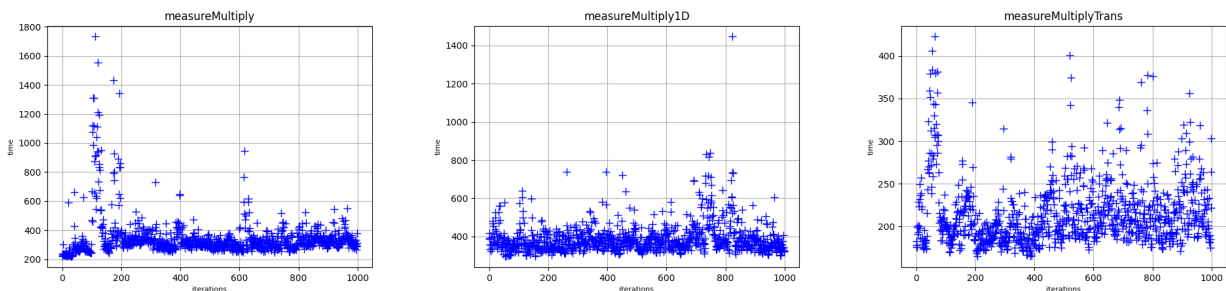
II. MACHINE SPECIFICATION

- The benchmarking runs on WSL2 on hardware equipped with 8 CPUs and 8GB of RAM.
- The code runs on OpenJDK 21.0.2 without any JVM parameters.

III. RESULTS

A. Warm-up

- The initial results were unsatisfactory. The error was likely due to running the program in WSL with insufficient prioritization and performance, thus being influenced by other factors. Nevertheless, I initiated the benchmarking program with 50 warm-up iterations and 50 forks, monitoring the terminal output to observe how the data changes.



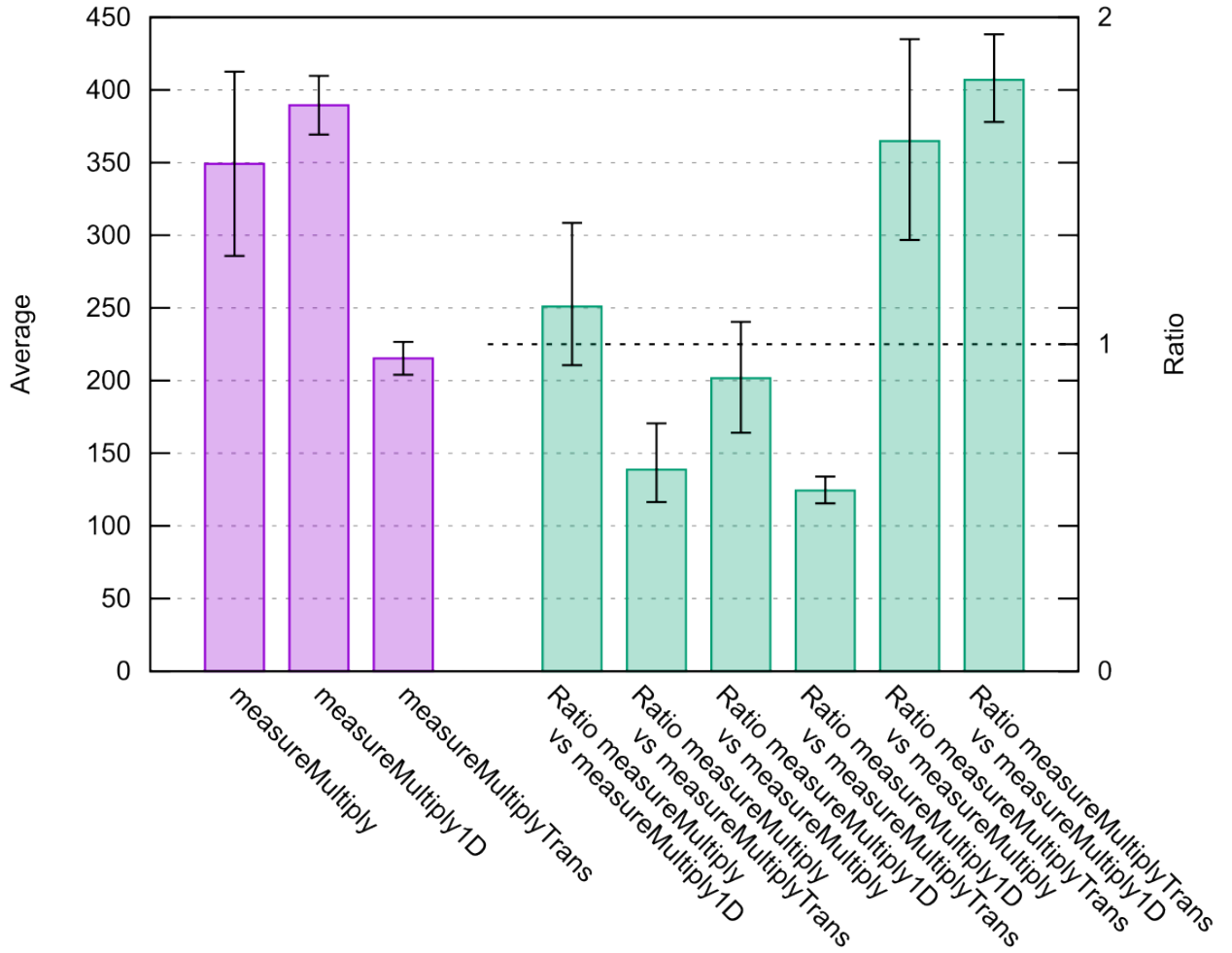
- After analyzing the data, I discovered that the program consistently starts very quickly, and the warm-up only affects the initial launch. Subsequent errors are related to other programs running concurrently with the benchmark. The data can be reviewed in the project repository: <https://github.com/petrkucerak/ESW/tree/main/hw/hw04/assets>.

B. Measurements

- Results of the time measurements including a graph with displayed confidence intervals are displayed in violet color.

C. Comparison

- Results of the comparison are displayed in green color.



IV. CONCLUSION

- In this task, I experimented with benchmarking techniques and explored the tools available for testing Java code. It was intriguing to observe the behavior of the JVM on WSL and its interaction with other processes. Additionally, the fluctuation in data quality with increasing measurement count and the distinct performance characteristics of each algorithm type were fascinating.
- The fastest algorithm was matrix multiplication with transpose, while the slowest was measured on a 1D array. This trend is likely due to the computer architecture's affinity for matrix operations, as it can leverage vector instructions, parallelization, and other powerful tools for acceleration.