Introduction of some well known parallel algorithms

www.see-grid.eu

SEE-GRID-SCI Training Event,
Yerevan, Armenia, 24-25 July 2008

Mikayel Gyurjyan
Institute for Informatics and Automation Problems
National Academy of Sciences of the Republic of Armenia
Mikayel_Gyurjyan@ipia.sci.am

The SEE-GRID-SCI initiative is co-funded by the European Commission under the FP7 Research Infrastructures contract no. 211338

Outline

- Matrix Multiplication (First Parallel Algorithm)
- Complexity Analysis (First Parallel Algorithm)
- Weakness of Algorithm 1
- Matrix Multiplication (Cannon's Algorithm)
- Complexity Analysis (Cannon's Algorithm)

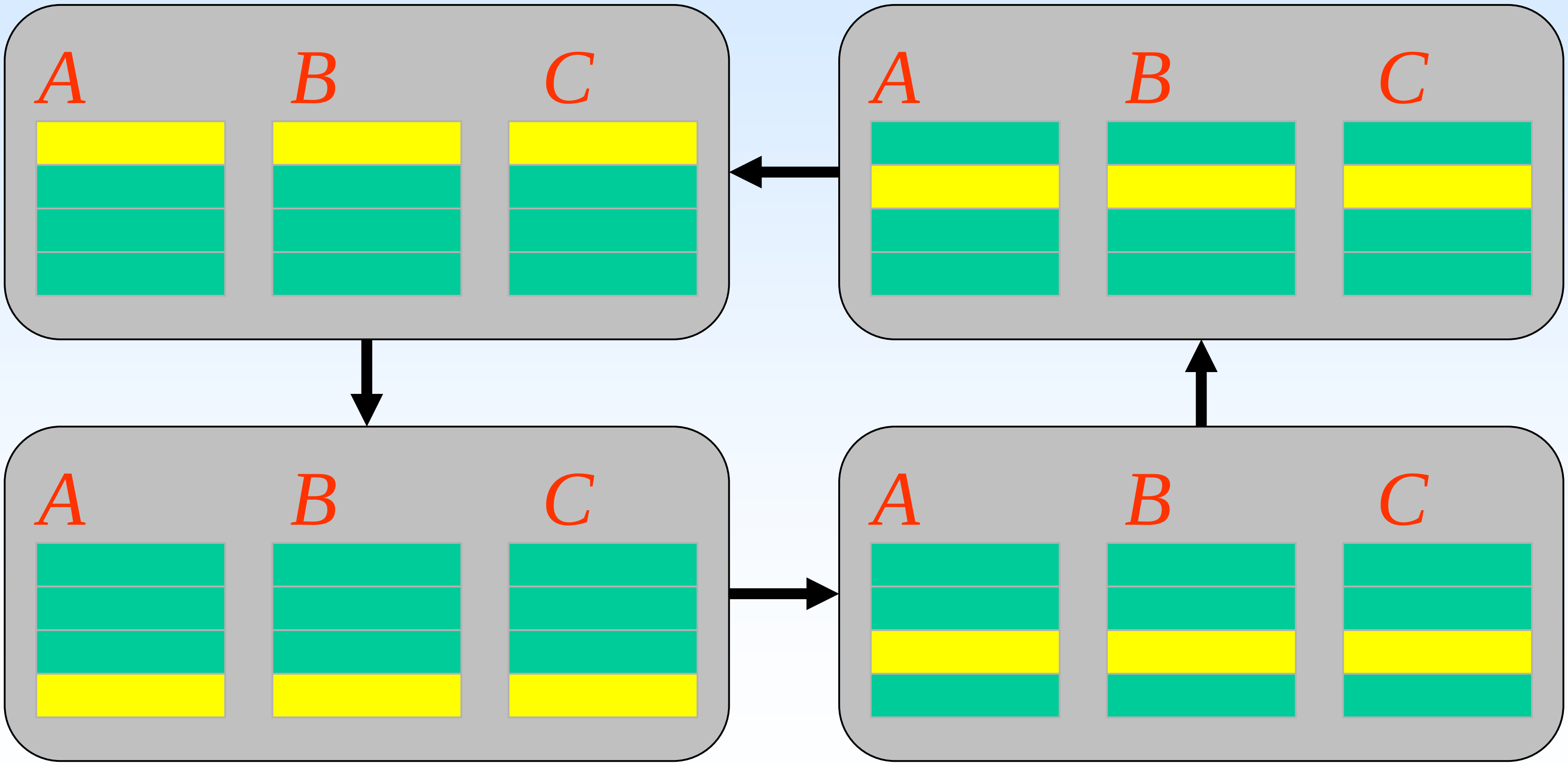
Matrix Multiplication (First Parallel Algorithm)

