University at Buffalo
The State University of New York

PARALLEL COMPUTING

MATRIX MULTIPLICATION

USING MPI

INSTRUCTOR : DR. RUSS MILLER

PRESENTED BY : NEEL DUNGARANI

CONTENT

- Problem definition
- Sequential algorithm
- Parallel Model
- Performance readings
- Observation & Conclusion

PROBLEM DEFINITION

- Given a matrix A(n x m) n rows and m columns, where each of its elements is denoted Aij with 1 ≤ i ≤ n and 1 ≤ j ≤ m, and a matrix B(m × p) of m rows and p columns, where each of its elements is denoted Bij with 1 ≤ i ≤ m, and 1 ≤ j ≤ p, the matrix C resulting from the operation of multiplication of matrices A and B, C = A × B, is such that each of its elements is denoted Cij with 1 ≤ i ≤ n and 1 ≤ j ≤ p, and is calculated follows

$$C_{r,c} = AB_{r,c} = \sum_{i=1}^{n} A_{r,i} * B_{i,c}$$

SEQUENTIAL IMPLEMENTATION

The sequential algorithm for $C = A \times B$

$$C_{ij} = 0$$
$$for(i = 0; i < n; i++)$$
$$for(j = 0; j < n; j++)$$
$$for(k = 0; k < n; k++)$$
$$C_{ij} = C_{ij} + A_{ik} \times B_{kj}$$

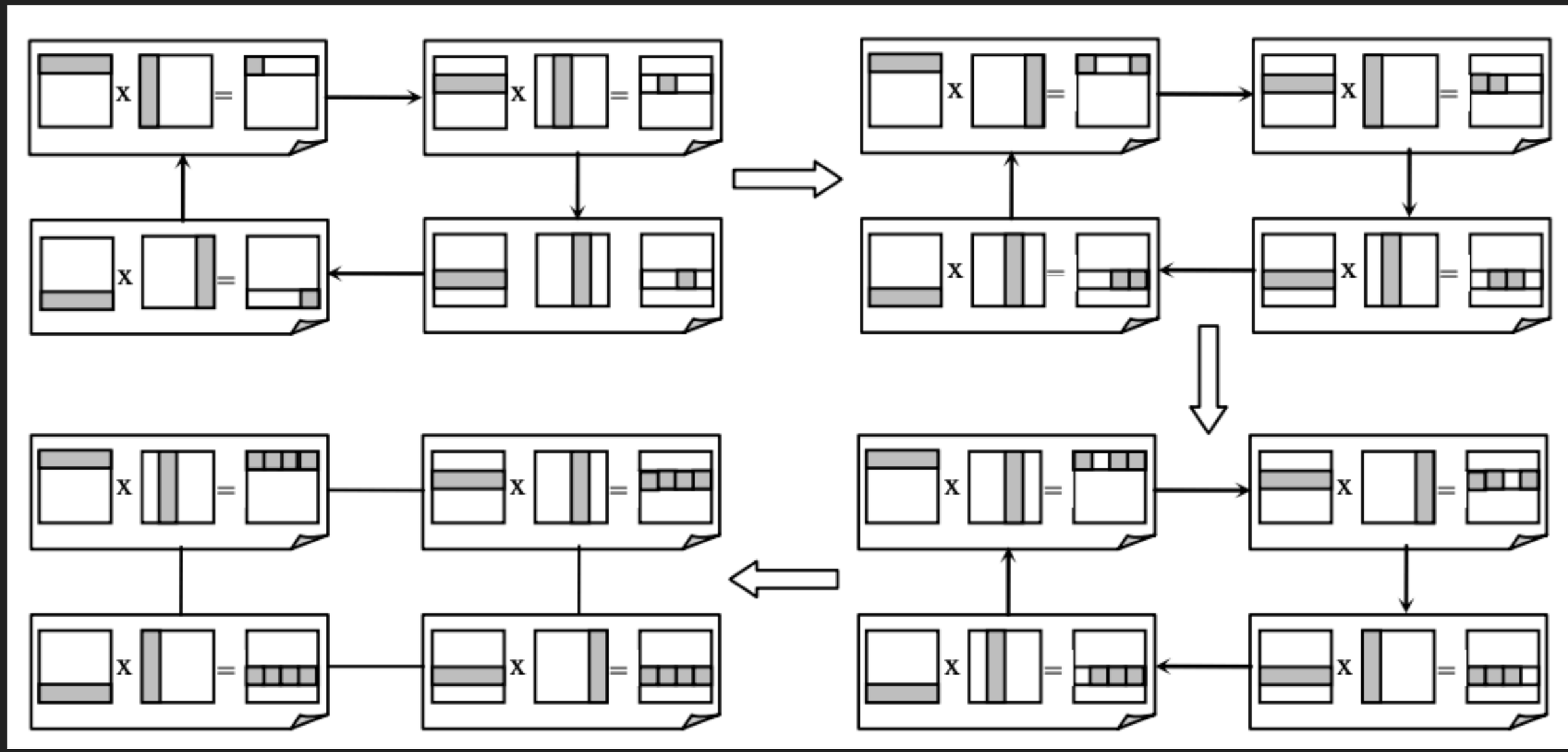
Remark: The algorithm performs n^3 scalar multiplications

PARALLEL MATRIX–MATRIX MULTIPLICATION IN CASE OF BLOCK–STRIPED DATA DECOMPOSITION

PARALLEL APPROACH

- Data decomposition : Partition matrices in such a way that each processor holds n/p number of rows from first matrix and m/p number of columns from second matrix. P is the number of processors
- This is an iterative approach, at each iteration for each processor the scalar products of rows and columns are computed and corresponding elements of resultant matrix is obtained
- After completing all computations the columns of second matrix are transmitted so that processor will have new columns of second matrix and new corresponding elements of the resultant matrix could be calculated
- Iterations are performed until each processor does computation with each column set
- This transmission of columns should be done sequentially and for that we can use ring topology
- We can also perform the same algorithm by taking the row sets of both matrices with little change in generating resultant matrix

