PROJECT PROPOSAL

Details

Project Title	Performance Analysis of Matrix Multiplication on CPU and GPU
Created by	Matrizz Masters/ Group 37
Submission Date	26/03/2025

Introduction and Background

Matrix multiplication is a fundamental operation in computer science with applications in image processing, physics simulations, and especially machine learning. However, it is computationally intensive, especially as matrix sizes increase. This project aims to analyze how efficiently this operation can be executed on different hardware platforms: CPUs and GPUs.

CPUs and GPUs are built with different architectural philosophies — CPUs excel in task-level parallelism and general-purpose computing, while GPUs are optimized for high-throughput data-level parallelism. This project will explore how these differences impact performance in the specific context of matrix multiplication.

Preliminary Work and Experience

Preliminary research into parallel programming techniques, SIMD optimization, and CUDA programming has been completed during the week preceding this proposal. Additionally, team members have academic exposure from key computer engineering courses, notably Analysis of Algorithms, Numerical Methods in Computer Engineering, and Computer Organization, which provided foundational knowledge relevant to this project. The team has not previously undertaken a direct performance comparison between CPU and GPU architectures.

Project Description

🔹 We will:

- Implement matrix multiplication on CPU with multithreading (e.g., using OpenMP or pthreads).
- Implement the same operation on GPU using CUDA.
- Measure and compare performance for various matrix sizes.
- Analyze how each architecture handles parallelism and memory.
- Present speedup graphs and performance trade-offs.
- Conduct a literature review on existing CPU-GPU comparison methods.
- Provide detailed documentation for all implementations.
- Prepare clear, effective presentation materials.
- Rehearse and refine the in-class presentation.
- Write a comprehensive final project report with results and analysis.
  ➤ **Validate implementation correctness through result comparison across all versions.**
  ➤ **Discuss optimization challenges and limitations observed during development.**

🔹 The project will result in:

- A comprehensive comparative study supported by detailed experimental data and performance benchmarks.
- Clearly illustrated graphs and tables showing execution times, speedups, memory usage, and performance trade-offs between CPU and GPU implementations across varying matrix sizes.
- Complete, well-documented source code for both CPU (multithreading with OpenMP/pthreads) and GPU (CUDA) implementations, enabling easy reproducibility.
- Concise and effective presentation materials (slides, visuals) prepared for clear communication of key findings.
- A well-organized final report that summarizes our methods, experiments, observations, and conclusions.
- A collaborative effort that reflects teamwork in software development, performance analysis, and technical communication.
- A deeper understanding of how architectural differences affect computation speed and parallel processing.

Timeline and Task Distribution

Week	Tasks	Responsible Members
1	Research background, finalize implementation plan	Emre Safa Yalçın
2	CPU implementation (single-threaded and multi-threaded)	Doruk İlhan, Barış Özgenç, Emre Safa Yalçın
3	GPU implementation with CUDA	Elvan Teke, Emre Toy, Emre Safa Yalçın
4	Run benchmarks, gather data, analyze results	Emre Toy, Doruk İlhan
5	Report writing, presentation prep	Barış Özgenç, Elvan Teke

Tools, Libraries, and Data

- **Languages:** C/C++
- **Libraries:** OpenMP, CUDA
- **Hardware:** Personal PCs with Intel/AMD CPUs and NVIDIA GPUs
- **OS:** Linux or Windows

No external datasets are required — matrices will be generated randomly.

Project Value

- Team members will gain practical experience in parallel programming, architecture-specific optimization techniques, and performance benchmarking.
- Classmates will benefit from understanding how architectural choices impact software performance, emphasizing the importance of hardware-aware software development.
- The broader public and computing community may find this comparison useful for understanding performance trade-offs in computing platforms, relevant for hardware decisions in computationally intensive applications.

Team Members

Name	Student ID	Main Role	E-mail
Emre Safa Yalçın	150210090	Project Lead	yalcinem21@itu.edu.tr
Doruk İlhan	820220323	CPU Implementation Lead	ilhand22@itu.edu.tr
Elvan Teke	150190102	GPU Implementation Lead	teke19@itu.edu.tr
Emre Toy	150200023	Benchmarking & Data Analysis	toye20@itu.edu.tr
Barış Özgenç	150220091	Presentation & Documentation Specialist	ozgenc22@itu.edu.tr