Project 4 Report for Probelm 4.1

Zhu Liang

November 13, 2023

1 Project Description

The primary objective of this project is the implementation of the parallel Strassen algorithm for matrix multiplication across three levels (1, 2, and 3), focusing on minimizing memory usage and communication costs.

Our approach utilizes a hierarchical multi master-slave tree (or leader-worker tree) structure. Each layer comprises a single leader and seven workers, where workers can serve as leaders for the subsequent layer, facilitating the next level of the Strassen algorithm. This design allows the leader to distribute specific submatrices to its workers, who later return their computed results. By limiting data exchange to only essential information between leaders and workers, our implementation effectively reduces communication overhead and optimizes memory consumption.

2 Algorithm Description

The general idea of the parallel Strassen algorithm is to divide-and-conquer the matrix multiplication problem into subproblems. The subproblems are then distributed to different processors to compute. The results are then collected and combined to form the final result. This process can be repeated recursively until the

Algorithm 1 Strassen Multiply Parallel

```
procedure StrassenMultiplyParallel($A, B, N, max\_level$)
    for $level \leftarrow 1$ to $max\_level$ do
        DistributeData($level$)
    end for
    $C \leftarrow$ MatrixMultiply($A, B$)
    for $level \leftarrow max\_level$ down to 1 do
        CollectResults($level$)
    end for
    return $C$ on $ROOT$
end procedure
```

This description is intentionally streamlined, omitting many details and supporting functions for brevity. More details can be gleaned from the source code file `functions.c`.

3 Results

3.1 Performance

Notice that the level 1, 2, 3 of the parallel Strassen algorithm requires 7, 49, 343 processors respectively to fully utilize the parallelism. We will run the parallel Strassen algorithm with 7, 49, 343 processors respectively.

	Parallel Strassen Algorithm		
(in seconds)	level 1 (cores = 7)	level 2 (cores = 49)	level 3 (cores = 343)
$N = 2^8$	0.029430	0.010317	0.040137
$N = 2^{10}$	1.757423	0.284435	0.274502
$N = 2^{12}$	63.811983	9.049598	2.902417

Table 1: Performance of the Parallel Strassen Algorithm at different levels.

3.2 Speedup

Speedup is a metric that quantifies the performance improvement of a parallel algorithm compared to its sequential counterpart. The orginal speedup S for P processors for size N problem is defined as:

$$S(P, N) = \frac{T(1, N)}{T(P, N)}$$

where $T(1, N)$ is the execution time of the sequential algorithm and $T(P, N)$ is the execution time of the parallel algorithm using P processors.

However, in our case, the sequential algorithm on single core is not implemented. Instead, we use the performance of the parallel algorithm with $P = 7$ as the baseline. The updated speedup S is defined as:

$$S(P, N) = 7 \times \frac{T(7, N)}{T(P, N)} \tag{1}$$

Using the data from Table 1, we can compute the speedup for matrix multiplication for different matrix sizes N and varying number of processors P. The speedup values for each N (under level 2 and level 3) are plotted in the following Figure 1.

4 Analysis

File Notes

The source code of the program is in the `project4` folder. For more details, please refer to the `README.md` file in the folder.

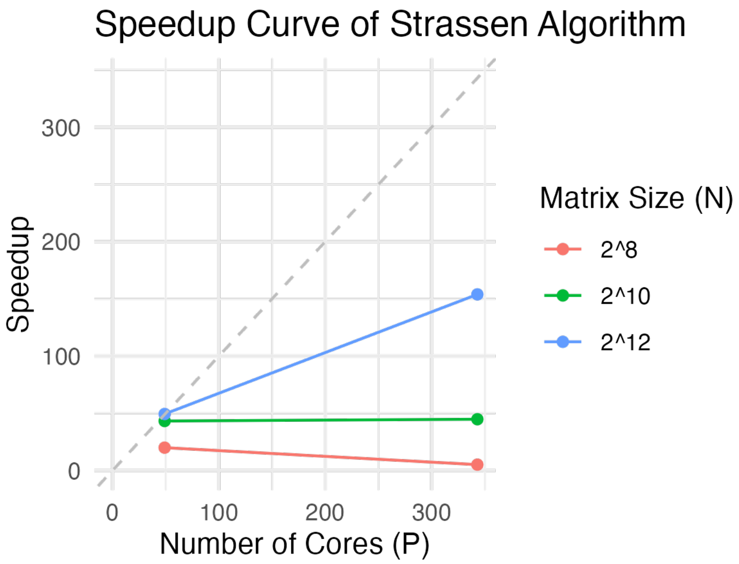

Figure 1: Speedup Curve for Matrix Sizes N