Study of GP-GPU calculations on graphics cards

MASTER INFORMATIQUE

Enzo Ghizzardi

Supervised by
Sid Touati

Fall semester 2024-2025

Plan

Introduction

Methods and Materials

Discussion

Conclusion

History

- General-Purpose computing on Graphics Processing Unit
- Calculations exploiting the parallelism offered by graphics cards

History

- 1990s : Fixed-function GPUs for 2D/3D rendering.
- 2007 : CUDA enabled general-purpose GPU computing.
- 2009 : OpenCL offered cross-platform GPU programming.

Graphic cards today

- High parallelism : Thousands of cores.
- Specialized cores for AI (Tensor Cores, etc.).
- Applications : AI, simulations, video games.

Principles

- Compare performance between CPU (with OpenMP) and GPU (AMD with OpenCL)
- Analyze the results and explore intriguing questions related to graphic cards

Benchmarks Used

- **Matrix Multiplication :** Fundamental in linear algebra.
- **Bellman-Ford :** Shortest path in a graph algorithm with edge-level parallelism.
- **Heat Diffusion :** Simulates temperature propagation in a grid.

Methods and Materials
Methodology - What we measure ?

What we measure

- Execution time
- Performance trends for varying dataset sizes
- Comparison between CPU (OpenMP) and GPUs (OpenCL)

How we measure

- Precision timing : `clock_gettime()`
- Benchmarks executed with and without GUI (graphical interface)

- CPU : Intel i5-12600KF (10 cores, 16 threads, 3.7 GHz)
- GPUs : AMD RX 6750 XT (2560 "stream processors" ($\approx$ cores), 2150 Mhz)
- OS : Fedora Linux 41
- Tools : GCC, OpenCL library, OpenMP library

Results - Matrix Multiplication

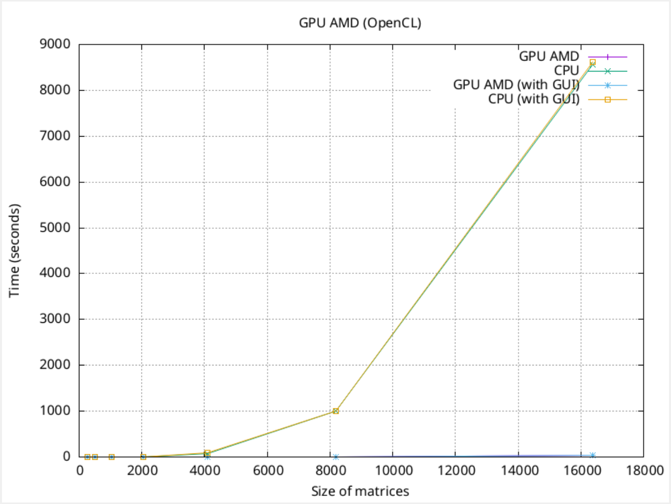

Figure – Performance comparison for Matrix Multiplication

Results - Bellman-Ford

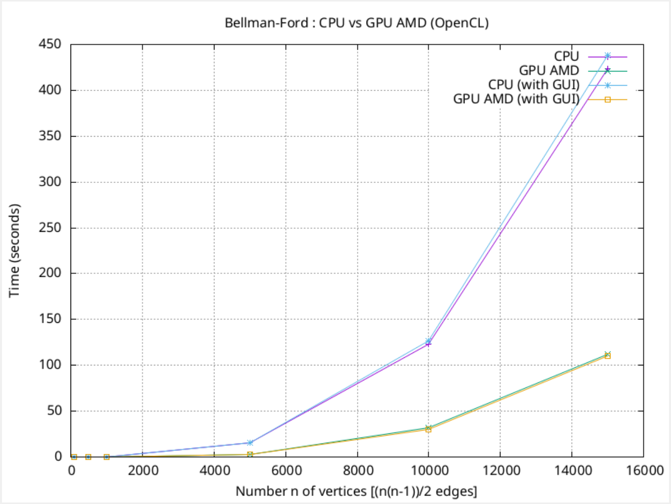

Figure – Performance comparison for Bellman-Ford

Results - Heat Diffusion

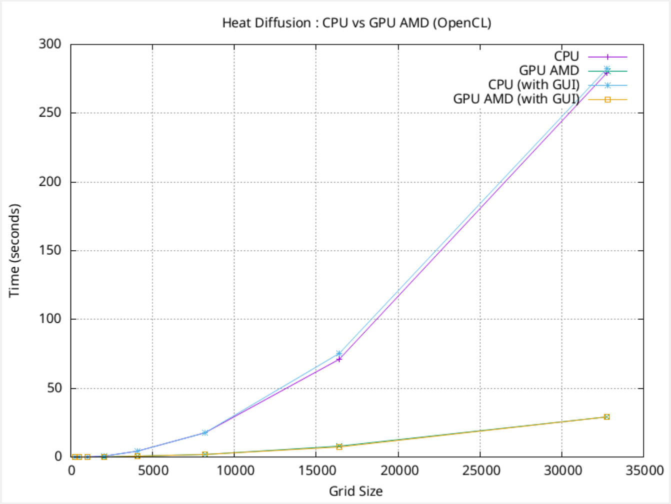

Figure – Performance comparison for heat diffusion

- Very low impact of the GUI on the measurements
- The larger the data size, the more the graphics card dominates the processor
- Processor sometimes outperforms GPUs for smaller data size

Suitable

- With high parallelism
- Without complex dependencies

Non-suitable

- Computations with complex dependencies
- With conditional instruction
- With limited parallelism (sequential)

CPUs

- Highly versatile
- Handles sequential tasks and OS management.
- Coordinates between hardware components.

GPUs

- Specialized for massively parallel computations.
- Handles graphics rendering and AI workloads.
- Allows advances in machine learning...

Conclusion

- GPUs are indispensable for modern computing
- CUDA and OpenCL offer powerful tools for exploiting GPU parallelism
- CPU and GPU play complementary roles in computation

Questions ?

Thank you for your attention !