- Partitioning
  - Divide matrices into rows
  - Each primitive task has corresponding rows of three matrices
- Communication
  - Each task must eventually see every row of B
  - Organize tasks into a ring

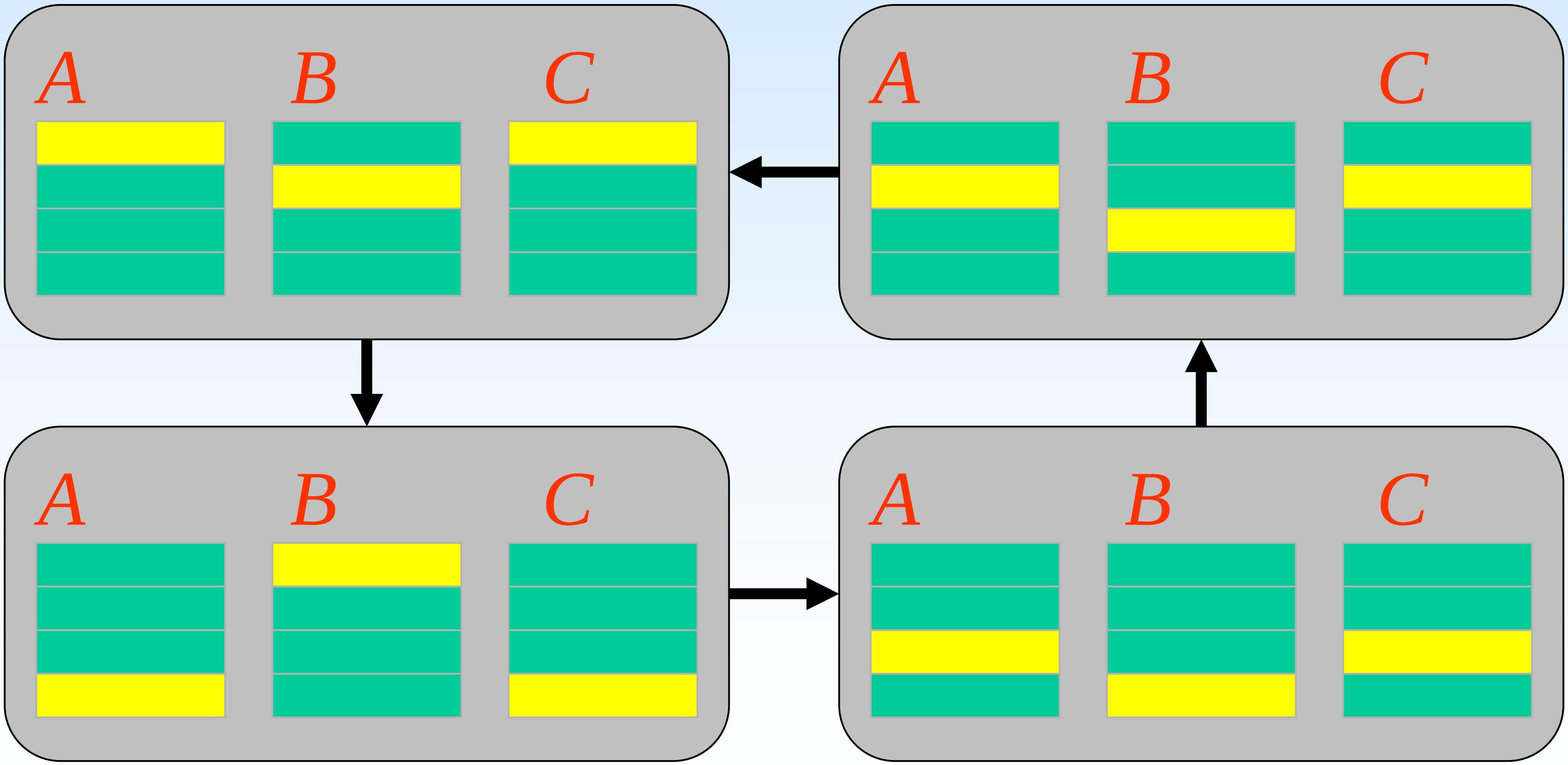
Matrix Multiplication (First Parallel Algorithm)

