Dhirubhai Ambani University (DAU)
(Formerly Dhirubhai Ambani Institute of Information and Communication Technology)

Gandhinagar, Gujarat, India

CS 301
High-Performance Computing

Lab 02: Matrix Multiplication Performance Analysis

Submitted by:
Ramani Drahsit (202301404)
Vasu Bhesandariya (202301416)

Department:
Information and Communication Technology

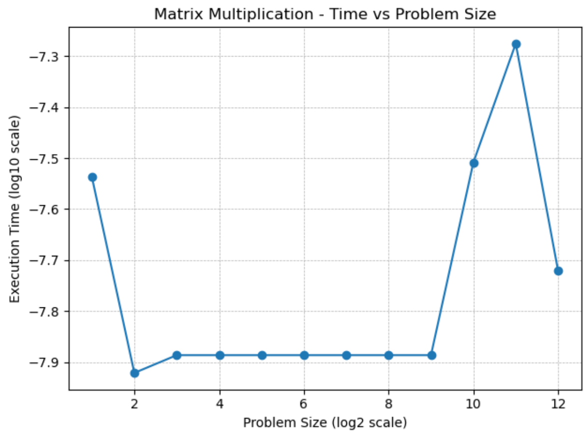
February 17, 2026

Contents

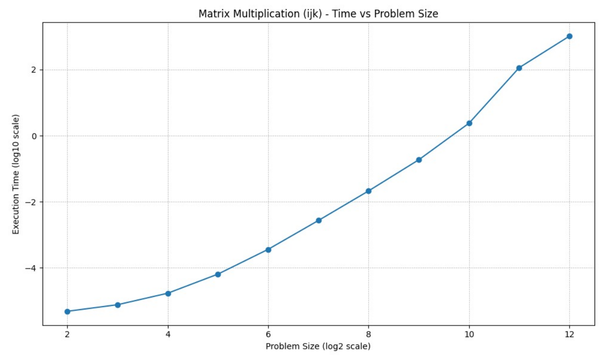
1 Matrix Multiplication: Problem Size vs Time

1.1 Standard Loop Permutations

1.1.1 ijk Order



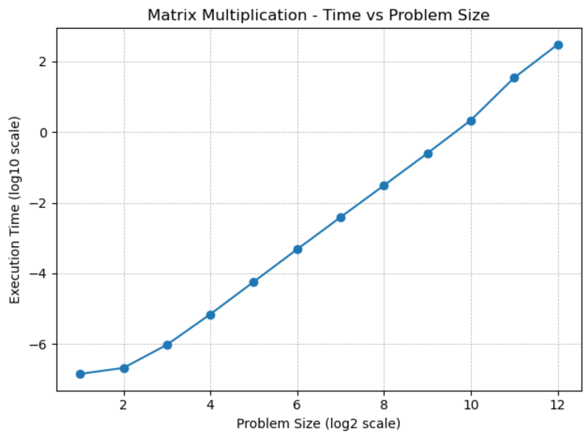
(a) Lab PC



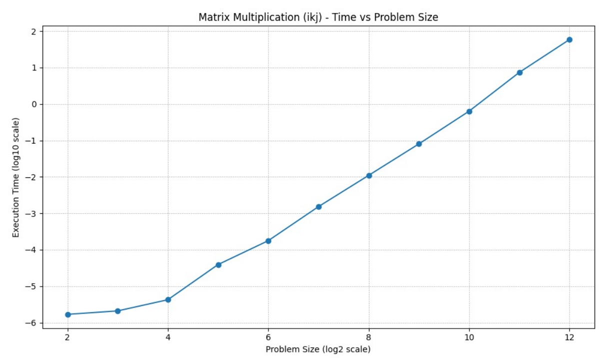
(b) HPC Cluster

Figure 1: Problem Size vs Time (ijk)