PARALLEL APPROACH



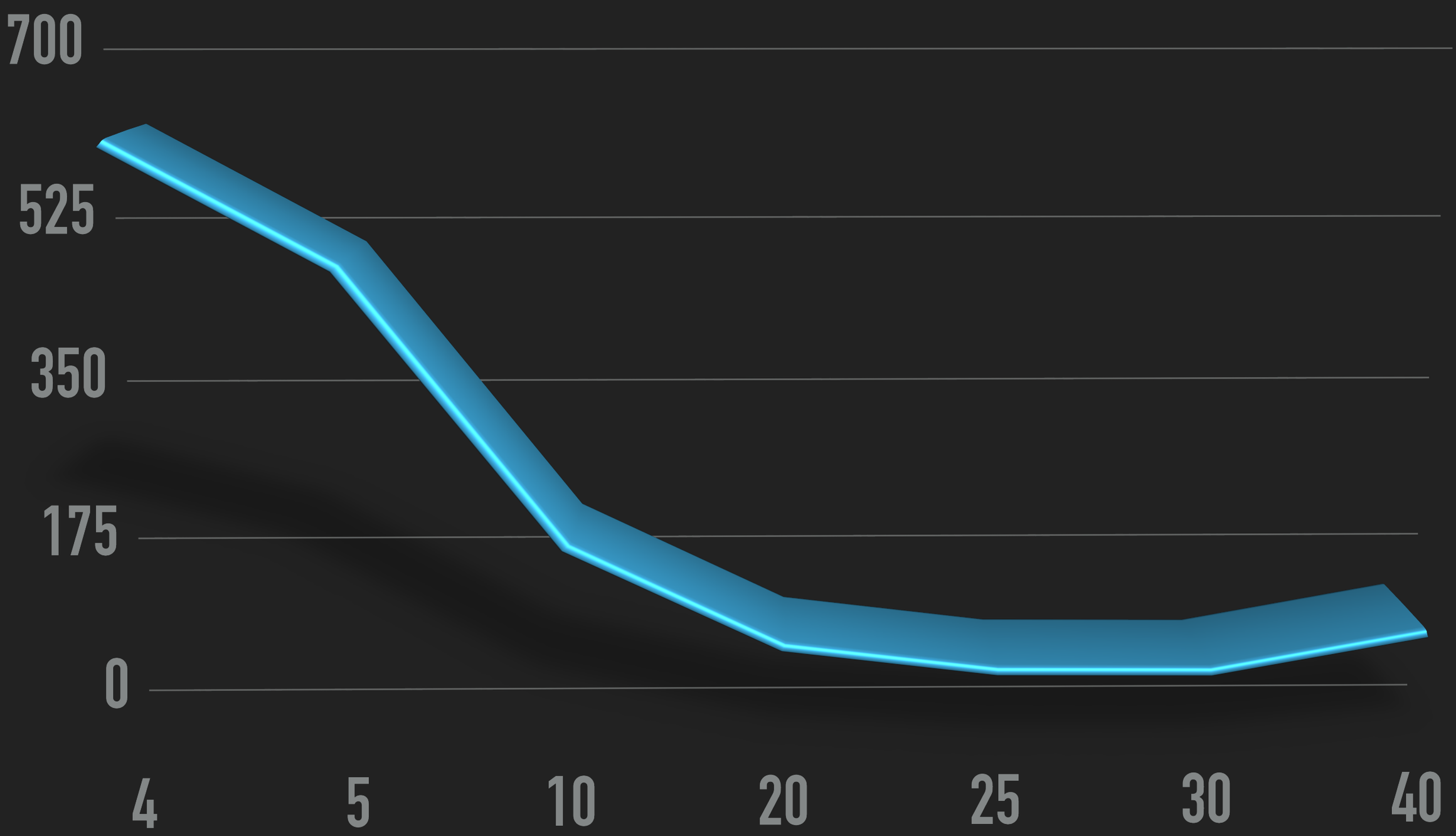
PERFORMANCE READINGS

AMDAHL'S SPEED-UP

	200	400	600	800	1000	1200
2	*1.106657791*	*7.973729563*	*26.76074004*	*64.41182442*	*124.0623418*	*213.9660168*
4	*0.6077876091*	*4.135692215*	*13.85184846*	*33.00653934*	*64.46088333*	*110.8799999*
5	*0.5182656765*	*3.364183807*	*11.09117699*	*26.85750055*	*52.27072926*	*88.86877859*
10	*0.3023694515*	*1.948149443*	*5.960002613*	*13.88259749*	*26.88257751*	*46.55224848*
20	*0.3562928677*	*1.353176594*	*4.222823381*	*8.31398592*	*15.82262983*	*26.83699918*
25	*0.441633749*	*1.381597185*	*3.886276817*	*8.772104883*	*14.54656553*	*24.24991018*
40	*2.86203618*	*3.995437574*	*7.586402416*	*13.83137565*	*23.78692217*	*35.19278491*

AMDAHL'S SPEED-UP

FOR 2000X2000 MATRICES



OBSERVATIONS

- Cannot plot Gustafson's plot
- Works fine only for smaller data : Data decomposition
- You can multiply non-square matrices
- Too much communication overhead
- Number of columns in each processor must be same

THANK YOU...!!

NEEL DUNGARANI