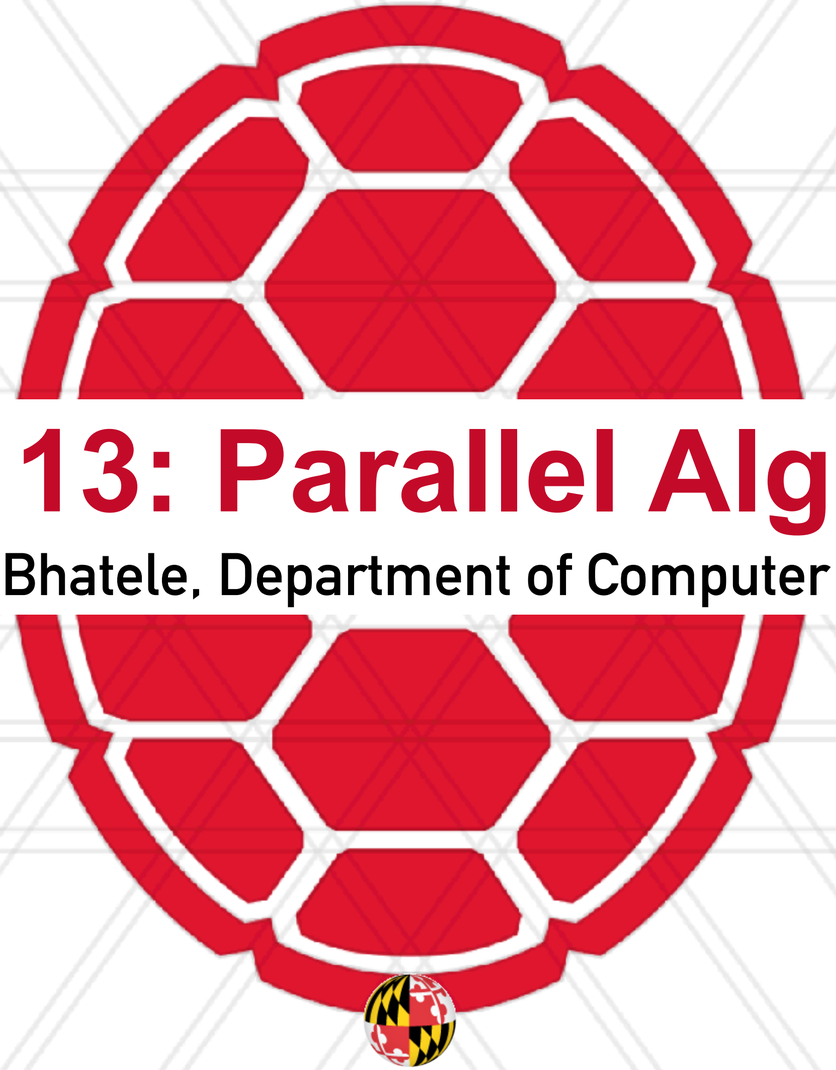
Introduction to Parallel Computing (CMSC498X / CMSC818X)



# Lecture 13: Parallel Algorithms

Abhinav Bhatele, Department of Computer Science

UNIVERSITY OF MARYLAND

# Announcements

- Assignment 2 is due on Oct 19
- Midterm on Oct 27
- Interim report due on November 16

# Group project

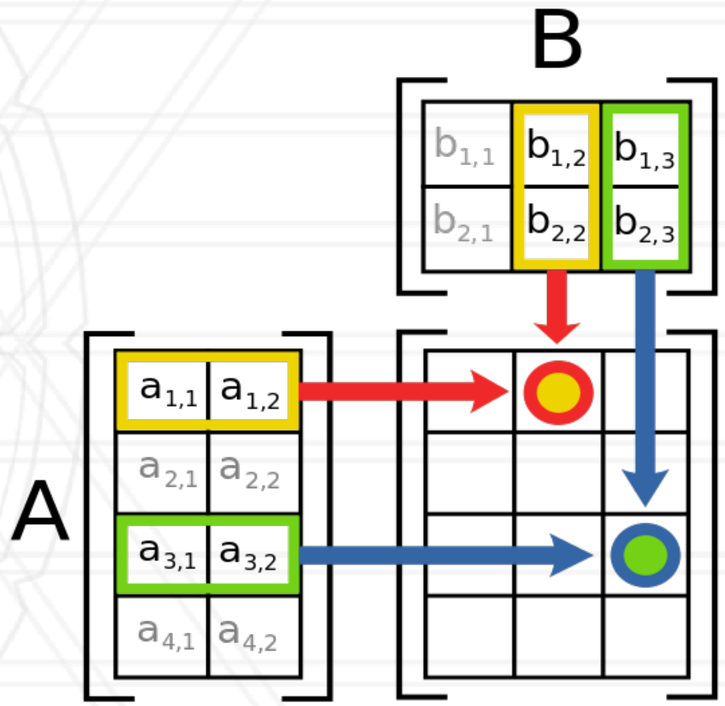
- Check your email for project feedback
- Reply to email if you want to discuss further