1.1.2 ikj Order



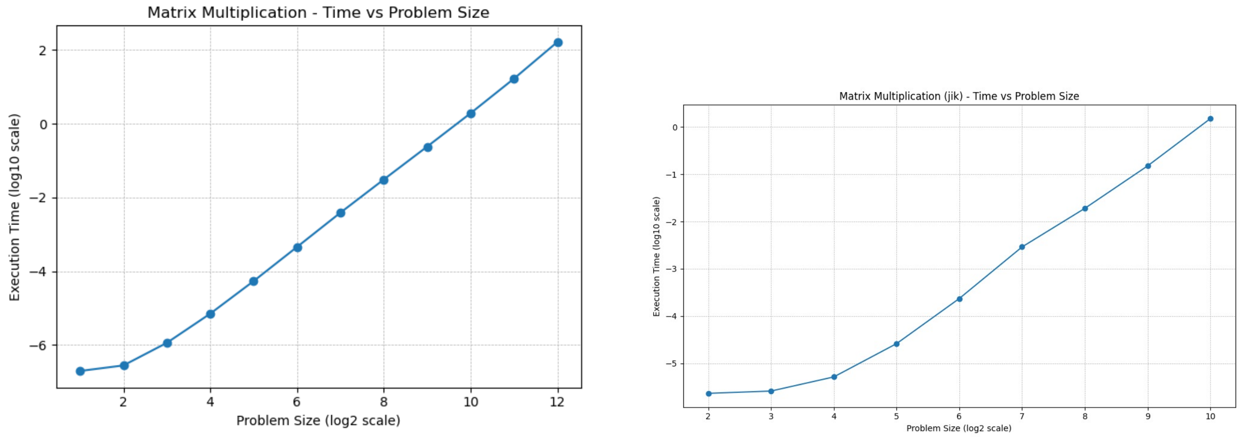
(a) Lab PC



(b) HPC Cluster

Figure 2: Problem Size vs Time (ikj)

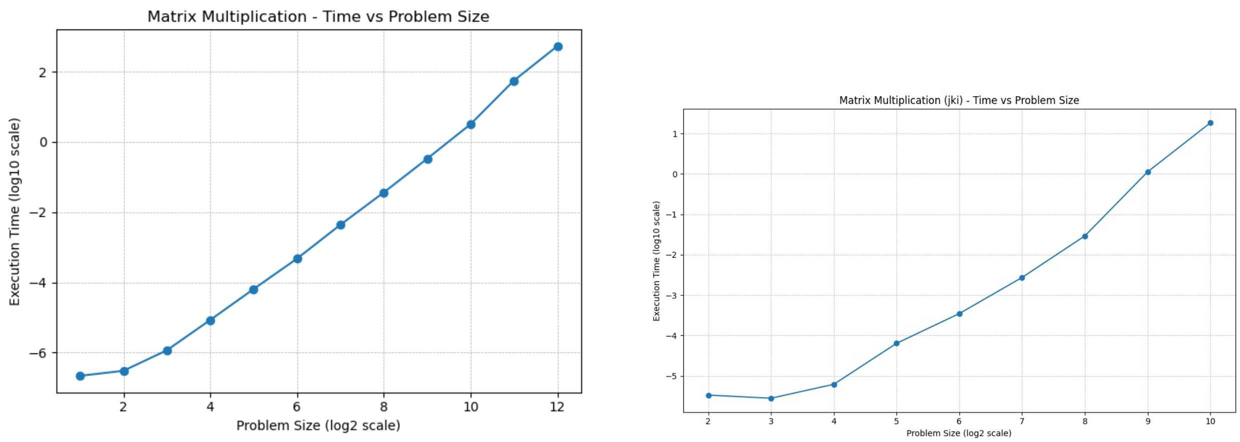
1.1.3 jik Order



(a) Lab PC (b) HPC Cluster

Figure 3: Problem Size vs Time (jik)

