

Midterm Project Report - CSE440 Parallel Programming
Akdeniz University, Faculty of Engineering, Computer Engineering
Department
2023-2024 Spring Semester

Objective This project aims to implement parallel matrix multiplication using Microsoft MPI (MSMPI) for large matrices of size 5000x5000. These matrices are loaded from external binary files, and the primary goal is to showcase proficiency in parallel computing techniques, particularly applied to matrix operations.

Methodology The approach taken involves the following steps:

1. **Implementation of Parallel Matrix Multiplication:** Utilizing MSMPI to distribute the matrix multiplication workload across multiple processes.
2. **Matrix Loading:** Matrices A and B are loaded from binary files named `matrixA.bin` and `matrixB.bin` using a Python script.
3. **Optimization Techniques:** The code was optimized for performance by applying loop restructuring and **minimizing data communication** between processes. A significant optimization was the use of **row-major order** access to improve memory locality and cache utilization.
4. **Execution Time Measurement:** The `MPI_Wtime()` function was employed to measure the execution time accurately.
5. **Performance Evaluation:** The system's performance was assessed by calculating speedup and efficiency, comparing the parallel execution against a theoretical single-processor execution.

Execution Instructions

- **Compilation:**

```
mpicc main.c -Ofast -o main
```

- **Running the Application:**

```
mpirun --oversubscribe -np 8 ./main 5000 matrixA.bin matrixB.bin
```

- **Generating Matrices:**

```
python generate_matrices.py -s 5000 -f matrixA.bin  
python generate_matrices.py -s 5000 -f matrixB.bin
```

Benchmarking The performance of the matrix multiplication was benchmarked with varying numbers of processes. The benchmark script **benchmark_script.py** was utilized with parameters adjusted to measure the impact of different process counts on the computation time.

```
python benchmark_script.py --size 50 --processes 1 2 5 10 25
```

Results

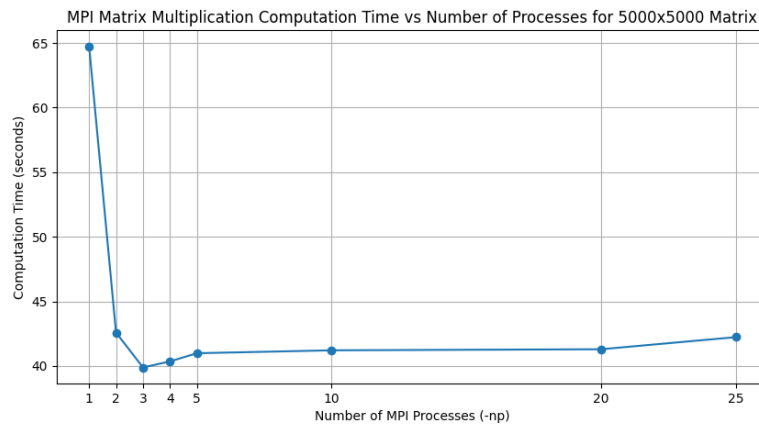


Figure 1: benchmark-mpi

The following graph illustrates the MPI Matrix Multiplication Computation Time vs. Number of Processes for a 5000x5000 Matrix:

Observations:

- A significant decrease in computation time was observed as the number of processes increased from 1 to 5.
- Beyond 5 processes, the decrease in time became more gradual, indicating diminishing returns on adding more processes since I have 4 core CPU it makes sense.