| | |
|---|---|
| **Presentation** | 40 |
| **Final report and code** | 40 |
| **Peer evaluation** | 20 |

- Peer evaluation: you are given $100 that you will allocate as a performance bonus to your group members based on your assessment of their contributions to the project (you cannot keep any money for yourself)

# Matrix multiplication

```
for (i=0; i<M; i++)
  for (j=0; j<N; j++)
    for (k=0; k<L; k++)
      C[i][j] += A[i][k]*B[k][j];
```

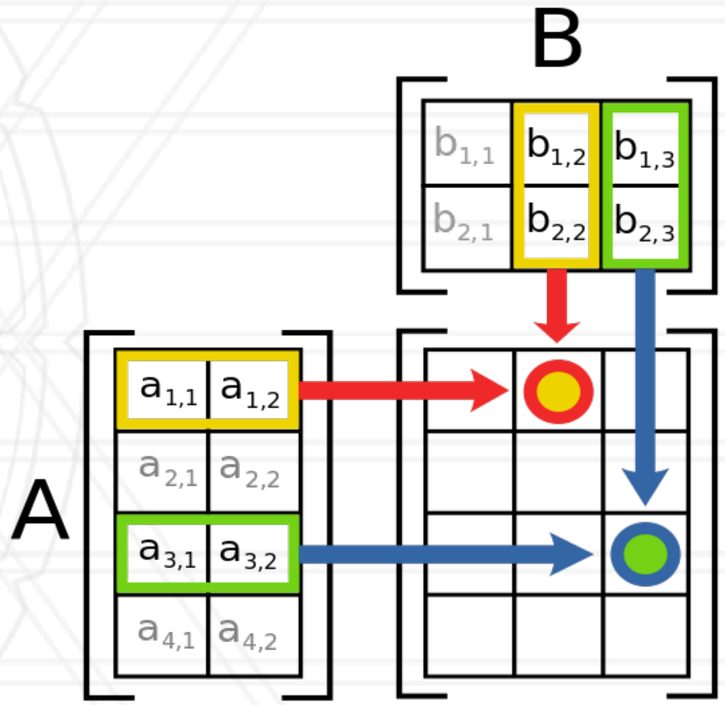



https://en.wikipedia.org/wiki/Matrix_multiplication

# Matrix multiplication

```
for (i=0; i<M; i++)
  for (j=0; j<N; j++)
    for (k=0; k<L; k++)
      C[i][j] += A[i][k]*B[k][j];
```

