Parallelizing Strassen's Algorithm

Project 4.1

Zhu Liang

Introduction

Strassen's: divide multiplication into [7 sub-multiplications](#).

- It improves efficiency with a potential *recursive* approach.
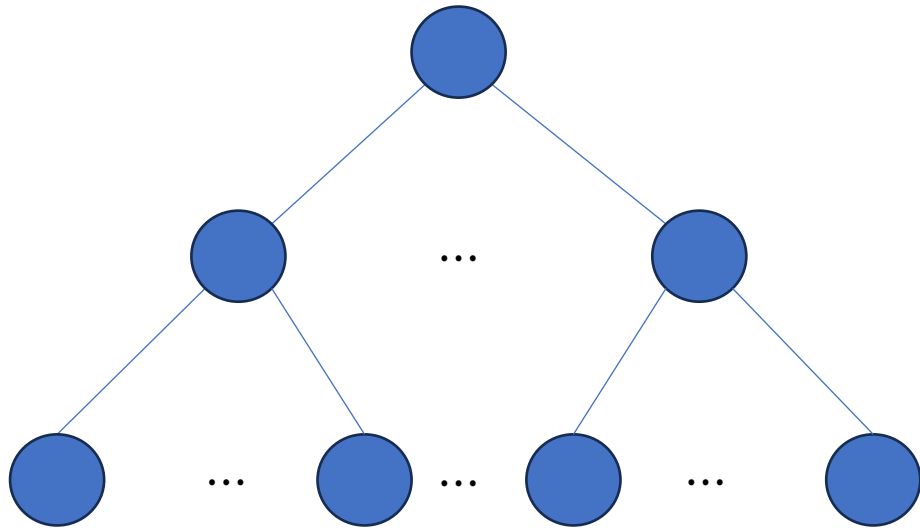
This project:

- Implements Strassen's algorithm in parallel to leverage efficiency.

- Draws on insights from *Chou, C. C., Deng, Y. F., Li, G., & Wang, Y. (1995),* applying a *recursive* strategy for efficient data distribution and collection.

Table of Content

- Parallel Implementation
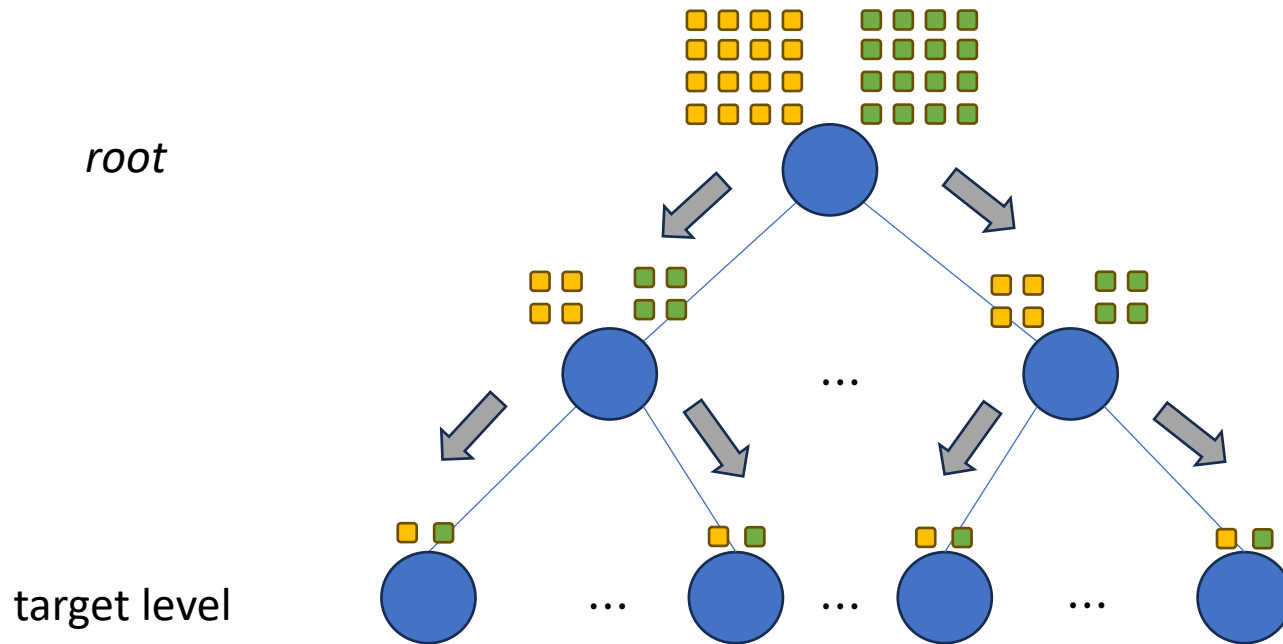
- Result & Analysis

