Parallel Algorithms

Matrix Multiplication

Pramod Ganapathi

Square Matrix Multiplication

Example

$$\begin{bmatrix} 2 & 7 & 3 & 6 \\ 5 & 8 & 3 & 8 \\ 6 & 4 & 5 & 6 \\ 0 & 3 & 9 & 7 \end{bmatrix} \times \begin{bmatrix} 8 & 4 & 4 & 3 \\ 7 & 7 & 6 & 8 \\ 5 & 3 & 8 & 4 \\ 2 & 5 & 5 & 7 \end{bmatrix} = \begin{bmatrix} 92 & 96 & 104 & 116 \\ 127 & 125 & 132 & 147 \\ 113 & 97 & 118 & 112 \\ 80 & 83 & 125 & 109 \end{bmatrix}$$

- A's ith row $\times$ B's jth column $= C[i, j]$ cell
- E.g.: $5 \times 4 + 8 \times 6 + 3 \times 8 + 8 \times 5 = 132$

Square Matrix Multiplication

Example

$$\begin{bmatrix} 2 & 7 & 3 & 6 \\ 5 & 8 & 3 & 8 \\ 6 & 4 & 5 & 6 \\ 0 & 3 & 9 & 7 \end{bmatrix} \times \begin{bmatrix} 8 & 4 & 4 & 3 \\ 7 & 7 & 6 & 8 \\ 5 & 3 & 8 & 4 \\ 2 & 5 & 5 & 7 \end{bmatrix} = \begin{bmatrix} 92 & 96 & 104 & 116 \\ 127 & 125 & 132 & 147 \\ 113 & 97 & 118 & 112 \\ 80 & 83 & 125 & 109 \end{bmatrix}$$

- A's ith row $\times$ B's jth column $= C[i,j]$ cell
- E.g.: $5 \times 4 + 8 \times 6 + 3 \times 8 + 8 \times 5 = 132$

Definition

If A and B are $n \times n$ matrices consisting of real numbers, then the matrix product $C = A \times B$ is defined and computed as

$$C[i,j] = \sum_{k=1}^{n} A[i,k] \times B[k,j] \text{ for } i,j \in [1,n]$$

MM Loops

MM-Loop(A, B, C, n)

for $i \leftarrow 1$ to n do
  for $j \leftarrow 1$ to n do
    $C[i, j] \leftarrow 0$
    for $k \leftarrow 1$ to n do
      $C[i, j] \leftarrow C[i, j] + A[i, k] \times B[k, j]$

MM Loops

MM-Loop(A, B, C, n)

1. for $i \leftarrow 1$ to n do
2. for $j \leftarrow 1$ to n do
3. $C[i,j] \leftarrow 0$
4. for $k \leftarrow 1$ to n do
5. $C[i,j] \leftarrow C[i,j] + A[i,k] \times B[k,j]$

$T_1(n)$	$T_\infty(n)$	$E_1(n)$	$S_\infty(n)$	$Q_1(n)$
$\Theta(n^3)$	$\Theta(n^3)$	$\Theta(1)$	$\Theta(n^2)$	$\Theta(n^3)$

MM Loops

MM-Loop(A, B, C, n)

1. for $i \leftarrow 1$ to n do
2. for $j \leftarrow 1$ to n do
3. $C[i, j] \leftarrow 0$
4. for $k \leftarrow 1$ to n do
5. $C[i, j] \leftarrow C[i, j] + A[i, k] \times B[k, j]$

$T_1(n)$	$T_\infty(n)$	$E_1(n)$	$S_\infty(n)$	$Q_1(n)$
$\Theta\left(n^3\right)$	$\Theta\left(n^3\right)$	$\Theta\left(1\right)$	$\Theta\left(n^2\right)$	$\Theta\left(n^3\right)$

- The A, B, C matrices are stored in row-major order
- How can we improve cache complexity?

MM Loops

MM-Loop(A, B, C, n)

1. for $i \leftarrow 1$ to n do
2. for $j \leftarrow 1$ to n do
3. $C[i, j] \leftarrow 0$
4. for $k \leftarrow 1$ to n do
5. $C[i, j] \leftarrow C[i, j] + A[i, k] \times B[k, j]$

$T_1(n)$	$T_\infty(n)$	$E_1(n)$	$S_\infty(n)$	$Q_1(n)$
$\Theta(n^3)$	$\Theta(n^3)$	$\Theta(1)$	$\Theta(n^2)$	$\Theta(n^3)$

- The A, B, C matrices are stored in row-major order
- How can we improve cache complexity?
  Reorder the loops!

MM Loops ($3! = 6$ possible ways)

MM-ijk(A, B, C, n)

1. for $i \leftarrow 1$ to n do
2. for $j \leftarrow 1$ to n do
3. for $k \leftarrow 1$ to n do
4. $C[i, j] \leftarrow C[i, j] + A[i, k] \cdot B[k, j]$

MM-ikj(A, B, C, n)

1. for $i \leftarrow 1$ to n do
2. for $k \leftarrow 1$ to n do
3. for $j \leftarrow 1$ to n do
4. $C[i, j] \leftarrow C[i, j] + A[i, k] \cdot B[k, j]$

MM-jik(A, B, C, n)

1. for $j \leftarrow 1$ to n do
2. for $i \leftarrow 1$ to n do
3. for $k \leftarrow 1$ to n do
4. $C[i, j] \leftarrow C[i, j] + A[i, k] \cdot B[k, j]$

MM-jki(A, B, C, n)

1. for $j \leftarrow 1$ to n do
2. for $k \leftarrow 1$ to n do
3. for $i \leftarrow 1$ to n do
4. $C[i, j] \leftarrow C[i, j] + A[i, k] \cdot B[k, j]$

MM-kij(A, B, C, n)

1. for $k \leftarrow 1$ to n do
2. for $i \leftarrow 1$ to n do
3. for $j \leftarrow 1$ to n do
4. $C[i, j] \leftarrow C[i, j] + A[i, k] \cdot B[k, j]$

MM-kji(A, B, C, n)

1. for $k \leftarrow 1$ to n do
2. for $j \leftarrow 1$ to n do
3. for $i \leftarrow 1$ to n do
4. $C[i, j] \leftarrow C[i, j] + A[i, k] \cdot B[k, j]$

MM Loops ($3! = 6$ possible ways)

MM-ijk(A, B, C, n)

```
for i ← 1 to n do
  for j ← 1 to n do
    for k ← 1 to n do
      C[i, j] ← C[i, j] + A[i, k] · B[k, j]
```

MM-ikj(A, B, C, n)

```
for i ← 1 to n do
  for k ← 1 to n do
    for j ← 1 to n do
      C[i, j] ← C[i, j] + A[i, k] · B[k, j]
```

MM-jik(A, B, C, n)

```
for j ← 1 to n do
  for i ← 1 to n do
    for k ← 1 to n do
      C[i, j] ← C[i, j] + A[i, k] · B[k, j]
```

MM-jki(A, B, C, n)

```
for j ← 1 to n do
  for k ← 1 to n do
    for i ← 1 to n do
      C[i, j] ← C[i, j] + A[i, k] · B[k, j]
```

MM-kij(A, B, C, n)

```
for k ← 1 to n do
  for i ← 1 to n do
    for j ← 1 to n do
      C[i, j] ← C[i, j] + A[i, k] · B[k, j]
```

MM-kji(A, B, C, n)

```
for k ← 1 to n do
  for j ← 1 to n do
    for i ← 1 to n do
      C[i, j] ← C[i, j] + A[i, k] · B[k, j]
```

- Which of these algorithms are correct?