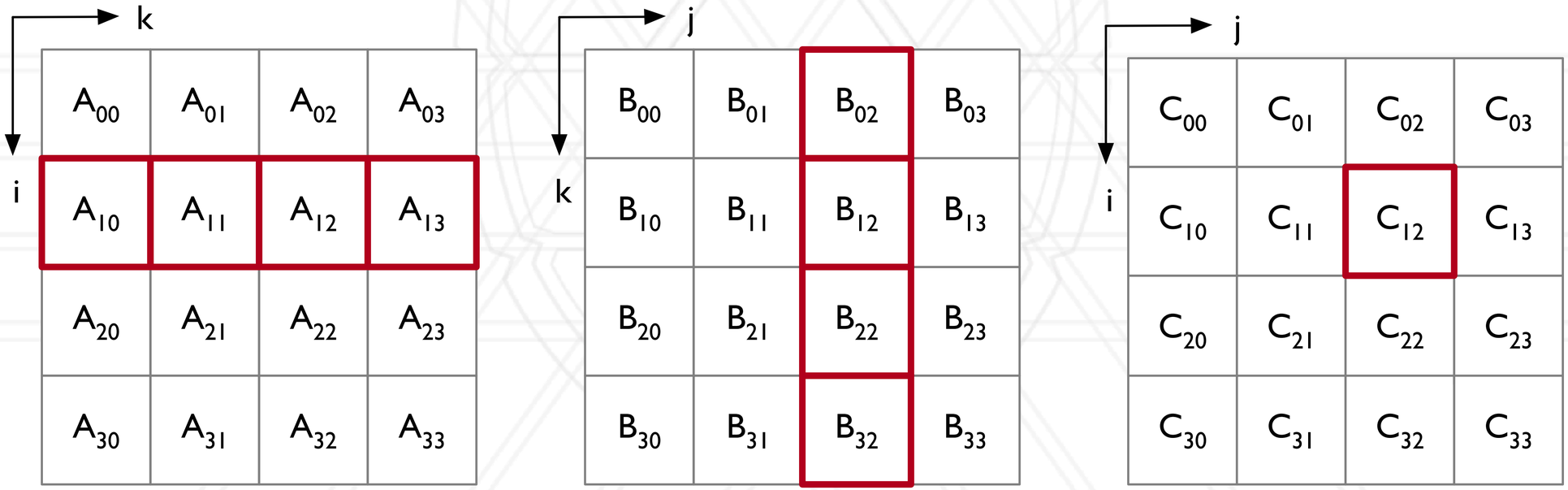
Any performance issues for large arrays?




https://en.wikipedia.org/wiki/Matrix_multiplication

# Blocking to improve cache performance

- Create smaller blocks that fit in cache: leads to cache reuse
- $C_{12} = A_{10} * B_{02} + A_{11} * B_{12} + A_{12} * B_{22} + A_{13} * B_{32}$



# Parallel matrix multiply

- Store A and B in a distributed manner
- Communication between processes to get the right sub-matrices to each process
- Each process computes a portion of C

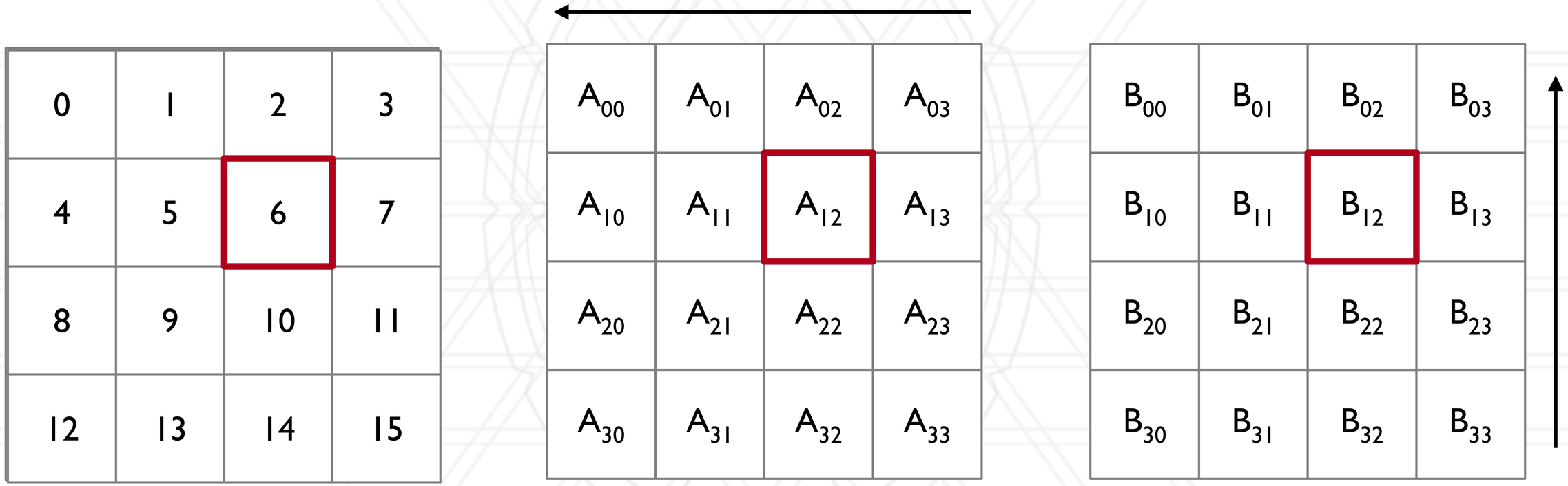
# Cannon's 2D matrix multiply

| | | | |
|---|---|---|---|
| 0 | 1 | 2 | 3 |
| 4 | 5 | 6 | 7 |
| 8 | 9 | 10 | 11 |
| 12 | 13 | 14 | 15 |

| | | | |
|---|---|---|---|
| $A_{00}$ | $A_{01}$ | $A_{02}$ | $A_{03}$ |
| $A_{10}$ | $A_{11}$ | $A_{12}$ | $A_{13}$ |
| $A_{20}$ | $A_{21}$ | $A_{22}$ | $A_{23}$ |
| $A_{30}$ | $A_{31}$ | $A_{32}$ | $A_{33}$ |