- Agglomeration and mapping
  - Fixed number of tasks, each requiring same amount of computation
  - Regular communication among tasks
  - Strategy: Assign each process a contiguous group of rows

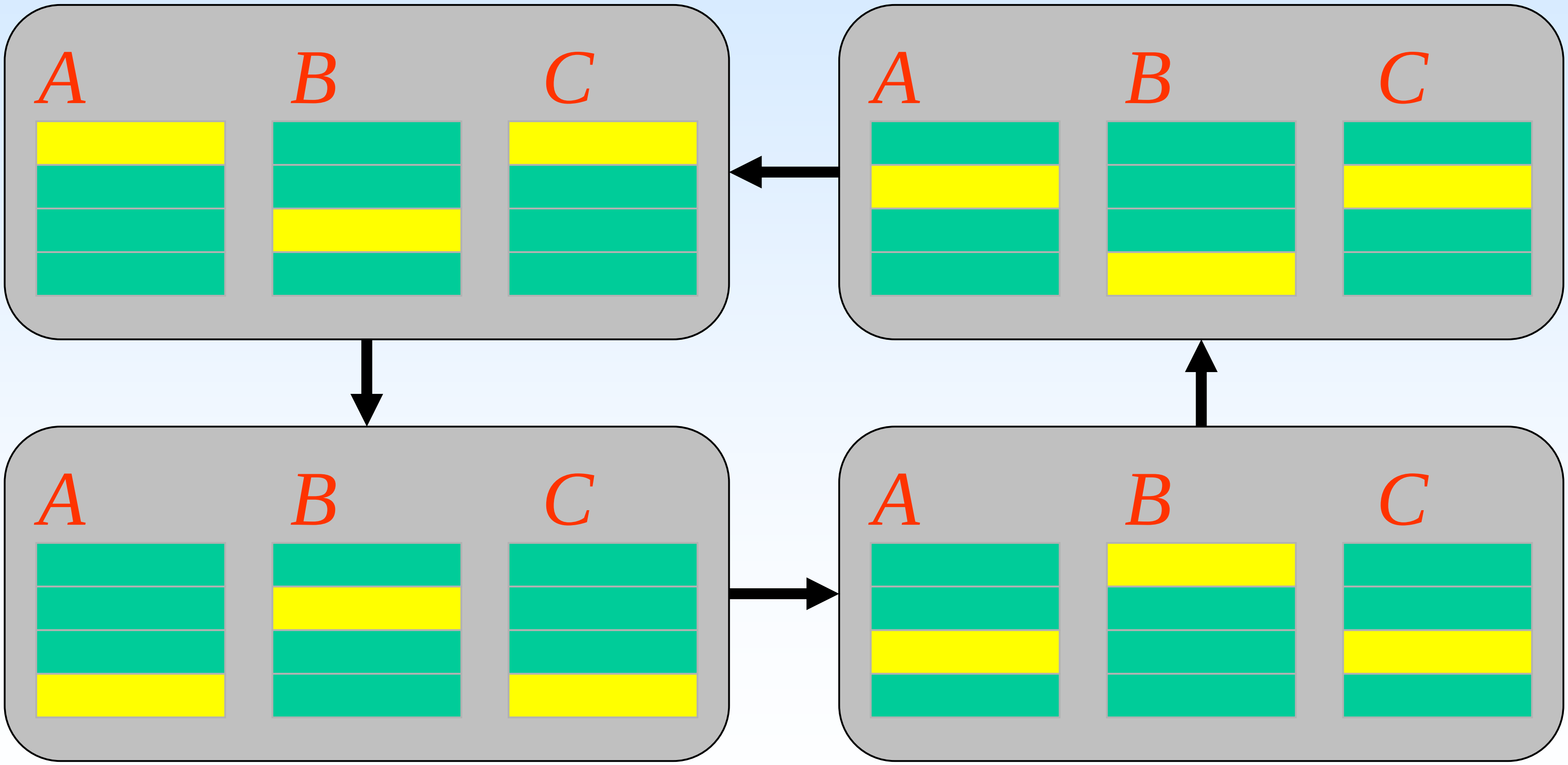
Communication of B(1)