MM Loops ($3! = 6$ possible ways)

MM-ijk(A, B, C, n)

1. for $i \leftarrow 1$ to n do
2. for $j \leftarrow 1$ to n do
3. for $k \leftarrow 1$ to n do
4. $C[i,j] \leftarrow C[i,j] + A[i,k] \cdot B[k,j]$

MM-ikj(A, B, C, n)

1. for $i \leftarrow 1$ to n do
2. for $k \leftarrow 1$ to n do
3. for $j \leftarrow 1$ to n do
4. $C[i,j] \leftarrow C[i,j] + A[i,k] \cdot B[k,j]$

MM-jik(A, B, C, n)

1. for $j \leftarrow 1$ to n do
2. for $i \leftarrow 1$ to n do
3. for $k \leftarrow 1$ to n do
4. $C[i,j] \leftarrow C[i,j] + A[i,k] \cdot B[k,j]$

MM-jki(A, B, C, n)

1. for $j \leftarrow 1$ to n do
2. for $k \leftarrow 1$ to n do
3. for $i \leftarrow 1$ to n do
4. $C[i,j] \leftarrow C[i,j] + A[i,k] \cdot B[k,j]$

MM-kij(A, B, C, n)

1. for $k \leftarrow 1$ to n do
2. for $i \leftarrow 1$ to n do
3. for $j \leftarrow 1$ to n do
4. $C[i,j] \leftarrow C[i,j] + A[i,k] \cdot B[k,j]$

MM-kji(A, B, C, n)

1. for $k \leftarrow 1$ to n do
2. for $j \leftarrow 1$ to n do
3. for $i \leftarrow 1$ to n do
4. $C[i,j] \leftarrow C[i,j] + A[i,k] \cdot B[k,j]$

- Which of these algorithms are correct?
- Which of the correct algorithms have improved cache locality?

MM Loops: Correctness

Correctness

All 6 algorithms are correct because they satisfy the read-write constraints of the MM definition.

Technique to prove correctness of the looping algorithms

- Check if the read-write constraints are satisfied
- Check if the order of computations are as desired
- Check if no more instructions are executed
- Check if no fewer instructions are executed

MM Loops: Complexity

Algorithm	$T_1(n)$	$T_\infty(n)$	$E_1(n)$	$S_\infty(n)$	$Q_1(n)$
MM-ijk	$\Theta\left(n^3\right)$	$\Theta\left(n^3\right)$	$\Theta\left(1\right)$	$\Theta\left(n^2\right)$	$\Theta\left(n^3\right)$
MM-ikj	$\Theta\left(n^3\right)$	$\Theta\left(n^3\right)$	$\Theta\left(1\right)$	$\Theta\left(n^2\right)$	$\Theta\left(\frac{n^3}{B} + n^2\right)$
MM-jik	$\Theta\left(n^3\right)$	$\Theta\left(n^3\right)$	$\Theta\left(1\right)$	$\Theta\left(n^2\right)$	$\Theta\left(n^3\right)$
MM-jki	$\Theta\left(n^3\right)$	$\Theta\left(n^3\right)$	$\Theta\left(1\right)$	$\Theta\left(n^2\right)$	$\Theta\left(n^3\right)$
MM-kij	$\Theta\left(n^3\right)$	$\Theta\left(n^3\right)$	$\Theta\left(1\right)$	$\Theta\left(n^2\right)$	$\Theta\left(n^3\right)$
MM-kji	$\Theta\left(n^3\right)$	$\Theta\left(n^3\right)$	$\Theta\left(1\right)$	$\Theta\left(n^2\right)$	$\Theta\left(n^3\right)$

MM Loops: Complexity

Algorithm	$T_1(n)$	$T_\infty(n)$	$E_1(n)$	$S_\infty(n)$	$Q_1(n)$
MM-ijk	$\Theta\left(n^3\right)$	$\Theta\left(n^3\right)$	$\Theta\left(1\right)$	$\Theta\left(n^2\right)$	$\Theta\left(n^3\right)$
MM-ikj	$\Theta\left(n^3\right)$	$\Theta\left(n^3\right)$	$\Theta\left(1\right)$	$\Theta\left(n^2\right)$	$\Theta\left(\frac{n^3}{B} + n^2\right)$
MM-jik	$\Theta\left(n^3\right)$	$\Theta\left(n^3\right)$	$\Theta\left(1\right)$	$\Theta\left(n^2\right)$	$\Theta\left(n^3\right)$
MM-jki	$\Theta\left(n^3\right)$	$\Theta\left(n^3\right)$	$\Theta\left(1\right)$	$\Theta\left(n^2\right)$	$\Theta\left(n^3\right)$
MM-kij	$\Theta\left(n^3\right)$	$\Theta\left(n^3\right)$	$\Theta\left(1\right)$	$\Theta\left(n^2\right)$	$\Theta\left(n^3\right)$
MM-kji	$\Theta\left(n^3\right)$	$\Theta\left(n^3\right)$	$\Theta\left(1\right)$	$\Theta\left(n^2\right)$	$\Theta\left(n^3\right)$

- The matrices A, B, C are stored in row-major order
- **MM-ikj** exploits spatial cache locality

MM Loops: Complexity

Algorithm	$T_1(n)$	$T_\infty(n)$	$E_1(n)$	$S_\infty(n)$	$Q_1(n)$
MM-ijk	$\Theta\left(n^3\right)$	$\Theta\left(n^3\right)$	$\Theta\left(1\right)$	$\Theta\left(n^2\right)$	$\Theta\left(n^3\right)$
MM-ikj	$\Theta\left(n^3\right)$	$\Theta\left(n^3\right)$	$\Theta\left(1\right)$	$\Theta\left(n^2\right)$	$\Theta\left(\frac{n^3}{B} + n^2\right)$
MM-jik	$\Theta\left(n^3\right)$	$\Theta\left(n^3\right)$	$\Theta\left(1\right)$	$\Theta\left(n^2\right)$	$\Theta\left(n^3\right)$
MM-jki	$\Theta\left(n^3\right)$	$\Theta\left(n^3\right)$	$\Theta\left(1\right)$	$\Theta\left(n^2\right)$	$\Theta\left(n^3\right)$
MM-kij	$\Theta\left(n^3\right)$	$\Theta\left(n^3\right)$	$\Theta\left(1\right)$	$\Theta\left(n^2\right)$	$\Theta\left(n^3\right)$
MM-kji	$\Theta\left(n^3\right)$	$\Theta\left(n^3\right)$	$\Theta\left(1\right)$	$\Theta\left(n^2\right)$	$\Theta\left(n^3\right)$

- The matrices A, B, C are stored in row-major order
- **MM-ikj** exploits spatial cache locality
- If we store the B matrix in column-major order, then which of the 6 algorithms will have better cache complexity?