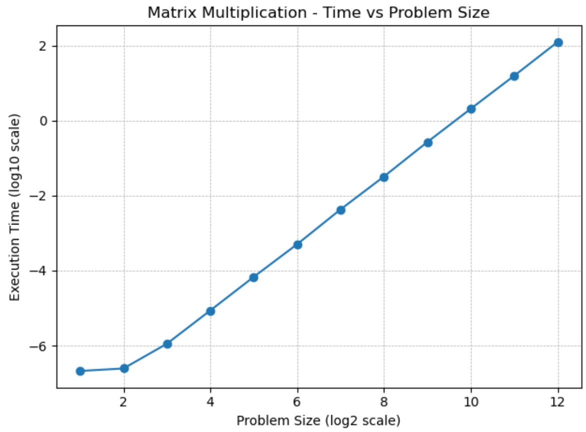
1.1.4 jki Order



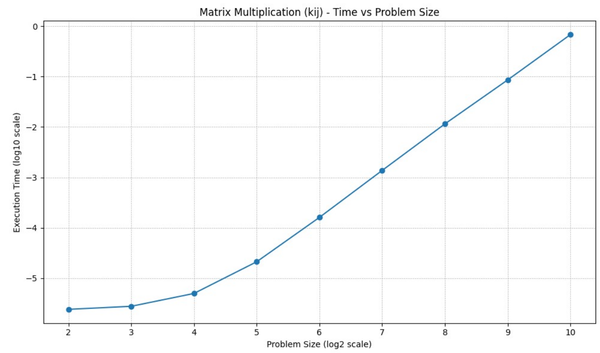
(a) Lab PC (b) HPC Cluster

Figure 4: Problem Size vs Time (jki)

1.1.5 kij Order



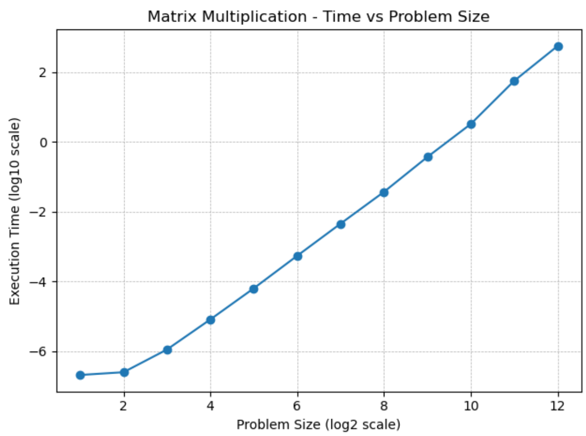
(a) Lab PC



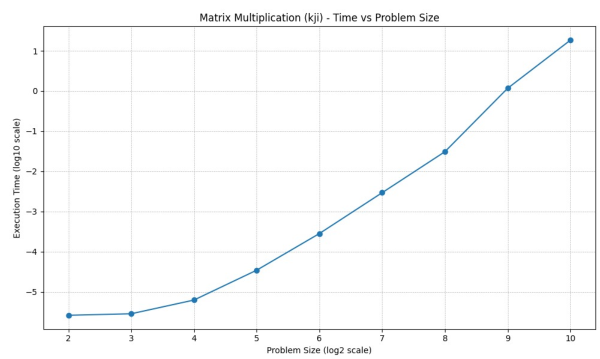
(b) HPC Cluster

Figure 5: Problem Size vs Time (kij)

