Accelerating K-Means Clustering with OpenMP and GPU Parallelization

Final project

Master's program in Informatics Engineering and Internet of Things (IoT)

Student: Darya Martsinouskaya - 26610

Course Unit: High Performance Computing

Teacher: José Jasnau Caeiro

01 Introduction

K-means is a widely used clustering algorithm in machine learning.

It partitions datasets into k clusters by minimizing intra-cluster distances.

The computational cost grows significantly for large datasets [1].

The work explores parallelization strategies using OpenMP and GPU acceleration to improve efficiency.

02 Problem description

Sequential K-means:
The standard algorithm iteratively assigns points to clusters and updates centroids.

Challenges:
High time complexity limits scalability for large datasets (e.g., MNIST).

Goal:
Reduce execution time while maintaining accuracy.

03 Parallelization Strategies

OpenMP (Multicore CPU):
Parallelization of main loops for point assignment and centroid updates.

GPU Acceleration:
Offloading computations to GPU for enhanced parallel execution.

Speedup Calculation:
Speedup (S) = Time (Sequential) / Time (Parallel)

04 Experimental Setup

Dataset: MNIST (grayscale images, 28x28 pixels).

Metrics: Execution time, accuracy, speedup.

Implementations:
- Sequential (baseline)
- OpenMP parallelized
- GPU-accelerated

05 Experimental Setup

Dataset: MNIST (grayscale images, 28x28 pixels).

Metrics: Execution time, accuracy, speedup.

Implementations:
- Sequential (baseline)
- OpenMP parallelized
- GPU-accelerated

06 Results

k	Sequential Time (s)	OpenMP Time (s)	GPU Time (s)	Speedup OpenMP	Speedup GPU
10	838.213	100.817	1.26322	8.31	663.55
50	1495.33	48.6627	1.23969	30.73	1206.21
100	921.122	20.0596	1.2539	45.92	734.61
200	691.691	40.2045	1.24277	17.20	556.57

- GPU acceleration significantly outperforms both sequential and OpenMP implementations, achieving up to 1206.21× speedup
- OpenMP provides moderate improvements (up to 45.92× speedup) but loses efficiency at higher k values
- All implementations maintain identical clustering accuracy, confirming correctness

07 Conclusion

- Implemented sequential, OpenMP parallel, and GPU-accelerated versions of K-means
- OpenMP parallelization achieved up to 45.92× speedup, but gains decreased for large cluster sizes
- GPU acceleration provided the highest improvement, reaching 1206.21× speedup with excellent scalability
- Accuracy remained consistent across all implementations, validating parallelization correctness
- Further optimizations (e.g., CUDA, memory management) could enhance performance for larger datasets

References

1. AnjaTanovićandVukVranjković."ImplementationofparallelK-meansalgorithmfor image classification using OpenMP and MPI libraries". In: 2024 Zooming Innovation in Consumer Technologies Conference (ZINC). 2024, pp. 54–59. doi: 10.1109/ZINC61849.2024.10579351.