MM Loops: Complexity

Algorithm	$T_1(n)$	$T_\infty(n)$	$E_1(n)$	$S_\infty(n)$	$Q_1(n)$
MM-ijk	$\Theta\left(n^3\right)$	$\Theta\left(n^3\right)$	$\Theta\left(1\right)$	$\Theta\left(n^2\right)$	$\Theta\left(n^3\right)$
MM-ikj	$\Theta\left(n^3\right)$	$\Theta\left(n^3\right)$	$\Theta\left(1\right)$	$\Theta\left(n^2\right)$	$\Theta\left(\frac{n^3}{B} + n^2\right)$
MM-jik	$\Theta\left(n^3\right)$	$\Theta\left(n^3\right)$	$\Theta\left(1\right)$	$\Theta\left(n^2\right)$	$\Theta\left(n^3\right)$
MM-jki	$\Theta\left(n^3\right)$	$\Theta\left(n^3\right)$	$\Theta\left(1\right)$	$\Theta\left(n^2\right)$	$\Theta\left(n^3\right)$
MM-kij	$\Theta\left(n^3\right)$	$\Theta\left(n^3\right)$	$\Theta\left(1\right)$	$\Theta\left(n^2\right)$	$\Theta\left(n^3\right)$
MM-kji	$\Theta\left(n^3\right)$	$\Theta\left(n^3\right)$	$\Theta\left(1\right)$	$\Theta\left(n^2\right)$	$\Theta\left(n^3\right)$

- The matrices A, B, C are stored in row-major order
- **MM-ikj** exploits spatial cache locality
- If we store the B matrix in column-major order, then which of the 6 algorithms will have better cache complexity?
- How can we parallelize the algorithms?

MM Parallel Loops

MM-ijk(A, B, C, n)

1. **parallel** for $i \leftarrow 1$ to n do
2. **parallel** for $j \leftarrow 1$ to n do
3. **parallel** for $k \leftarrow 1$ to n do
4. $C[i,j] \leftarrow C[i,j] + A[i,k] \cdot B[k,j]$

MM-ikj(A, B, C, n)

1. **parallel** for $i \leftarrow 1$ to n do
2. **parallel** for $k \leftarrow 1$ to n do
3. **parallel** for $j \leftarrow 1$ to n do
4. $C[i,j] \leftarrow C[i,j] + A[i,k] \cdot B[k,j]$

MM-jik(A, B, C, n)

1. **parallel** for $j \leftarrow 1$ to n do
2. **parallel** for $i \leftarrow 1$ to n do
3. **parallel** for $k \leftarrow 1$ to n do
4. $C[i,j] \leftarrow C[i,j] + A[i,k] \cdot B[k,j]$

MM-jki(A, B, C, n)

1. **parallel** for $j \leftarrow 1$ to n do
2. **parallel** for $k \leftarrow 1$ to n do
3. **parallel** for $i \leftarrow 1$ to n do
4. $C[i,j] \leftarrow C[i,j] + A[i,k] \cdot B[k,j]$

MM-kij(A, B, C, n)

1. **parallel** for $k \leftarrow 1$ to n do
2. **parallel** for $i \leftarrow 1$ to n do
3. **parallel** for $j \leftarrow 1$ to n do
4. $C[i,j] \leftarrow C[i,j] + A[i,k] \cdot B[k,j]$

MM-kji(A, B, C, n)

1. **parallel** for $k \leftarrow 1$ to n do
2. **parallel** for $j \leftarrow 1$ to n do
3. **parallel** for $i \leftarrow 1$ to n do
4. $C[i,j] \leftarrow C[i,j] + A[i,k] \cdot B[k,j]$

- All the three loops are parallelized
- Are these algorithms correct?

MM Parallel Loops

MM-ijk(A, B, C, n)

1. **parallel** for $i \leftarrow 1$ to n do
2. **parallel** for $j \leftarrow 1$ to n do
3. **parallel** for $k \leftarrow 1$ to n do
4. $C[i,j] \leftarrow C[i,j] + A[i,k] \cdot B[k,j]$

MM-ikj(A, B, C, n)

1. **parallel** for $i \leftarrow 1$ to n do
2. **parallel** for $k \leftarrow 1$ to n do
3. **parallel** for $j \leftarrow 1$ to n do
4. $C[i,j] \leftarrow C[i,j] + A[i,k] \cdot B[k,j]$

MM-jik(A, B, C, n)

1. **parallel** for $j \leftarrow 1$ to n do
2. **parallel** for $i \leftarrow 1$ to n do
3. **parallel** for $k \leftarrow 1$ to n do
4. $C[i,j] \leftarrow C[i,j] + A[i,k] \cdot B[k,j]$

MM-jki(A, B, C, n)

1. **parallel** for $j \leftarrow 1$ to n do
2. **parallel** for $k \leftarrow 1$ to n do
3. **parallel** for $i \leftarrow 1$ to n do
4. $C[i,j] \leftarrow C[i,j] + A[i,k] \cdot B[k,j]$

MM-kij(A, B, C, n)

1. **parallel** for $k \leftarrow 1$ to n do
2. **parallel** for $i \leftarrow 1$ to n do
3. **parallel** for $j \leftarrow 1$ to n do
4. $C[i,j] \leftarrow C[i,j] + A[i,k] \cdot B[k,j]$

MM-kji(A, B, C, n)

1. **parallel** for $k \leftarrow 1$ to n do
2. **parallel** for $j \leftarrow 1$ to n do
3. **parallel** for $i \leftarrow 1$ to n do
4. $C[i,j] \leftarrow C[i,j] + A[i,k] \cdot B[k,j]$

- All the three loops are parallelized
- Are these algorithms correct? No! (race conditions)

MM Parallel Loops

MM-ijk(A, B, C, n)

1. **parallel** for $i \leftarrow 1$ to n do
2. **parallel** for $j \leftarrow 1$ to n do
3. **parallel** for $k \leftarrow 1$ to n do
4. $C[i,j] \leftarrow C[i,j] + A[i,k] \cdot B[k,j]$

MM-ikj(A, B, C, n)

1. **parallel** for $i \leftarrow 1$ to n do
2. **parallel** for $k \leftarrow 1$ to n do
3. **parallel** for $j \leftarrow 1$ to n do
4. $C[i,j] \leftarrow C[i,j] + A[i,k] \cdot B[k,j]$

MM-jik(A, B, C, n)

1. **parallel** for $j \leftarrow 1$ to n do
2. **parallel** for $i \leftarrow 1$ to n do
3. **parallel** for $k \leftarrow 1$ to n do
4. $C[i,j] \leftarrow C[i,j] + A[i,k] \cdot B[k,j]$

MM-jki(A, B, C, n)

1. **parallel** for $j \leftarrow 1$ to n do
2. **parallel** for $k \leftarrow 1$ to n do
3. **parallel** for $i \leftarrow 1$ to n do
4. $C[i,j] \leftarrow C[i,j] + A[i,k] \cdot B[k,j]$

MM-kij(A, B, C, n)

1. **parallel** for $k \leftarrow 1$ to n do
2. **parallel** for $i \leftarrow 1$ to n do
3. **parallel** for $j \leftarrow 1$ to n do
4. $C[i,j] \leftarrow C[i,j] + A[i,k] \cdot B[k,j]$

MM-kji(A, B, C, n)

1. **parallel** for $k \leftarrow 1$ to n do
2. **parallel** for $j \leftarrow 1$ to n do
3. **parallel** for $i \leftarrow 1$ to n do
4. $C[i,j] \leftarrow C[i,j] + A[i,k] \cdot B[k,j]$

- All the three loops are parallelized
- Are these algorithms correct? No! (race conditions)
- How can we avoid the race conditions?