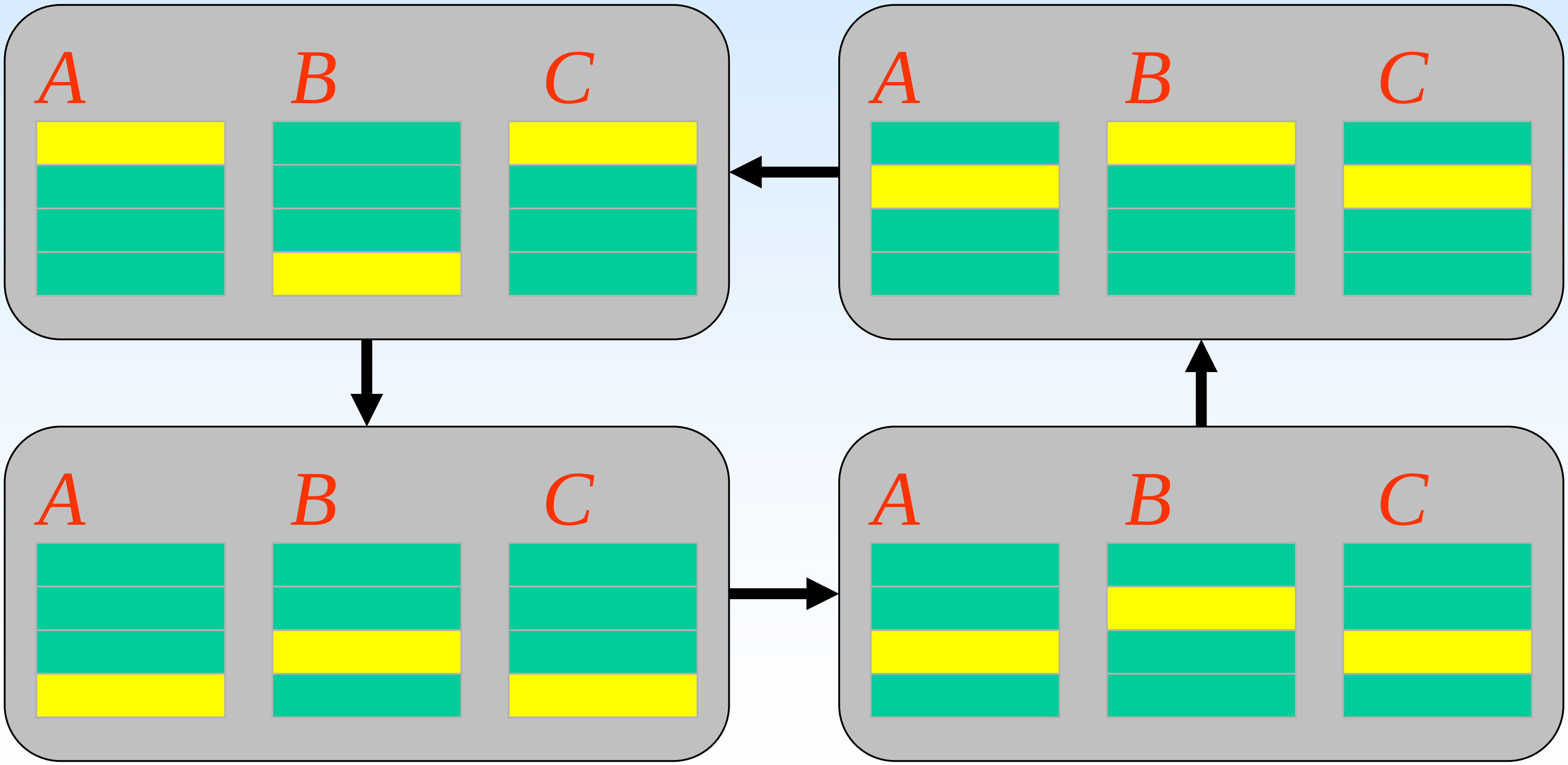
Communication of B(2)



Communication of B(3)



Communication of B(4)



Complexity Analysis

- Algorithm has p iterations
- During each iteration a process multiplies $(n / p) \times (n / p)$ block of A by $(n / p) \times n$ block of B: (n^3 / p^2)
- Total computation time: (n^3 / p)
- Each process ends up passing $(p-1)n^2/p = (n^2)$ elements of B

Weakness of Algorithm 1

- Blocks of B being manipulated have *p* times more columns than rows
- Each process must access every element of matrix B
- Ratio of computations per communication is poor: only $2n / p$

