PARALLEL MATRIX MULTIPLICATION

Mojitha Kurup

CSE 708

University at Buffalo
The Graduate School

Problem Statement

Let $A = [aij]$ and $B = [bij]$ be n × n matrices. Compute C = AB

Sequential approach of matrix multiplication

- θ(n^3) time complexity
- Drastic change in run time for large size matrices
- Pseudo code:

```
procedure seq_matrix_multiplication (A, B, C)
Begin
   for i=0 to n-1 do
      for j=0 to n-1 do
             C[I, j] = 0
             for k=0 to n-1 do
                C[I, j] += A[i. k] X B[k, j];
             end for;
end seq_matrix_multiplication
```

Parallel approach of matrix multiplication

Cannon's Algorithm

- We partition input matrices into P square blocks (P is the number of processors available)
- Mesh of √p x √p will be created using Cartesian topology where P_{ij} store A_{ij} and B_{ij} which will compute C ij.
- Each block will be sent to each process determined by its owner
- Wrap-around shifts will be perfomed
- Total no of steps required will be √P
- Data per processor will be $(n/\sqrt{p})\times(n/\sqrt{p})$
- Assume P to be perfect square and n as a multiple of √p

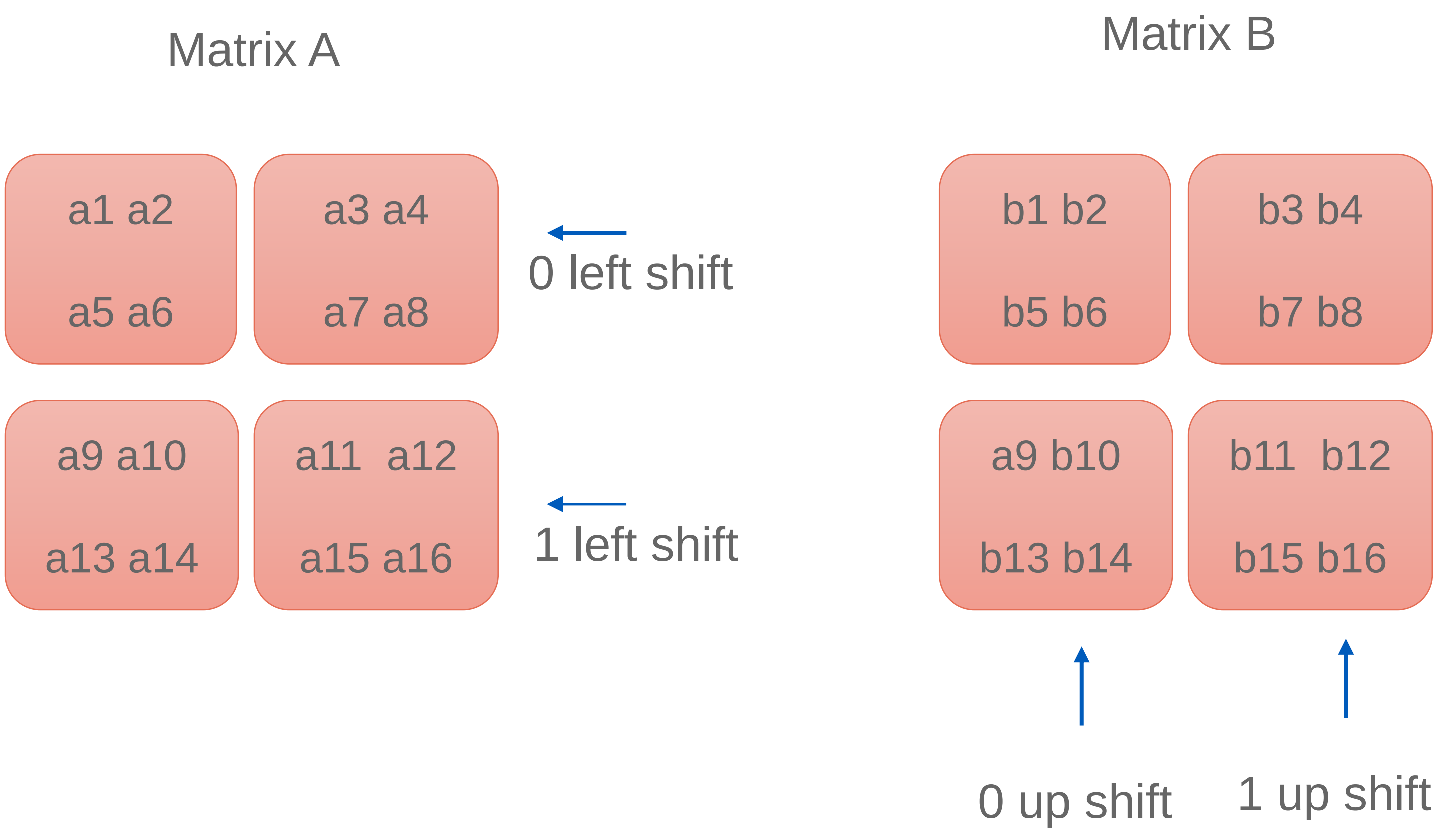
Initial Alignment:

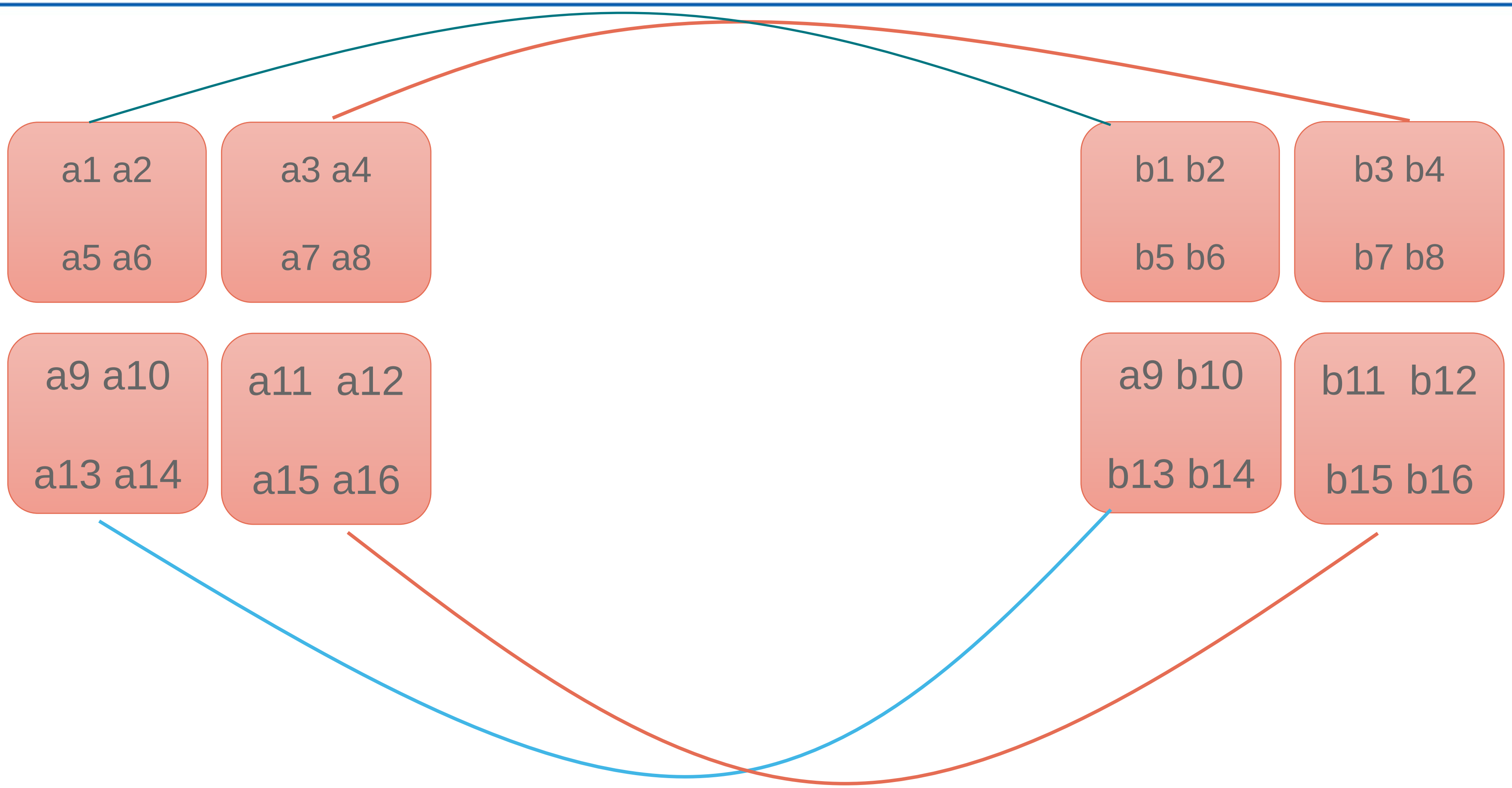
for i,j :=0 to $p - 1$ do

Send block Ai,j to process $i, j - i + p \ mod \ p$
and block Bi,j to process $i - j + p \ mod \ p,j$;
endfor;

Process Pi,j multiply received submatrices together and add the result to Ci,j ;

In this step, the send operation is to: shift Ai,j to the left (with wraparound) by i steps
and shift Bi,j to the up (with wraparound) by j steps.


Matrix A
Matrix B
a1 a2
a5 a6
a3 a4
a7 a8
0 left shift
a9 a10
a13 a14
a11 a12
a15 a16
1 left shift
b1 b2
b5 b6
b3 b4
b7 b8
a9 b10
b13 b14
b11 b12
b15 b16
0 up shift
1 up shift