MM Parallel Loops

MM-ijk(A, B, C, n)

1. **parallel** for $i \leftarrow 1$ to n do
2.   **parallel** for $j \leftarrow 1$ to n do
3.   for $k \leftarrow 1$ to n do
4. $C[i,j] \leftarrow C[i,j] + A[i,k] \cdot B[k,j]$

MM-ikj(A, B, C, n)

1. **parallel** for $i \leftarrow 1$ to n do
2.   for $k \leftarrow 1$ to n do
3.   **parallel** for $j \leftarrow 1$ to n do
4. $C[i,j] \leftarrow C[i,j] + A[i,k] \cdot B[k,j]$

MM-jik(A, B, C, n)

1. **parallel** for $j \leftarrow 1$ to n do
2.   **parallel** for $i \leftarrow 1$ to n do
3.   for $k \leftarrow 1$ to n do
4. $C[i,j] \leftarrow C[i,j] + A[i,k] \cdot B[k,j]$

MM-jki(A, B, C, n)

1. **parallel** for $j \leftarrow 1$ to n do
2.   for $k \leftarrow 1$ to n do
3.   **parallel** for $i \leftarrow 1$ to n do
4. $C[i,j] \leftarrow C[i,j] + A[i,k] \cdot B[k,j]$

MM-kij(A, B, C, n)

1. for $k \leftarrow 1$ to n do
2.   **parallel** for $i \leftarrow 1$ to n do
3.   **parallel** for $j \leftarrow 1$ to n do
4. $C[i,j] \leftarrow C[i,j] + A[i,k] \cdot B[k,j]$

MM-kji(A, B, C, n)

1. for $k \leftarrow 1$ to n do
2.   **parallel** for $j \leftarrow 1$ to n do
3.   **parallel** for $i \leftarrow 1$ to n do
4. $C[i,j] \leftarrow C[i,j] + A[i,k] \cdot B[k,j]$

- The k-loop is serialized
- Are these algorithms correct?

MM Parallel Loops

MM-ijk(A, B, C, n)

parallel for $i \leftarrow 1$ to n do
  parallel for $j \leftarrow 1$ to n do
    for $k \leftarrow 1$ to n do
      $C[i,j] \leftarrow C[i,j] + A[i,k] \cdot B[k,j]$

MM-ikj(A, B, C, n)

parallel for $i \leftarrow 1$ to n do
  for $k \leftarrow 1$ to n do
    parallel for $j \leftarrow 1$ to n do
      $C[i,j] \leftarrow C[i,j] + A[i,k] \cdot B[k,j]$

MM-jik(A, B, C, n)

parallel for $j \leftarrow 1$ to n do
  parallel for $i \leftarrow 1$ to n do
    for $k \leftarrow 1$ to n do
      $C[i,j] \leftarrow C[i,j] + A[i,k] \cdot B[k,j]$

MM-jki(A, B, C, n)

parallel for $j \leftarrow 1$ to n do
  for $k \leftarrow 1$ to n do
    parallel for $i \leftarrow 1$ to n do
      $C[i,j] \leftarrow C[i,j] + A[i,k] \cdot B[k,j]$

MM-kij(A, B, C, n)

for $k \leftarrow 1$ to n do
  parallel for $i \leftarrow 1$ to n do
    parallel for $j \leftarrow 1$ to n do
      $C[i,j] \leftarrow C[i,j] + A[i,k] \cdot B[k,j]$

MM-kji(A, B, C, n)

for $k \leftarrow 1$ to n do
  parallel for $j \leftarrow 1$ to n do
    parallel for $i \leftarrow 1$ to n do
      $C[i,j] \leftarrow C[i,j] + A[i,k] \cdot B[k,j]$

- The k-loop is serialized
- Are these algorithms correct? Yes! (no race conditions)

MM Parallel Loops: Complexity

Algorithm	$T_1(n)$	$T_\infty(n)$	$E_1(n)$	$S_\infty(n)$	$Q_1(n)$
MM-ijk	$\Theta\left(n^3\right)$	$\Theta\left(n\right)$	$\Theta\left(\log n\right)$	$\Theta\left(n^2\right)$	$\Theta\left(n^3\right)$
MM-ikj	$\Theta\left(n^3\right)$	$\Theta\left(n\right)$	$\Theta\left(\log n\right)$	$\Theta\left(n^2\right)$	$\Theta\left(\frac{n^3}{B}+n^2\right)$
MM-jik	$\Theta\left(n^3\right)$	$\Theta\left(n\right)$	$\Theta\left(\log n\right)$	$\Theta\left(n^2\right)$	$\Theta\left(n^3\right)$
MM-jki	$\Theta\left(n^3\right)$	$\Theta\left(n\right)$	$\Theta\left(\log n\right)$	$\Theta\left(n^2\right)$	$\Theta\left(n^3\right)$
MM-kij	$\Theta\left(n^3\right)$	$\Theta\left(n\right)$	$\Theta\left(\log n\right)$	$\Theta\left(n^2\right)$	$\Theta\left(n^3\right)$
MM-kji	$\Theta\left(n^3\right)$	$\Theta\left(n\right)$	$\Theta\left(\log n\right)$	$\Theta\left(n^2\right)$	$\Theta\left(n^3\right)$

- How can we compute $S_p(n)$?

MM Parallel Loops: Complexity

Algorithm	$T_1(n)$	$T_\infty(n)$	$E_1(n)$	$S_\infty(n)$	$Q_1(n)$
MM-ijk	$\Theta\left(n^3\right)$	$\Theta\left(n\right)$	$\Theta\left(\log n\right)$	$\Theta\left(n^2\right)$	$\Theta\left(n^3\right)$
MM-ikj	$\Theta\left(n^3\right)$	$\Theta\left(n\right)$	$\Theta\left(\log n\right)$	$\Theta\left(n^2\right)$	$\Theta\left(\frac{n^3}{B} + n^2\right)$
MM-jik	$\Theta\left(n^3\right)$	$\Theta\left(n\right)$	$\Theta\left(\log n\right)$	$\Theta\left(n^2\right)$	$\Theta\left(n^3\right)$
MM-jki	$\Theta\left(n^3\right)$	$\Theta\left(n\right)$	$\Theta\left(\log n\right)$	$\Theta\left(n^2\right)$	$\Theta\left(n^3\right)$
MM-kij	$\Theta\left(n^3\right)$	$\Theta\left(n\right)$	$\Theta\left(\log n\right)$	$\Theta\left(n^2\right)$	$\Theta\left(n^3\right)$
MM-kji	$\Theta\left(n^3\right)$	$\Theta\left(n\right)$	$\Theta\left(\log n\right)$	$\Theta\left(n^2\right)$	$\Theta\left(n^3\right)$

- How can we compute $S_p(n)$?
- How can we improve parallelism?