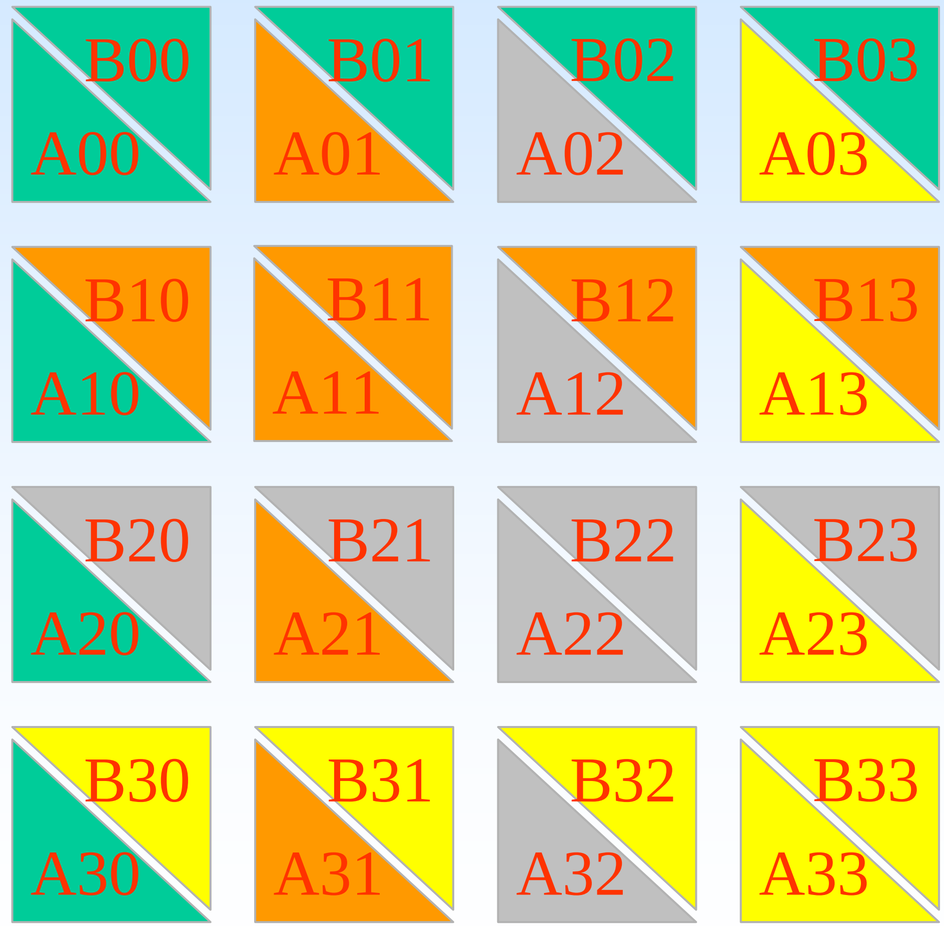
Parallel Algorithm 2 (Cannon's Algorithm)

- Associate a primitive task with each matrix element
- Agglomerate tasks responsible for a square (or nearly square) block of C
- Computation-to-communication ratio rises to $n / \sqrt{p}$