- Summary

Parallelization

Multi Level Master-Slave
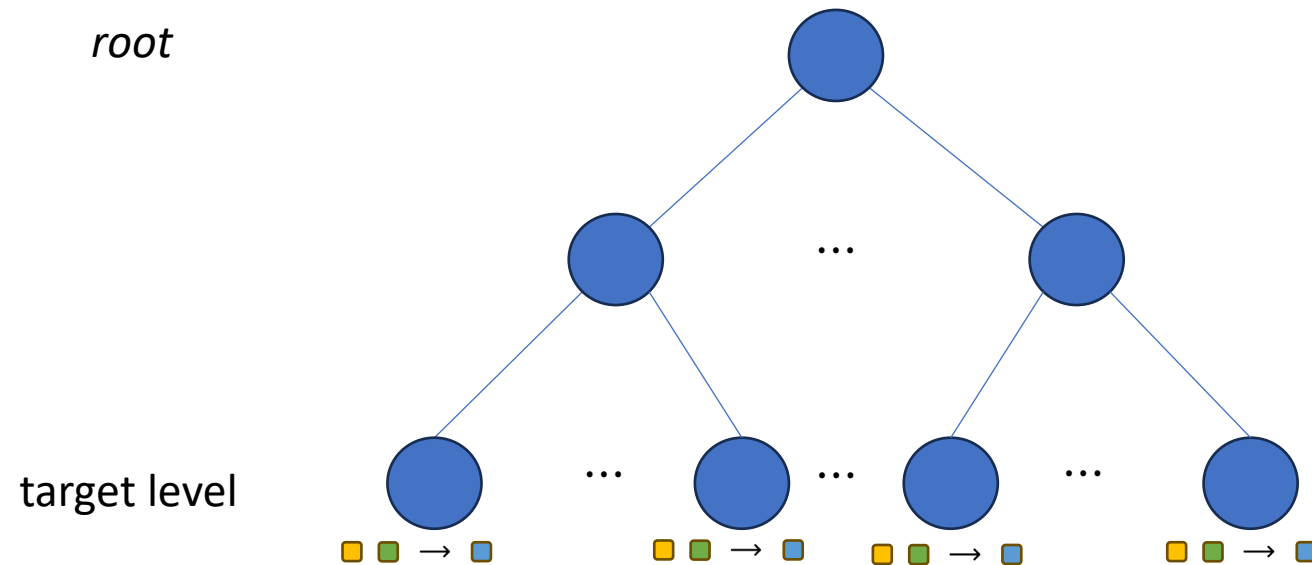(Leader-Worker) network

Three Steps:

1. Distribute Data

2. Local Multiplication

3. Collect Result

1. Distribute Data
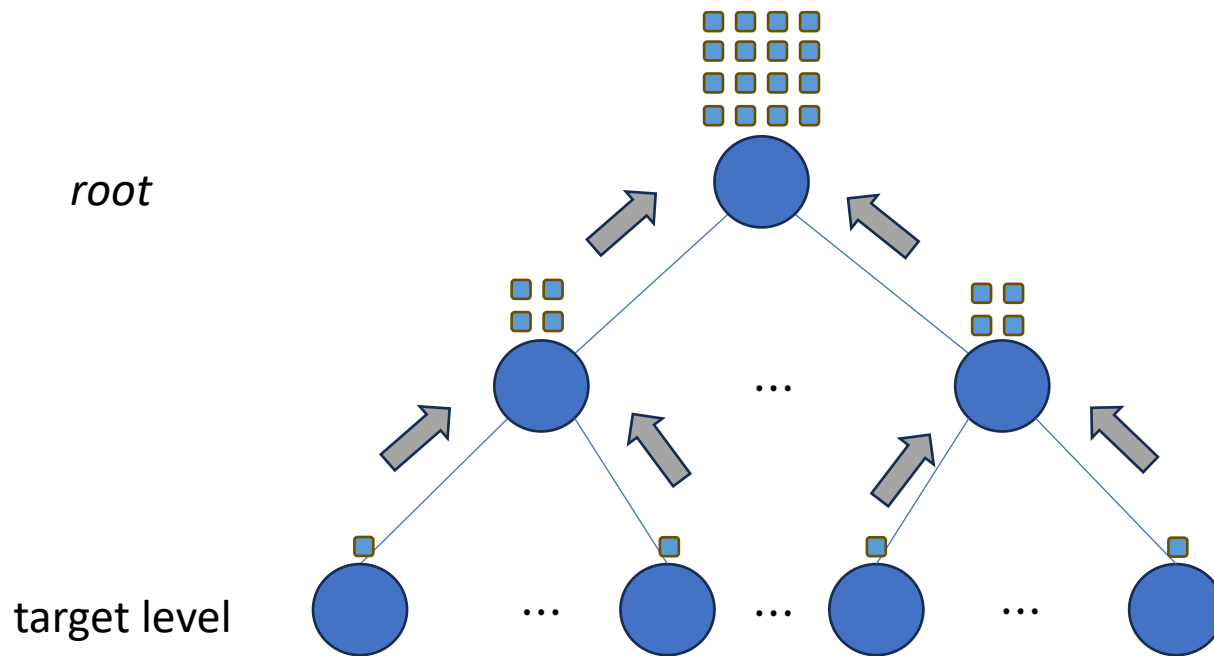

root

target level

- **Leader** [assembles]{.underline} M_i^A and M_i^B and sends to its workers.

- **Workers** receive its M_i^A and M_i^B.

- Repeat until reach the target level.

2. Local Multiplication

- **All** processors run its local multiplication

3. Collect Result

- **Workers** send results M_i.

- **Leader** receives and assembles into result C.

- Repeat until reach the *root.*

Static Leader-Worker Tree

- Ranks determine a static Leader-Worker relationship.
- Reuse leader as a worker.

Special Case: *root* processor

- a dual role as both sender and receiver, which could lead to deadlock.
- MPI data transfer is not necessary, direct memory copying.

Static Relationship Tree by Rank

Leader Rank	Worker Ranks
0	[0, 1, 2, 3, 4, 5, 6]
1	[7, 8, 9, 10, 11, 12, 13]
2	[14, 15, 16, 17, 18, 19, 20]
3	[21, 22, 23, 24, 25, 26, 27]
4	[28, 29, 30, 31, 32, 33, 34]
5	[35, 36, 37, 38, 39, 40, 41]
6	[42, 43, 44, 45, 46, 47, 48]
...	...

- Level-1 leader: $\{0\}$
- Level-2 leaders: $\{0, \rightarrow 6\}$
- Level-3 leaders: $\{0, \rightarrow 48\}$

Results

(in seconds)	Parallel Strassen Algorithm		
	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