MM Parallel Loops: Complexity

Algorithm	$T_1(n)$	$T_\infty(n)$	$E_1(n)$	$S_\infty(n)$	$Q_1(n)$
MM-ijk	$\Theta\left(n^3\right)$	$\Theta\left(n\right)$	$\Theta\left(\log n\right)$	$\Theta\left(n^2\right)$	$\Theta\left(n^3\right)$
MM-ikj	$\Theta\left(n^3\right)$	$\Theta\left(n\right)$	$\Theta\left(\log n\right)$	$\Theta\left(n^2\right)$	$\Theta\left(\frac{n^3}{B} + n^2\right)$
MM-jik	$\Theta\left(n^3\right)$	$\Theta\left(n\right)$	$\Theta\left(\log n\right)$	$\Theta\left(n^2\right)$	$\Theta\left(n^3\right)$
MM-jki	$\Theta\left(n^3\right)$	$\Theta\left(n\right)$	$\Theta\left(\log n\right)$	$\Theta\left(n^2\right)$	$\Theta\left(n^3\right)$
MM-kij	$\Theta\left(n^3\right)$	$\Theta\left(n\right)$	$\Theta\left(\log n\right)$	$\Theta\left(n^2\right)$	$\Theta\left(n^3\right)$
MM-kji	$\Theta\left(n^3\right)$	$\Theta\left(n\right)$	$\Theta\left(\log n\right)$	$\Theta\left(n^2\right)$	$\Theta\left(n^3\right)$

- How can we compute $S_p(n)$?
- How can we improve parallelism?
  Reduction!

MM Parallel Loops with Reduction

- The $C[i,j]$ shared variables are defined as reducers

MM-ijk(A,B,C,n)

1. **parallel** for $i \leftarrow 1$ to n do
2. **parallel** for $j \leftarrow 1$ to n do
3. **reduce** for $k \leftarrow 1$ to n do
4. $C[i,j] \leftarrow C[i,j] + A[i,k] \cdot B[k,j]$

MM-ikj(A,B,C,n)

1. **parallel** for $i \leftarrow 1$ to n do
2. **reduce** for $k \leftarrow 1$ to n do
3. **parallel** for $j \leftarrow 1$ to n do
4. $C[i,j] \leftarrow C[i,j] + A[i,k] \cdot B[k,j]$

MM-jik(A,B,C,n)

1. **parallel** for $j \leftarrow 1$ to n do
2. **parallel** for $i \leftarrow 1$ to n do
3. **reduce** for $k \leftarrow 1$ to n do
4. $C[i,j] \leftarrow C[i,j] + A[i,k] \cdot B[k,j]$

MM-jki(A,B,C,n)

1. **parallel** for $j \leftarrow 1$ to n do
2. **reduce** for $k \leftarrow 1$ to n do
3. **parallel** for $i \leftarrow 1$ to n do
4. $C[i,j] \leftarrow C[i,j] + A[i,k] \cdot B[k,j]$

MM-kij(A,B,C,n)

1. **reduce** for $k \leftarrow 1$ to n do
2. **parallel** for $i \leftarrow 1$ to n do
3. **parallel** for $j \leftarrow 1$ to n do
4. $C[i,j] \leftarrow C[i,j] + A[i,k] \cdot B[k,j]$

MM-kji(A,B,C,n)

1. **reduce** for $k \leftarrow 1$ to n do
2. **parallel** for $j \leftarrow 1$ to n do
3. **parallel** for $i \leftarrow 1$ to n do
4. $C[i,j] \leftarrow C[i,j] + A[i,k] \cdot B[k,j]$

- The k-loop is reduced
- Are these algorithms correct?

MM Parallel Loops with Reduction

- The $C[i,j]$ shared variables are defined as reducers

MM-ijk(A,B,C,n)

1. **parallel** for $i \leftarrow 1$ to n do
2. **parallel** for $j \leftarrow 1$ to n do
3. **reduce** for $k \leftarrow 1$ to n do
4. $C[i,j] \leftarrow C[i,j] + A[i,k] \cdot B[k,j]$

MM-ikj(A,B,C,n)

1. **parallel** for $i \leftarrow 1$ to n do
2. **reduce** for $k \leftarrow 1$ to n do
3. **parallel** for $j \leftarrow 1$ to n do
4. $C[i,j] \leftarrow C[i,j] + A[i,k] \cdot B[k,j]$

MM-jik(A,B,C,n)

1. **parallel** for $j \leftarrow 1$ to n do
2. **parallel** for $i \leftarrow 1$ to n do
3. **reduce** for $k \leftarrow 1$ to n do
4. $C[i,j] \leftarrow C[i,j] + A[i,k] \cdot B[k,j]$

MM-jki(A,B,C,n)

1. **parallel** for $j \leftarrow 1$ to n do
2. **reduce** for $k \leftarrow 1$ to n do
3. **parallel** for $i \leftarrow 1$ to n do
4. $C[i,j] \leftarrow C[i,j] + A[i,k] \cdot B[k,j]$

MM-kij(A,B,C,n)

1. **reduce** for $k \leftarrow 1$ to n do
2. **parallel** for $i \leftarrow 1$ to n do
3. **parallel** for $j \leftarrow 1$ to n do
4. $C[i,j] \leftarrow C[i,j] + A[i,k] \cdot B[k,j]$

MM-kji(A,B,C,n)

1. **reduce** for $k \leftarrow 1$ to n do
2. **parallel** for $j \leftarrow 1$ to n do
3. **parallel** for $i \leftarrow 1$ to n do
4. $C[i,j] \leftarrow C[i,j] + A[i,k] \cdot B[k,j]$

- The k-loop is reduced
- Are these algorithms correct? Yes! (no race conditions)