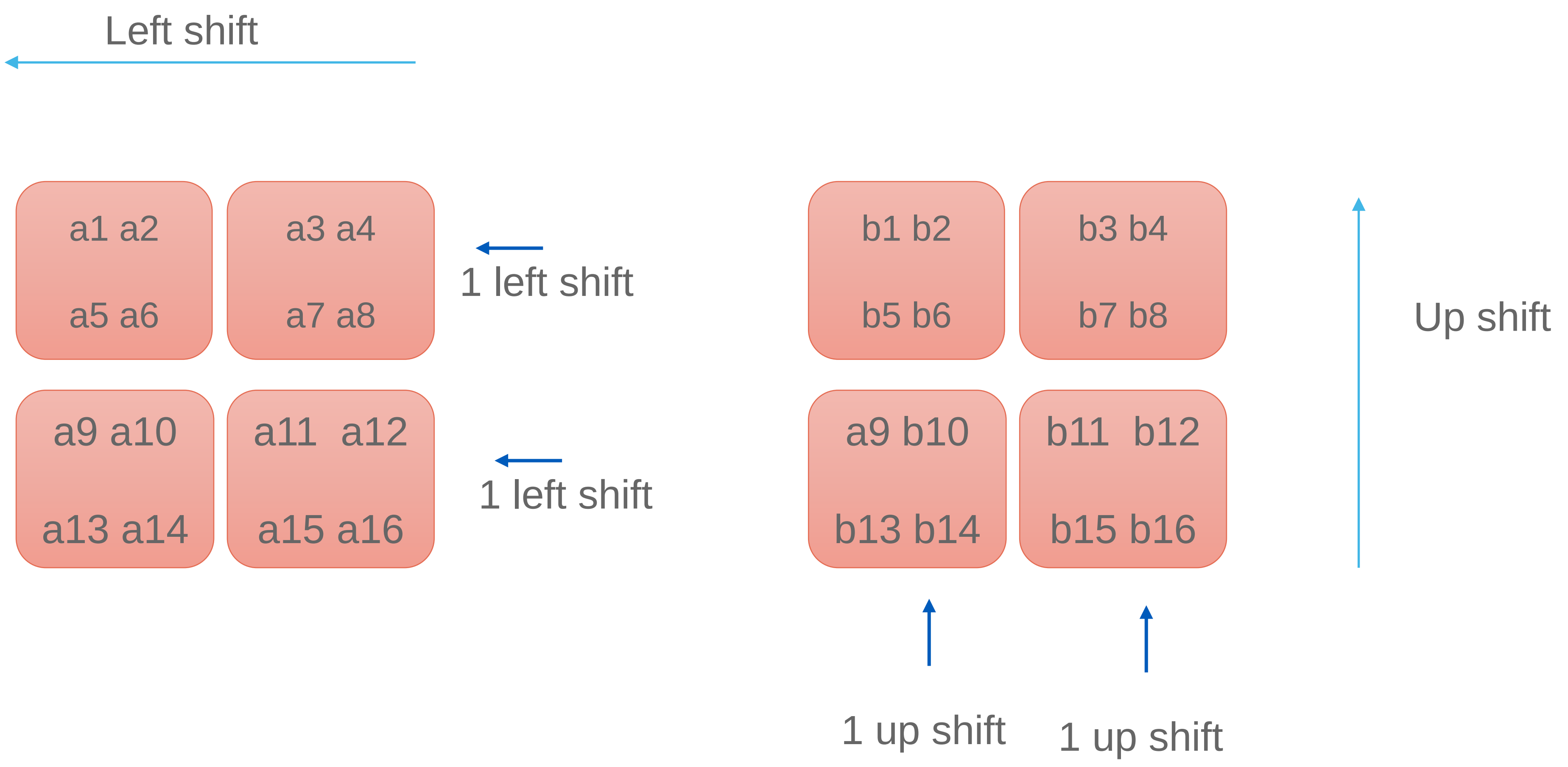


No of processors = 4

Shift and Compute:

for step :=1 to $p - 1$ do
Shift $A_{i,j}$ one step left (with wraparound) and $B_{i,j}$ one step up (with wraparound);

Process $P_{i,j}$ multiply received submatrices together and add the result to $C_{i,j}$;
Endfor;