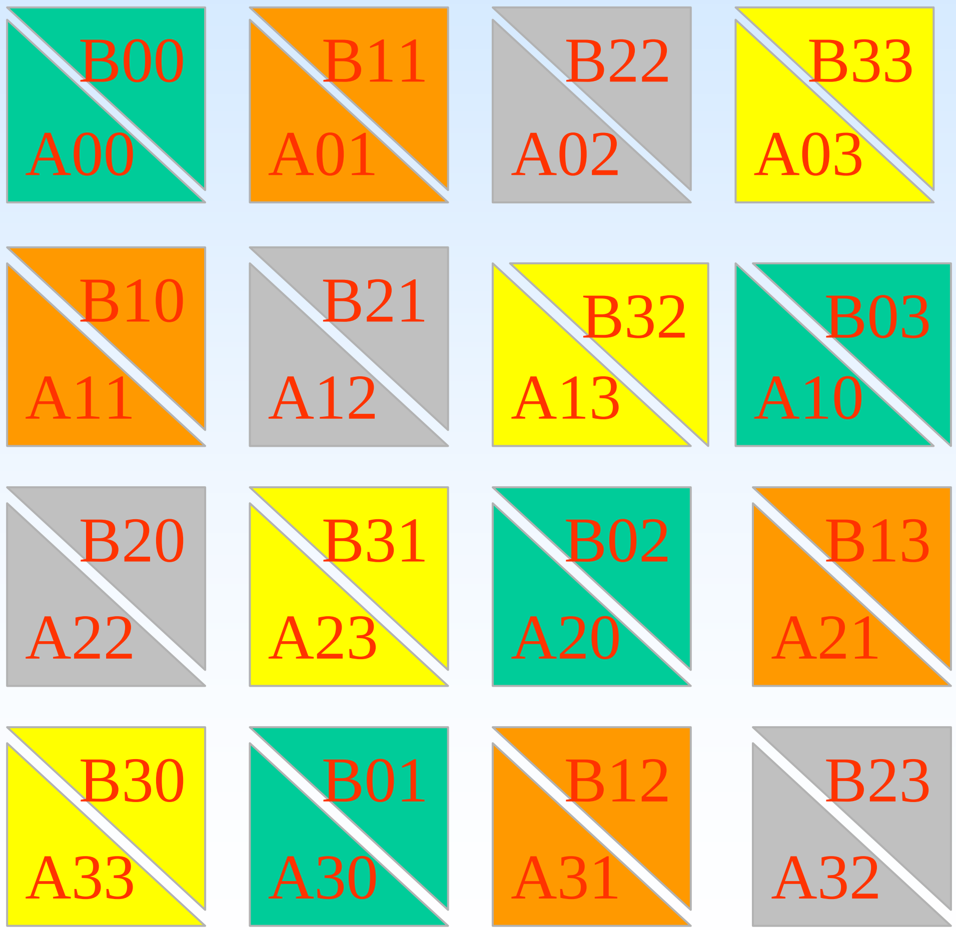
Blocks Need to Be Aligned

Each triangle represents a matrix block

Only same-color triangles should be multiplied



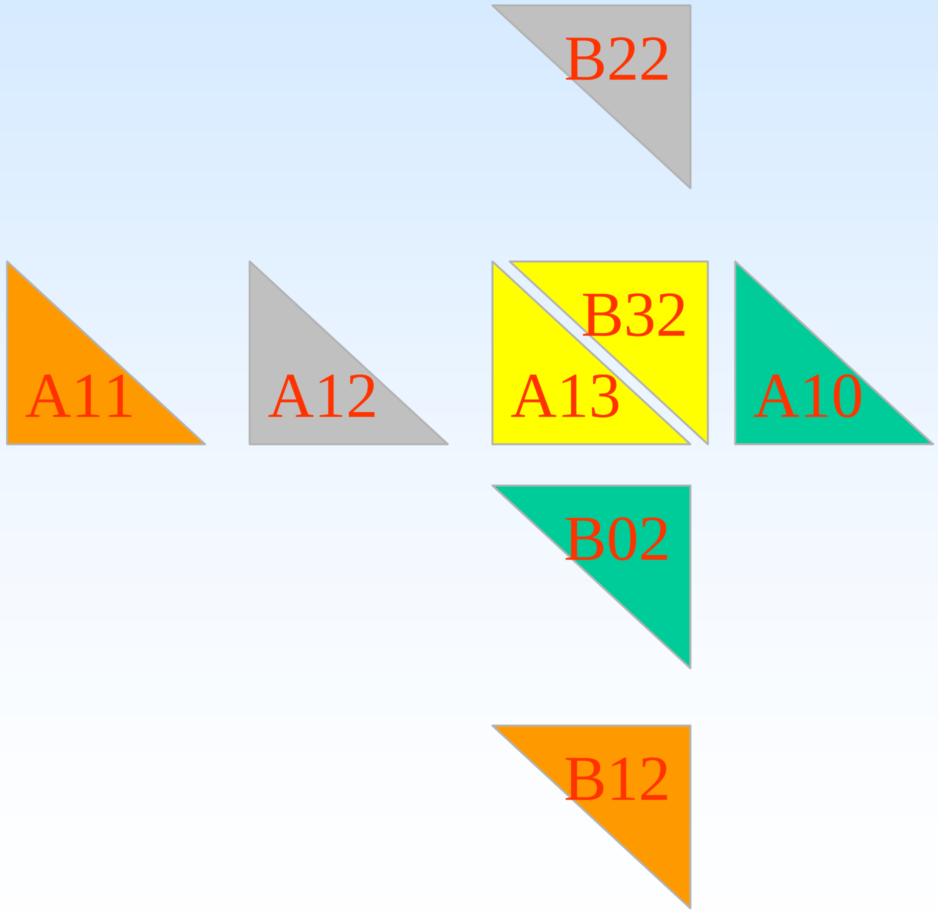
Rearrange Blocks



Block Aij cycles left i positions

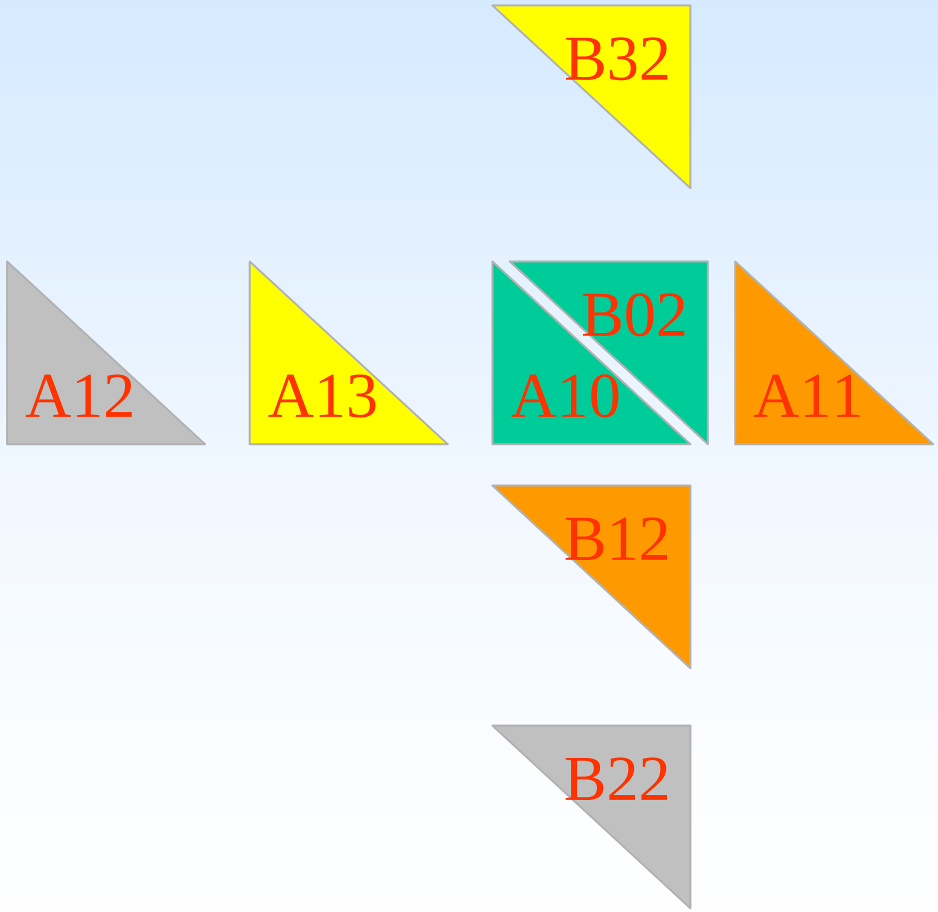
Block Bij cycles up j positions