MM Parallel Loops with Reduction: Complexity

Algorithm	$T_1(n)$	$T_\infty(n)$	$E_1(n)$	$S_\infty(n)$	$Q_1(n)$
MM-ijk	$\Theta\left(n^3\right)$	$\Theta\left(\log n\right)$	$\Theta\left(\log n\right)$	$\Theta\left(n^3\right)$	$\Theta\left(n^3\right)$
MM-ikj	$\Theta\left(n^3\right)$	$\Theta\left(\log n\right)$	$\Theta\left(\log n\right)$	$\Theta\left(n^3\right)$	$\Theta\left(\frac{n^3}{B} + n^2\right)$
MM-jik	$\Theta\left(n^3\right)$	$\Theta\left(\log n\right)$	$\Theta\left(\log n\right)$	$\Theta\left(n^3\right)$	$\Theta\left(n^3\right)$
MM-jki	$\Theta\left(n^3\right)$	$\Theta\left(\log n\right)$	$\Theta\left(\log n\right)$	$\Theta\left(n^3\right)$	$\Theta\left(n^3\right)$
MM-kij	$\Theta\left(n^3\right)$	$\Theta\left(\log n\right)$	$\Theta\left(\log n\right)$	$\Theta\left(n^3\right)$	$\Theta\left(n^3\right)$
MM-kji	$\Theta\left(n^3\right)$	$\Theta\left(\log n\right)$	$\Theta\left(\log n\right)$	$\Theta\left(n^3\right)$	$\Theta\left(n^3\right)$

MM Parallel Loops with Reduction: Complexity

Algorithm	$T_1(n)$	$T_\infty(n)$	$E_1(n)$	$S_\infty(n)$	$Q_1(n)$
MM-ijk	$\Theta\left(n^3\right)$	$\Theta\left(\log n\right)$	$\Theta\left(\log n\right)$	$\Theta\left(n^3\right)$	$\Theta\left(n^3\right)$
MM-ikj	$\Theta\left(n^3\right)$	$\Theta\left(\log n\right)$	$\Theta\left(\log n\right)$	$\Theta\left(n^3\right)$	$\Theta\left(\frac{n^3}{B}+n^2\right)$
MM-jik	$\Theta\left(n^3\right)$	$\Theta\left(\log n\right)$	$\Theta\left(\log n\right)$	$\Theta\left(n^3\right)$	$\Theta\left(n^3\right)$
MM-jki	$\Theta\left(n^3\right)$	$\Theta\left(\log n\right)$	$\Theta\left(\log n\right)$	$\Theta\left(n^3\right)$	$\Theta\left(n^3\right)$
MM-kij	$\Theta\left(n^3\right)$	$\Theta\left(\log n\right)$	$\Theta\left(\log n\right)$	$\Theta\left(n^3\right)$	$\Theta\left(n^3\right)$
MM-kji	$\Theta\left(n^3\right)$	$\Theta\left(\log n\right)$	$\Theta\left(\log n\right)$	$\Theta\left(n^3\right)$	$\Theta\left(n^3\right)$

Advantages	Disadvantages
Minimum depth	Maximum space
Easy to implement	Lack temporal cache locality

MM Parallel Loops with Reduction: Complexity

Algorithm	$T_1(n)$	$T_\infty(n)$	$E_1(n)$	$S_\infty(n)$	$Q_1(n)$
MM-ijk	$\Theta\left(n^3\right)$	$\Theta\left(\log n\right)$	$\Theta\left(\log n\right)$	$\Theta\left(n^3\right)$	$\Theta\left(n^3\right)$
MM-ikj	$\Theta\left(n^3\right)$	$\Theta\left(\log n\right)$	$\Theta\left(\log n\right)$	$\Theta\left(n^3\right)$	$\Theta\left(\frac{n^3}{B} + n^2\right)$
MM-jik	$\Theta\left(n^3\right)$	$\Theta\left(\log n\right)$	$\Theta\left(\log n\right)$	$\Theta\left(n^3\right)$	$\Theta\left(n^3\right)$
MM-jki	$\Theta\left(n^3\right)$	$\Theta\left(\log n\right)$	$\Theta\left(\log n\right)$	$\Theta\left(n^3\right)$	$\Theta\left(n^3\right)$
MM-kij	$\Theta\left(n^3\right)$	$\Theta\left(\log n\right)$	$\Theta\left(\log n\right)$	$\Theta\left(n^3\right)$	$\Theta\left(n^3\right)$
MM-kji	$\Theta\left(n^3\right)$	$\Theta\left(\log n\right)$	$\Theta\left(\log n\right)$	$\Theta\left(n^3\right)$	$\Theta\left(n^3\right)$

Advantages	Disadvantages
Minimum depth	Maximum space
Easy to implement	Lack temporal cache locality

- How can we exploit the temporal data locality?

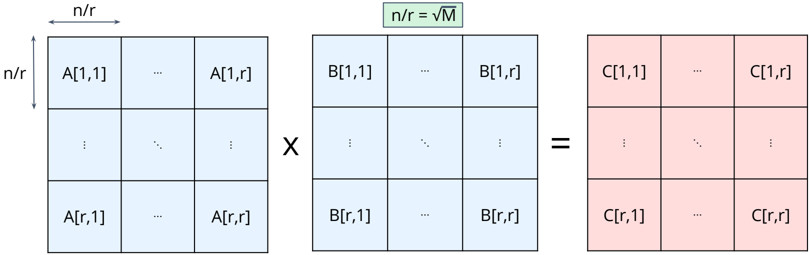
MM Parallel Loops with Reduction: Complexity

Algorithm	$T_1(n)$	$T_\infty(n)$	$E_1(n)$	$S_\infty(n)$	$Q_1(n)$
MM-ijk	$\Theta\left(n^3\right)$	$\Theta\left(\log n\right)$	$\Theta\left(\log n\right)$	$\Theta\left(n^3\right)$	$\Theta\left(n^3\right)$
MM-ikj	$\Theta\left(n^3\right)$	$\Theta\left(\log n\right)$	$\Theta\left(\log n\right)$	$\Theta\left(n^3\right)$	$\Theta\left(\frac{n^3}{B} + n^2\right)$
MM-jik	$\Theta\left(n^3\right)$	$\Theta\left(\log n\right)$	$\Theta\left(\log n\right)$	$\Theta\left(n^3\right)$	$\Theta\left(n^3\right)$
MM-jki	$\Theta\left(n^3\right)$	$\Theta\left(\log n\right)$	$\Theta\left(\log n\right)$	$\Theta\left(n^3\right)$	$\Theta\left(n^3\right)$
MM-kij	$\Theta\left(n^3\right)$	$\Theta\left(\log n\right)$	$\Theta\left(\log n\right)$	$\Theta\left(n^3\right)$	$\Theta\left(n^3\right)$
MM-kji	$\Theta\left(n^3\right)$	$\Theta\left(\log n\right)$	$\Theta\left(\log n\right)$	$\Theta\left(n^3\right)$	$\Theta\left(n^3\right)$

Advantages	Disadvantages
Minimum depth	Maximum space
Easy to implement	Lack temporal cache locality

- How can we exploit the temporal data locality? Localized computations!