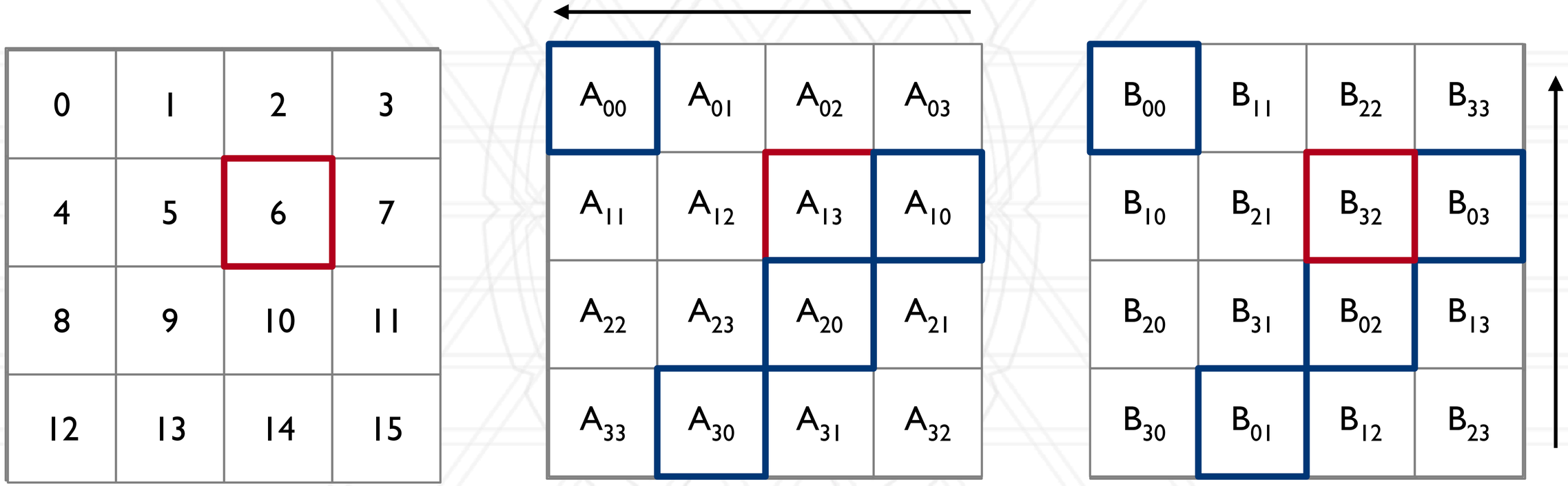
| | | | |
|---|---|---|---|
| $B_{00}$ | $B_{01}$ | $B_{02}$ | $B_{03}$ |
| $B_{10}$ | $B_{11}$ | $B_{12}$ | $B_{13}$ |
| $B_{20}$ | $B_{21}$ | $B_{22}$ | $B_{23}$ |
| $B_{30}$ | $B_{31}$ | $B_{32}$ | $B_{33}$ |

