Raymond Zhu

CPU vs GPU Performance Analysis for Matrix Multiplication

This project investigates the comparative performance of CPUs and GPUs in matrix multiplication tasks, focusing on different sized square matrices, dense and sparse matrices, and batch multiplication. Using Python libraries such as NumPy and CuPy, computation times were measured for matrices of varying sizes and densities. The study also highlights challenges encountered during the benchmarking process, such as memory limitations on GPUs for large sparse matrices. Results indicate that GPUs outperform CPUs for large and dense matrices but face limitations in scaling smaller and more sparse operations efficiently. These findings provide insights into the optimal use cases for GPU-based computation in scientific and machine-learning applications.

Features:

CPU Matrix Multiplication

The cpu_matrix_multiplication function leverages NumPy's dot method to perform matrix multiplication on the CPU. The computation time is measured using time.perf_counter.

GPU Matrix Multiplication

GPU-based matrix multiplication is handled by gpu_matrix_multiplication, using CuPy's GPU-accelerated dot method. The computation time is measured using time.perf_counter.

CPU Dense and Sparse Matrix Multiplication

The cpu_dense_matrix_multiplication function computes dense matrix products using NumPy's dot method. The cpu_sparse_matrix_multiplication function uses SciPy's sparse matrix methods to perform operations, optimizing computations by ignoring zero entries.

GPU Dense and Sparse Matrix Multiplication

The gpu_dense_matrix_multiplication function employs CuPy's dot method to perform dense operations on the GPU. The gpu_sparse_matrix_multiplication function uses CuPy's sparse matrix library and also has error handling due to GPU's memory limitations.

Time Measurement

The computation time is measured using time.perf_counter or time.time to record computation durations.

Graph Plotting

Visualization of results is implemented using Matplotlib. Functions like plot_results and plot_batch_results generate clear graphs comparing CPU and GPU performance across matrix sizes, sparsities, and batch sizes.

Adjustable Parameters

Key parameters such as matrix size, sparsity, and batch size.

Usage:
Compile and run the program using python:
python Final_Project.py

Hardware Specifications:
Laptop Model: Lenovo Thinkpad (4 years old)
CPU: Intel(R) Core(TM) i7-10850H @ 2.70GHz
GPU: NVIDIA GeForce GTX 1650 Ti with Max-Q Design

Benchmarked performance across:

Matrix sizes: 32x32 to 1024x1024
Sparsity levels: 1.0, 0.1, 0.01
Batch sizes: 1 to 5000

Results:

See Github