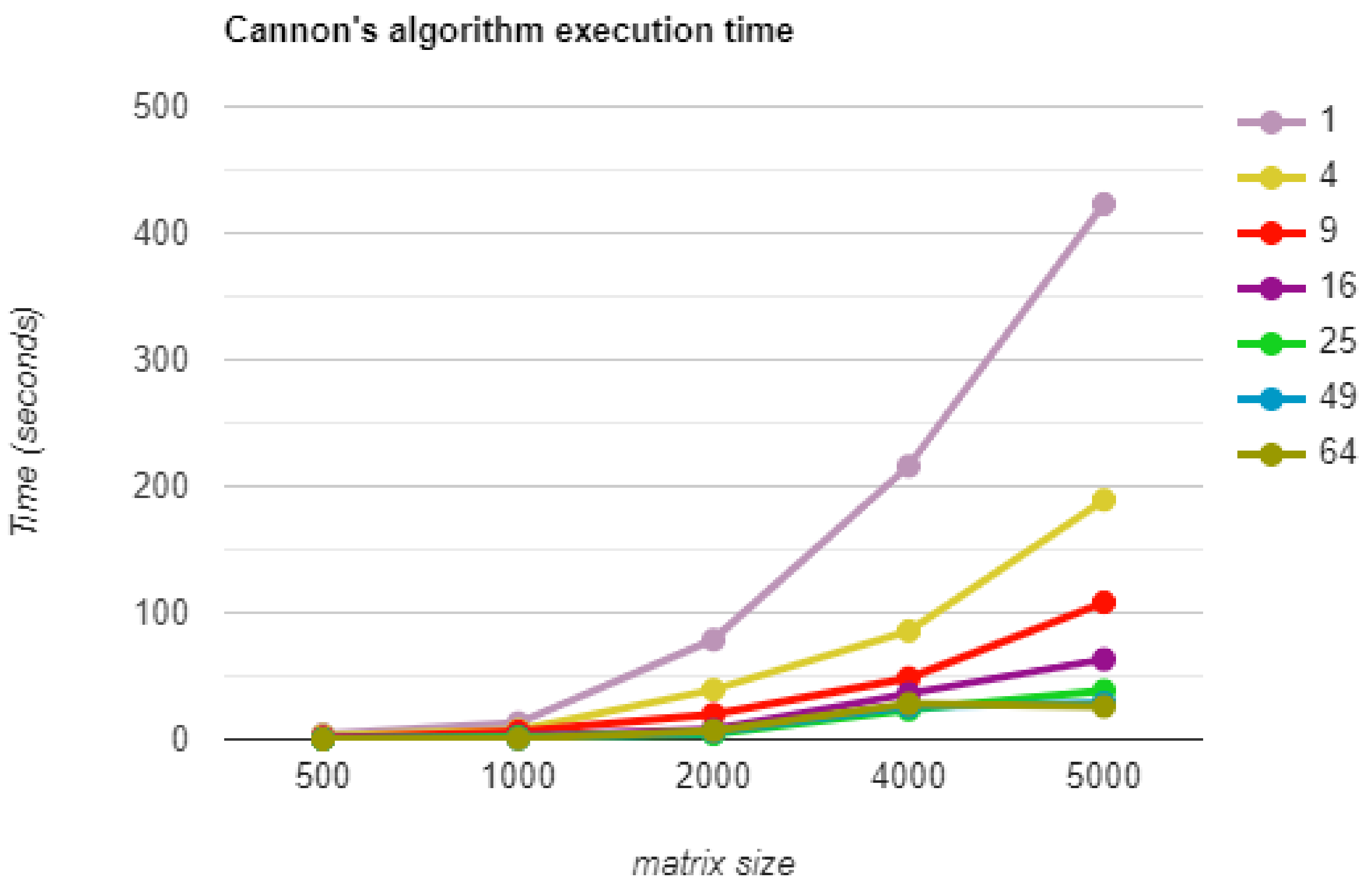
Repeat these steps for √p times

Resultant Matrix C

c1 c2 c5 c6	c3 c4 c7 c8
c9 c10 c13 c14	c11 c12 c15 c16

Execution time for Cannon's approach

	500 x 500	1000 x 1000	2000 x 2000	4000 x 4000	5000 x 5000
1	4	13	78.7	216	423
4	3.1	7.3	39.3	85.60	189.20
9	1.2	6.2	19.8	48.46	108.23
16	0.97	2.8	7.7	36.20	63.24
25	0.03	1.9	4.23	24.31	38.45
49	0.02	0.29	4.19	23.92	28.80
64	0.013	0.05	3.95	23.18	26.12


Cannon's algorithm execution time
500
400
300
200
100
0
Time (seconds)
500
1000
2000
4000
5000
matrix size
1
4
9
16
25
49
64

Observation

- Increasing number of processors did not always yield better results
- Although increasing number of processors yielded better results initially, not a huge difference was seen when with 25, 49 and 64 processors.
- Thus, for this particular problem we can conclude that (considering operational cost), 25 processors may work as well as 49 processors since the difference in run time is not significant.
- Manual experiment required to analyze right number of processors for different problems

References

https://cseweb.ucsd.edu//classes/fa12/cse260-b/Lectures/Lec13

https://people.eecs.berkeley.edu/~demmel/cs267/lecture11/lecture11.html#link_5

Gupta, Anshul; Kumar, Vipin; , "Scalability of Parallel Algorithms for Matrix Multiplication," Parallel Processing, 1993. ICPP 1993. International Conference on , vol.3, no., pp.115-123, 16-20 Aug. 1993 doi: 10.1109/ICPP.1993.160 URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4134256 &isnumber=4134231