MM Tiled Loops: Core Idea



Core idea

- Let $r = \frac{n}{\sqrt{M}}$. Split the three matrices into $r \times r$ tiles/blocks, each tile of size $\frac{n}{r} \times \frac{n}{r}$
- Load the three tiles $A[I, K], B[K, J]$, and $C[I, J]$, each of size $\frac{n}{r} \times \frac{n}{r}$, into cache of size $\Theta(M)$
- Compute $C[I, J]$

MM Tiled Loops

MM-Tiled(A, B, C, n, M)

1. $r \leftarrow n/\sqrt{M}$
2. for $I \leftarrow 1$ to r do
3. for $J \leftarrow 1$ to r do
4. $C[I, J] \leftarrow \{0\}$
5. for $K \leftarrow 1$ to r do
6. **MM-Loop**$(A[I, K], B[K, J], C[I, J], n/r)$
   $\triangleright\ C[I, J] \leftarrow C[I, J] + A[I, K] \cdot B[K, J]$

MM Tiled Loops

MM-Tiled(A, B, C, n, M)

1. $r \leftarrow n/\sqrt{M}$
2. for $I \leftarrow 1$ to r do
3. for $J \leftarrow 1$ to r do
4. $C[I, J] \leftarrow \{0\}$
5. for $K \leftarrow 1$ to r do
6. **MM-Loop**$(A[I, K], B[K, J], C[I, J], n/r)$
   $\triangleright$ $C[I, J] \leftarrow C[I, J] + A[I, K] \cdot B[K, J]$

Correctness

- Similar to our previous proofs

MM Tiled Loops

MM-Tiled(A, B, C, n, M)

1. $r \leftarrow n/\sqrt{M}$
2. for $I \leftarrow 1$ to r do
3. for $J \leftarrow 1$ to r do
4. $C[I, J] \leftarrow \{0\}$
5. for $K \leftarrow 1$ to r do
6. **MM-Loop**$(A[I, K], B[K, J], C[I, J], n/r)$

$\triangleright\ C[I, J] \leftarrow C[I, J] + A[I, K] \cdot B[K, J]$

Correctness

- Similar to our previous proofs

$T_1(n)$	$T_\infty(n)$	$E_1(n)$	$S_\infty(n)$	$Q_1(n)$
$\Theta\left(n^3\right)$	$\Theta\left(n^3\right)$	$\Theta\left(1\right)$	$\Theta\left(n^2\right)$	$\Theta\left(\frac{n^3}{B\sqrt{M}} + \frac{n^2}{B} + n\right)$

MM Tiled Loops

MM-Tiled(A, B, C, n, M)

1. $r \leftarrow n/\sqrt{M}$
2. for $I \leftarrow 1$ to r do
3. for $J \leftarrow 1$ to r do
4. $C[I, J] \leftarrow \{0\}$
5. for $K \leftarrow 1$ to r do
6. **MM-Loop**$(A[I, K], B[K, J], C[I, J], n/r)$
   $\triangleright\ C[I, J] \leftarrow C[I, J] + A[I, K] \cdot B[K, J]$

Correctness

- Similar to our previous proofs

$T_1(n)$	$T_\infty(n)$	$E_1(n)$	$S_\infty(n)$	$Q_1(n)$
$\Theta\left(n^3\right)$	$\Theta\left(n^3\right)$	$\Theta\left(1\right)$	$\Theta\left(n^2\right)$	$\Theta\left(\frac{n^3}{B\sqrt{M}} + \frac{n^2}{B} + n\right)$

- How can we parallelize this algorithm?

MM Parallel Tiled Loops

MM-Tiled-Parallel(A, B, C, n, M)

```
r ← n/√M
parallel for I ← 1 to r do
  parallel for J ← 1 to r do
    C[I, J] ← {0}
    for K ← 1 to r do
      MM-Loop-Parallel(A[I, K], B[K, J], C[I, J], n/r)
                                  ▷ C[I, J] ← C[I, J] + A[I, K] · B[K, J]
```

MM Parallel Tiled Loops

MM-Tiled-Parallel(A, B, C, n, M)

1. $r \leftarrow n/\sqrt{M}$
2. **parallel** for $I \leftarrow 1$ to r do
3. **parallel** for $J \leftarrow 1$ to r do
4. $C[I, J] \leftarrow \{0\}$
5. for $K \leftarrow 1$ to r do
6. **MM-Loop-Parallel**$(A[I, K], B[K, J], C[I, J], n/r)$
   $\triangleright\ C[I, J] \leftarrow C[I, J] + A[I, K] \cdot B[K, J]$

$T_1(n)$	$T_\infty(n)$	$E_1(n)$	$S_\infty(n)$	$Q_1(n)$
$\Theta\left(n^3\right)$	$\Theta\left(n\right)$	$\Theta\left(1\right)$	$\Theta\left(n^2\right)$	$\Theta\left(\frac{n^3}{B\sqrt{M}} + \frac{n^2}{B} + n\right)$

MM Parallel Tiled Loops

MM-Tiled-Parallel(A, B, C, n, M)

1. $r \leftarrow n/\sqrt{M}$
2. **parallel** for $I \leftarrow 1$ to r do
3. **parallel** for $J \leftarrow 1$ to r do
4. $C[I, J] \leftarrow \{0\}$
5. for $K \leftarrow 1$ to r do
6. **MM-Loop-Parallel**$(A[I, K], B[K, J], C[I, J], n/r)$
   $\triangleright\ C[I, J] \leftarrow C[I, J] + A[I, K] \cdot B[K, J]$

$T_1(n)$	$T_\infty(n)$	$E_1(n)$	$S_\infty(n)$	$Q_1(n)$
$\Theta\left(n^3\right)$	$\Theta\left(n\right)$	$\Theta\left(1\right)$	$\Theta\left(n^2\right)$	$\Theta\left(\frac{n^3}{B\sqrt{M}} + \frac{n^2}{B} + n\right)$

- The $\langle I, J, K\rangle$ and $\langle i, j, k\rangle$ loops can be permuted in $6 \times 6 = 36$ ways. Which is the fastest among these parallel tiled algorithms?

MM Parallel Tiled Loops

MM-Tiled-Parallel(A, B, C, n, M)

1. $r \leftarrow n/\sqrt{M}$
2. **parallel** for $I \leftarrow 1$ to r do
3. **parallel** for $J \leftarrow 1$ to r do
4. $C[I, J] \leftarrow \{0\}$
5. for $K \leftarrow 1$ to r do
6. **MM-Loop-Parallel**$(A[I, K], B[K, J], C[I, J], n/r)$
   $\triangleright\ C[I, J] \leftarrow C[I, J] + A[I, K] \cdot B[K, J]$

$T_1(n)$	$T_\infty(n)$	$E_1(n)$	$S_\infty(n)$	$Q_1(n)$
$\Theta\left(n^3\right)$	$\Theta\left(n\right)$	$\Theta\left(1\right)$	$\Theta\left(n^2\right)$	$\Theta\left(\frac{n^3}{B\sqrt{M}} + \frac{n^2}{B} + n\right)$

- The $\langle I, J, K\rangle$ and $\langle i, j, k\rangle$ loops can be permuted in $6 \times 6 = 36$ ways. Which is the fastest among these parallel tiled algorithms?
- Tiled algorithms are not easily portable across machines
  How can we get more portable algorithms?