Results

Speedup Curve of Strassen Algorithm
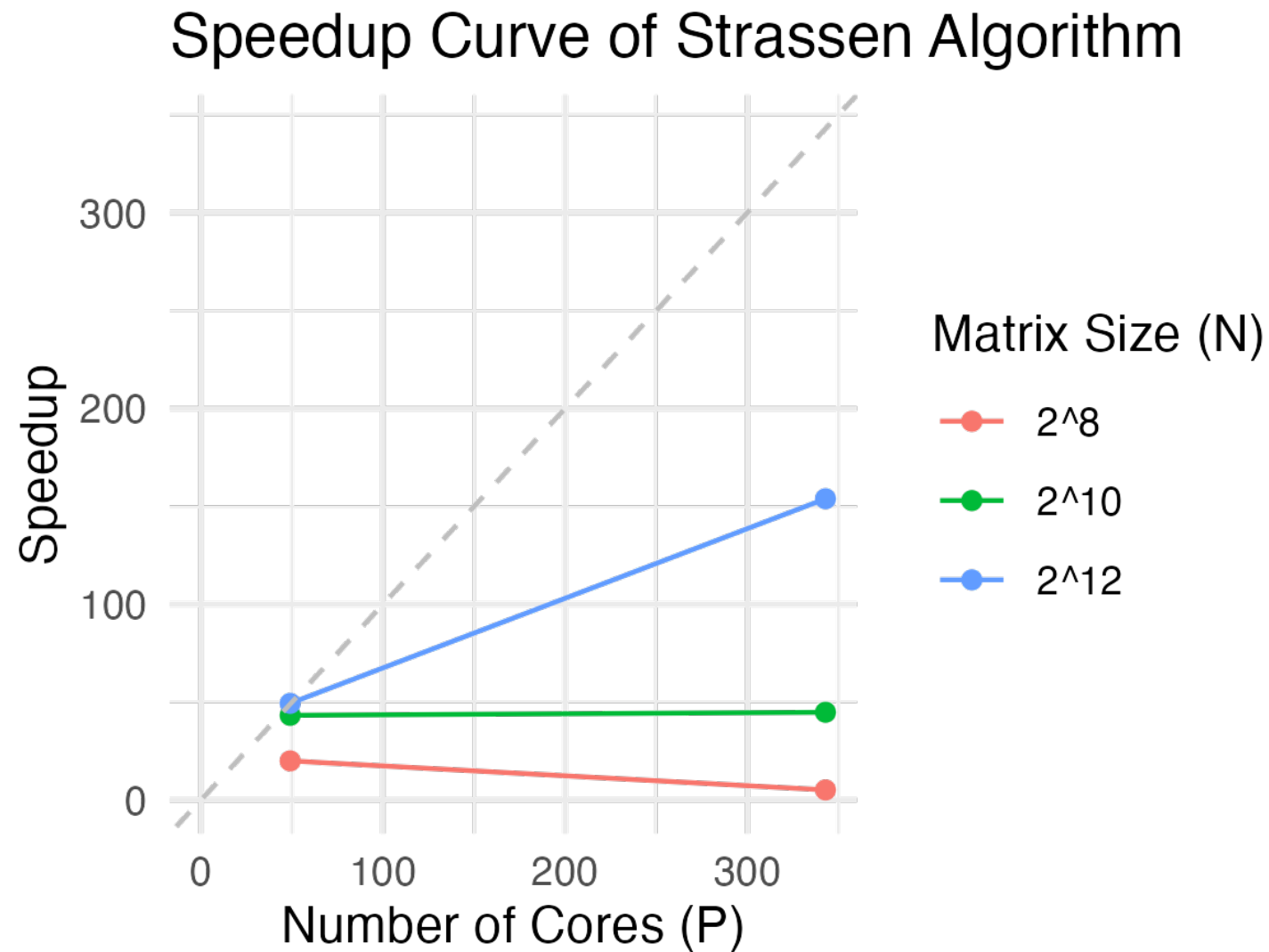
Analysis

- **Trends**

  - Speedup improves when matrices become larger.

- **Computation to Communication Ratio** :

  - improves as matrix size N increases, contributing to a more notable speedup.

- **Future Optimization**

  - Improve load distribution among processors

Summary

- Successfully implement the 3 levels via a multi-level master-slave network.

- Build a static tree to determine the leader-worker relationship.

- Only necessary data is communicated.

- Scalable to higher levels of Strassen's.

Q & A

Thanks for your attention!

Appendix: Strassen's Sub-Matrices

M_i^A	M_i^B	M_i	Assignment
$M_1^A = (A_{11} + A_{22})$	$M_1^B = (B_{11} + B_{22})$	$M_1 = M_1^A \times M_1^B$	$\rightarrow$ Worker 1
$M_2^A = (A_{21} + A_{22})$	$M_2^B = B_{11}$	$M_2 = M_2^A \times M_2^B$	$\rightarrow$ Worker 2
$M_3^A = A_{11}$	$M_3^B = (B_{12} - B_{22})$	$M_3 = M_3^A \times M_3^B$	$\rightarrow$ Worker 3
$M_4^A = A_{22}$	$M_4^B = (B_{21} - B_{11})$	$M_4 = M_4^A \times M_4^B$	$\rightarrow$ Worker 4
$M_5^A = (A_{11} + A_{12})$	$M_5^B = B_{22}$	$M_5 = M_5^A \times M_5^B$	$\rightarrow$ Worker 5
$M_6^A = (A_{21} - A_{11})$	$M_6^B = (B_{11} + B_{12})$	$M_6 = M_6^A \times M_6^B$	$\rightarrow$ Worker 6
$M_7^A = (A_{12} - A_{22})$	$M_7^B = (B_{21} + B_{22})$	$M_7 = M_7^A \times M_7^B$	$\rightarrow$ Worker 7