1.1.6 kji Order



(a) Lab PC

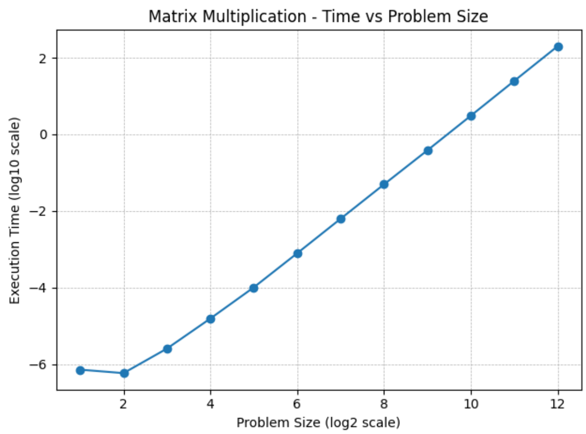


(b) HPC Cluster

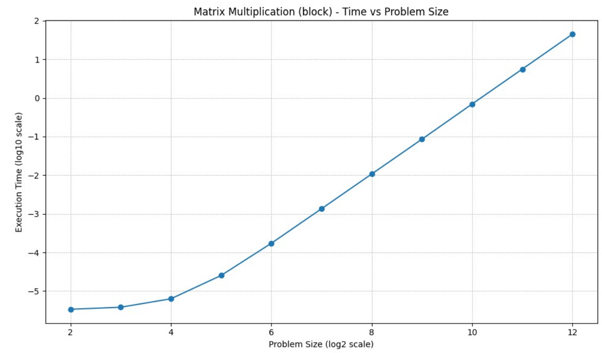
Figure 6: Problem Size vs Time (kji)

1.2 Optimized Matrix Multiplication

1.2.1 Block Matrix Multiplication



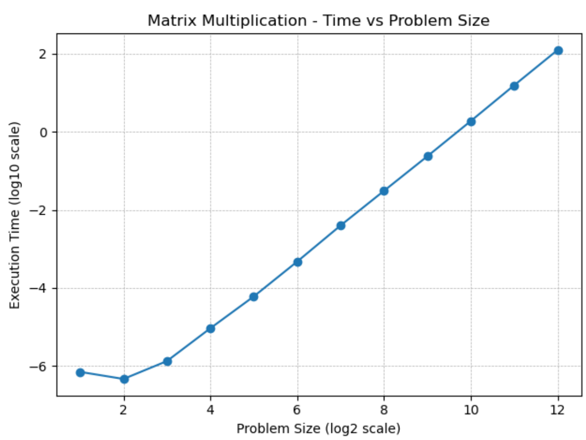
(a) Lab PC



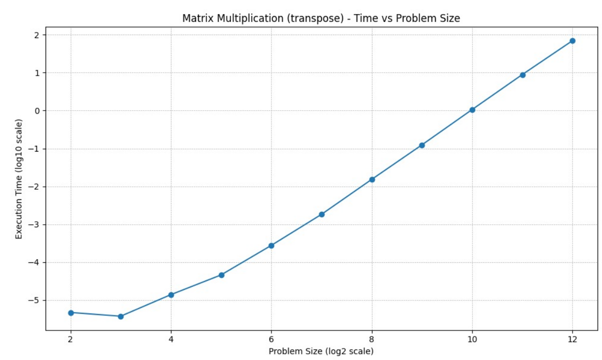
(b) HPC Cluster

Figure 7: Problem Size vs Time (Block)

1.2.2 Transpose-based Matrix Multiplication



(a) Lab PC



(b) HPC Cluster

Figure 8: Problem Size vs Time (Transpose)

2 Conclusion

This report presented a performance analysis of various matrix multiplication algorithms. We observed significant differences in execution time across different loop orderings, as well as with the transpose and block optimization strategies.

Generally, algorithms that exhibit better cache locality (e.g., **ikj**, **kij**, **block**, and **transpose**) tend to perform significantly better than those with poor cache utilization (e.g., **ijk**, **jik**, **jki**, and **kji**), especially as the problem size increases.

- **Memory Access Patterns:** The superior performance of the **block** matrix multiplication and the **transpose** method highlights the critical importance of spatial and temporal locality.
- **Hardware Impact:** While the Lab PC and HPC Cluster showed different absolute execution times, the trend remains consistent: hardware performance is heavily bottlenecked by memory bandwidth and cache misses rather than raw CPU cycles.
- **Optimization:** Effective cache usage is the primary driver for achieving high performance in numerical linear algebra.

The experimental results confirm that software-level optimizations targeting the memory hierarchy are essential for high-performance computing tasks.