Consider Process $P_{1,2}$



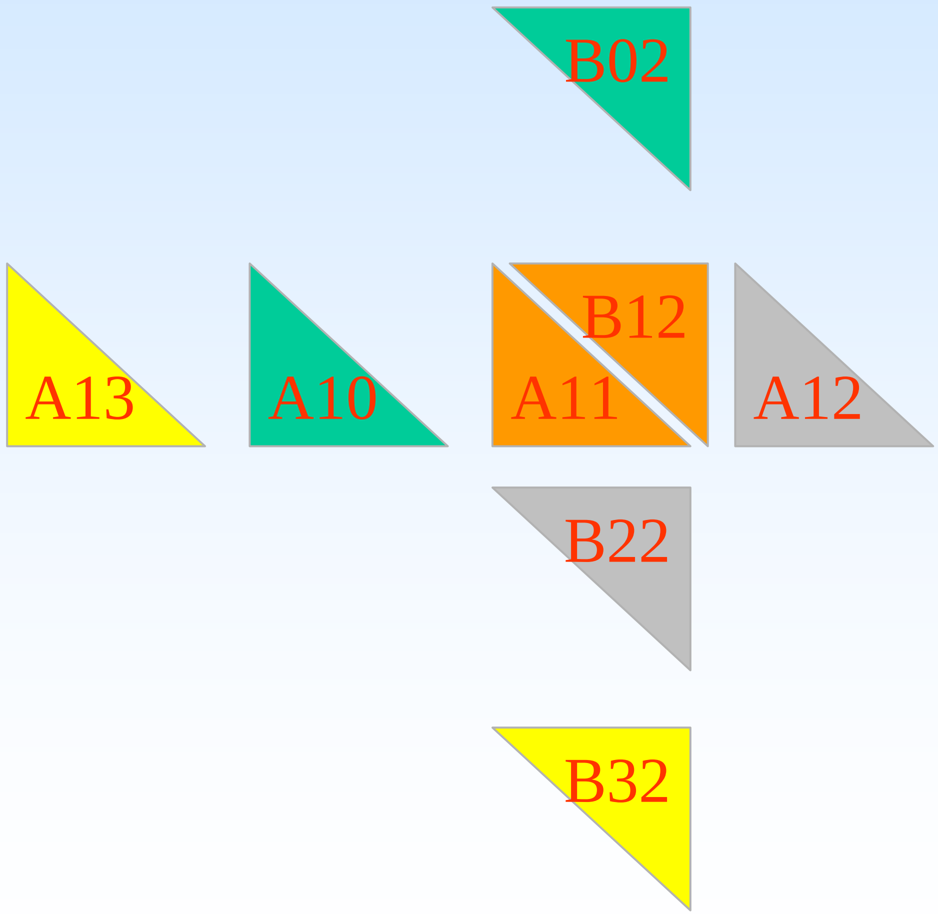
Step 1

Consider Process $P_{1,2}$



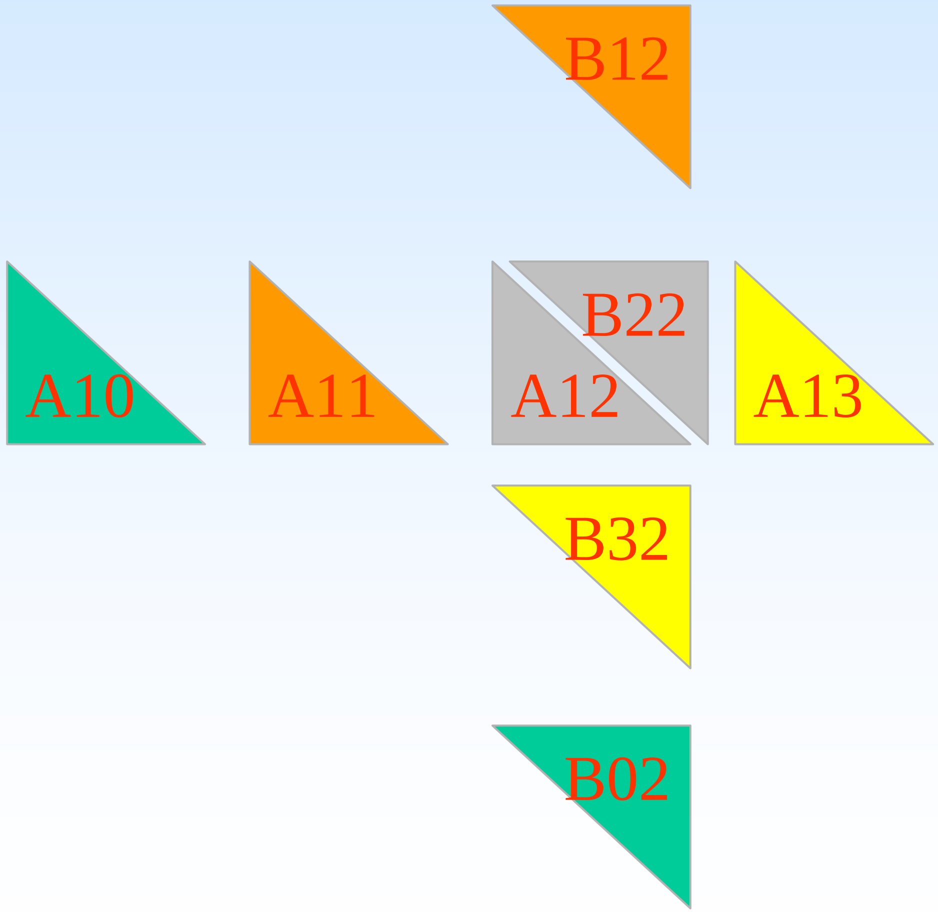
Step 2

Consider Process $P_{1,2}$



Step 3

Consider Process $P_{1,2}$



Step 4

Complexity Analysis

- Algorithm has $\sqrt{p}$ iterations
- During each iteration process multiplies two $(n / \sqrt{p}) \times (n / \sqrt{p})$ matrices: $(n^3 / p^{3/2})$
- Computational complexity: (n^3 / p)
- During each iteration process sends and receives two blocks of size $(n / \sqrt{p}) \times (n / \sqrt{p})$
- Communication complexity: $(n^2 / \sqrt{p})$

This system is highly scalable!

- Sequential algorithm: (n^3)
- Parallel overhead: (pn^2)