# Cannon's 2D matrix multiply



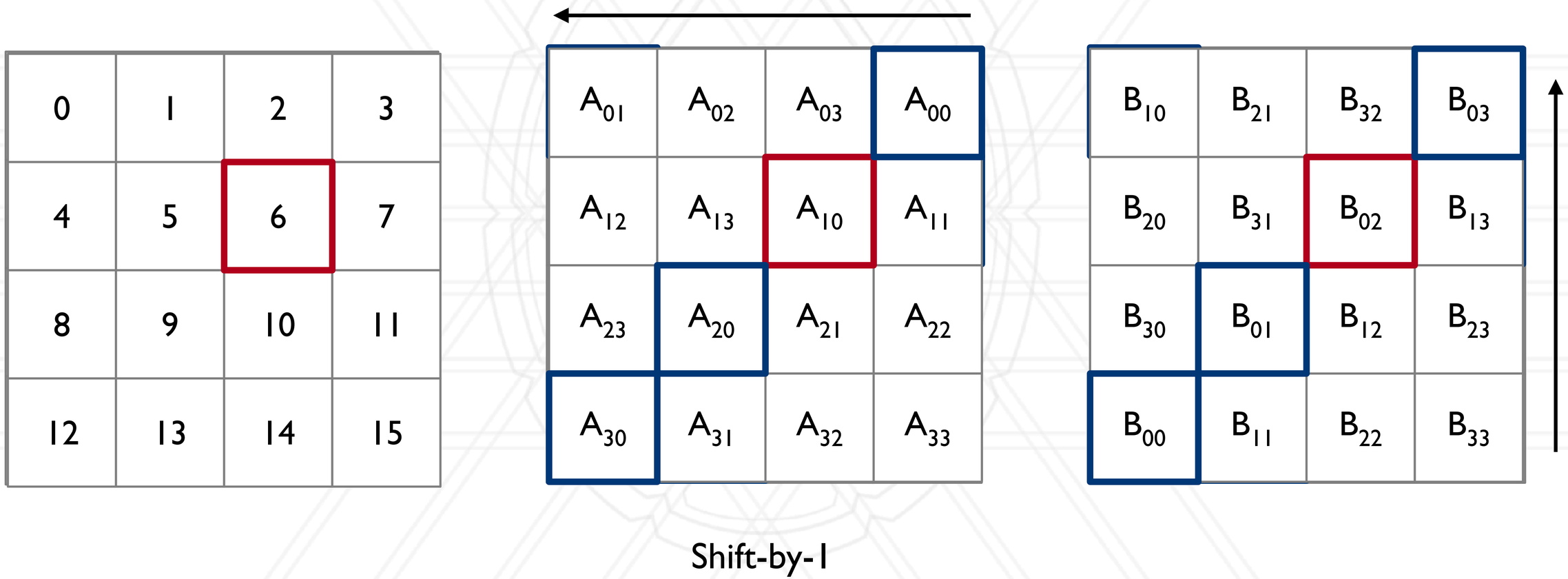

Initial skew

# Cannon's 2D matrix multiply




Initial skew

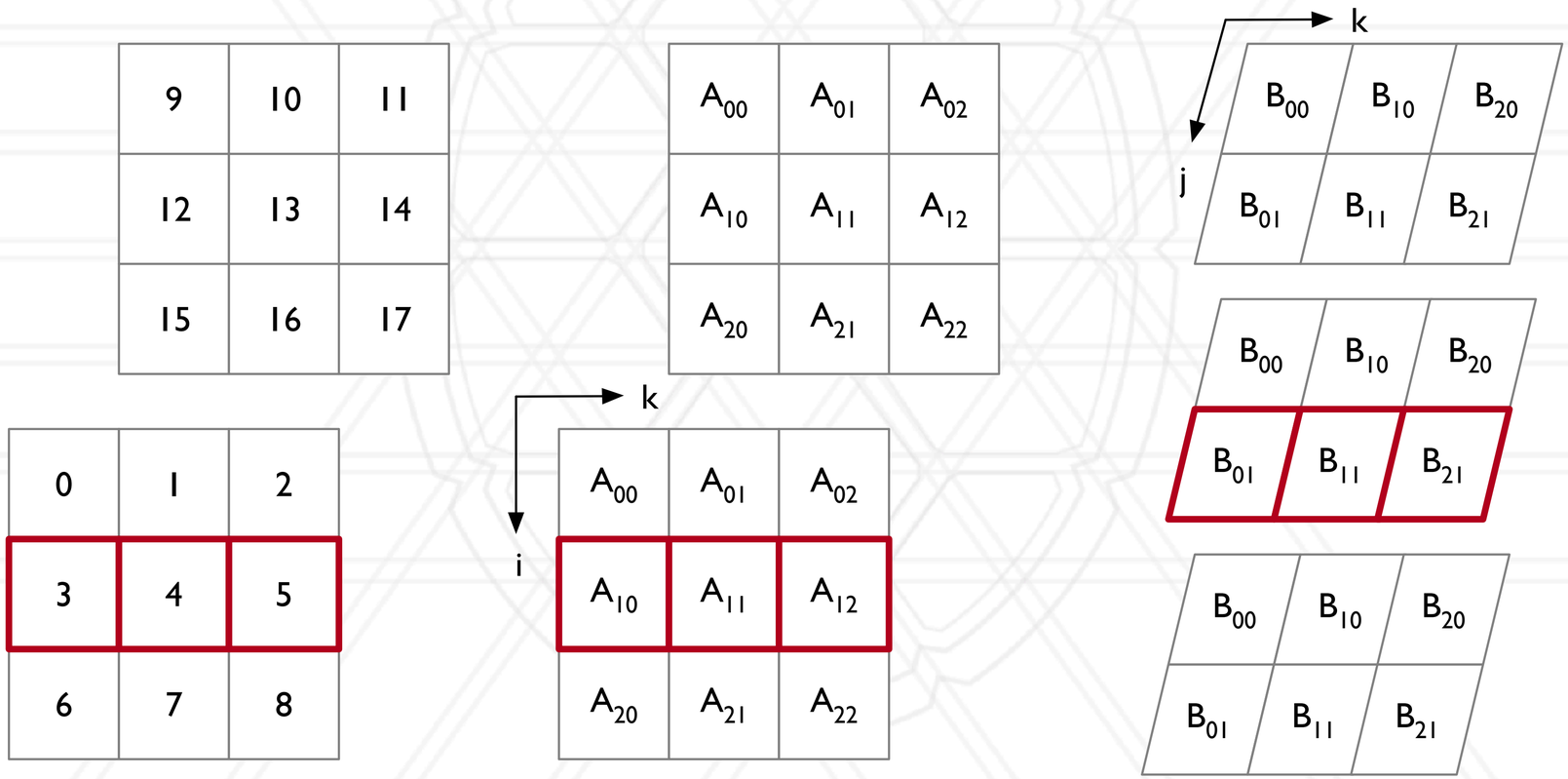
# Cannon's 2D matrix multiply




Shift-by-1

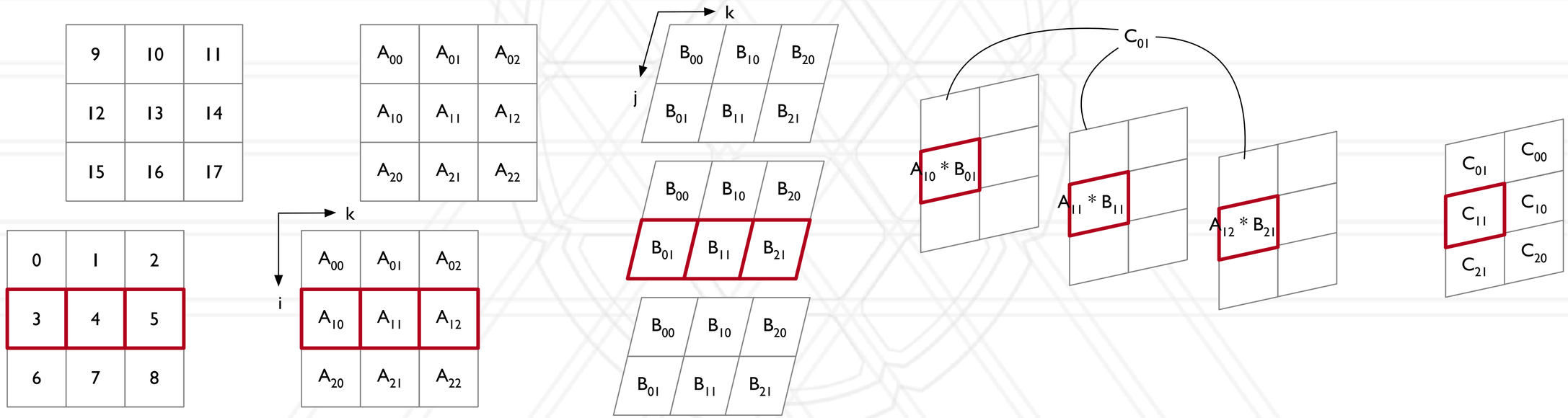
# Agarwal's 3D matrix multiply

- Copy A to all i-k planes and B to all j-k planes



# Agarwal's 3D matrix multiply

- Perform a single matrix multiply to calculate partial C
- All-to-all along i-j planes to calculate final result



UNIVERSITY OF
MARYLAND

Abhinav Bhatele

5218 Brendan Iribe Center (IRB) / College Park, MD 20742

phone: 301.405.4507 / e-mail: bhatele@cs.umd.edu