MM Parallel Tiled Loops

MM-Tiled-Parallel(A, B, C, n, M)

1. $r \leftarrow n/\sqrt{M}$
2. **parallel** for $I \leftarrow 1$ to r do
3.   **parallel** for $J \leftarrow 1$ to r do
4. $C[I, J] \leftarrow \{0\}$
5.   for $K \leftarrow 1$ to r do
6.   **MM-Loop-Parallel**$(A[I, K], B[K, J], C[I, J], n/r)$
   $\triangleright\ C[I, J] \leftarrow C[I, J] + A[I, K] \cdot B[K, J]$

$T_1(n)$	$T_\infty(n)$	$E_1(n)$	$S_\infty(n)$	$Q_1(n)$
$\Theta(n^3)$	$\Theta(n)$	$\Theta(1)$	$\Theta(n^2)$	$\Theta\left(\frac{n^3}{B\sqrt{M}} + \frac{n^2}{B} + n\right)$

- The $\langle I, J, K\rangle$ and $\langle i, j, k\rangle$ loops can be permuted in $6 \times 6 = 36$ ways. Which is the fastest among these parallel tiled algorithms?
- Tiled algorithms are not easily portable across machines
  How can we get more portable algorithms?
  Divide-and-conquer!

MM D&C

Block size: $n/2 \times n/2$

$$\begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \times \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}$$

$$= \begin{bmatrix} A_{11}B_{11} + A_{12}B_{21} & A_{11}B_{12} + A_{12}B_{22} \\ A_{21}B_{11} + A_{22}B_{21} & A_{21}B_{12} + A_{22}B_{22} \end{bmatrix}$$

MM D&C

MM(A, B, C, n)

if $n = 1$ then
  MM-Loop(A, B, C, n)
else
  MM$(A_{11}, B_{11}, C_{11}, n/2)$
  MM$(A_{12}, B_{21}, C_{11}, n/2)$
  MM$(A_{11}, B_{12}, C_{12}, n/2)$
  MM$(A_{12}, B_{22}, C_{12}, n/2)$
  MM$(A_{21}, B_{11}, C_{21}, n/2)$
  MM$(A_{22}, B_{21}, C_{21}, n/2)$
  MM$(A_{21}, B_{12}, C_{22}, n/2)$
  MM$(A_{22}, B_{22}, C_{22}, n/2)$

Complexity

$$T_1(n) = \begin{cases} \Theta(1) & \text{if } n = 1, \\ 8T_1(\frac{n}{2}) + \Theta(1) & \text{if } n > 1. \end{cases}$$

$$T_\infty(n) = \begin{cases} \Theta(1) & \text{if } n = 1, \\ 8T_\infty(\frac{n}{2}) + \Theta(1) & \text{if } n > 1. \end{cases}$$

$$S_\infty(n) = \begin{cases} \Theta(1) & \text{if } n = 1, \\ 4S_\infty(\frac{n}{2}) + \Theta(1) & \text{if } n > 1. \end{cases}$$

$$Q_1(n) = \begin{cases} \Theta\left(\frac{n^2}{B} + n\right) & \text{if } n^2 \le \alpha M, \\ 8Q_1(\frac{n}{2}) + \Theta(1) & \text{if } n^2 > \alpha M. \end{cases}$$

MM D&C

MM(A, B, C, n)

```
if n = 1 then
    MM-Loop(A, B, C, n)
else
    MM(A11, B11, C11, n/2)
    MM(A12, B21, C11, n/2)
    MM(A11, B12, C12, n/2)
    MM(A12, B22, C12, n/2)
    MM(A21, B11, C21, n/2)
    MM(A22, B21, C21, n/2)
    MM(A21, B12, C22, n/2)
    MM(A22, B22, C22, n/2)
```

Complexity

$$T_1(n) = \begin{cases} \Theta(1) & \text{if } n = 1, \\ 8T_1(\frac{n}{2}) + \Theta(1) & \text{if } n > 1. \end{cases}$$

$$T_\infty(n) = \begin{cases} \Theta(1) & \text{if } n = 1, \\ 8T_\infty(\frac{n}{2}) + \Theta(1) & \text{if } n > 1. \end{cases}$$

$$S_\infty(n) = \begin{cases} \Theta(1) & \text{if } n = 1, \\ 4S_\infty(\frac{n}{2}) + \Theta(1) & \text{if } n > 1. \end{cases}$$

$$Q_1(n) = \begin{cases} \Theta\left(\frac{n^2}{B} + n\right) & \text{if } n^2 \le \alpha M, \\ 8Q_1(\frac{n}{2}) + \Theta(1) & \text{if } n^2 > \alpha M. \end{cases}$$

$T_1(n)$	$T_\infty(n)$	$E_1(n)$	$S_\infty(n)$	$Q_1(n)$
$\Theta(n^3)$	$\Theta(n^3)$	$\Theta(\log n)$	$\Theta(n^2)$	$\Theta\left(\frac{n^3}{B\sqrt{M}} + \frac{n^2}{B} + n\right)$

MM D&C

MM(A, B, C, n)

```
if n = 1 then
  MM-Loop(A, B, C, n)
else
  MM(A11, B11, C11, n/2)
  MM(A12, B21, C11, n/2)
  MM(A11, B12, C12, n/2)
  MM(A12, B22, C12, n/2)
  MM(A21, B11, C21, n/2)
  MM(A22, B21, C21, n/2)
  MM(A21, B12, C22, n/2)
  MM(A22, B22, C22, n/2)
```

Complexity

$$T_1(n) = \begin{cases} \Theta(1) & \text{if } n = 1, \\ 8T_1(\frac{n}{2}) + \Theta(1) & \text{if } n > 1. \end{cases}$$

$$T_\infty(n) = \begin{cases} \Theta(1) & \text{if } n = 1, \\ 8T_\infty(\frac{n}{2}) + \Theta(1) & \text{if } n > 1. \end{cases}$$

$$S_\infty(n) = \begin{cases} \Theta(1) & \text{if } n = 1, \\ 4S_\infty(\frac{n}{2}) + \Theta(1) & \text{if } n > 1. \end{cases}$$

$$Q_1(n) = \begin{cases} \Theta\left(\frac{n^2}{B} + n\right) & \text{if } n^2 \leq \alpha M, \\ 8Q_1(\frac{n}{2}) + \Theta(1) & \text{if } n^2 > \alpha M. \end{cases}$$

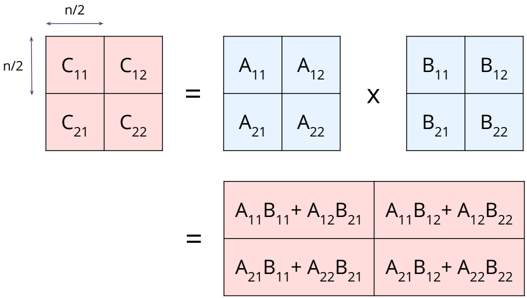
$T_1(n)$	$T_\infty(n)$	$E_1(n)$	$S_\infty(n)$	$Q_1(n)$
$\Theta(n^3)$	$\Theta(n^3)$	$\Theta(\log n)$	$\Theta(n^2)$	$\Theta\left(\frac{n^3}{B\sqrt{M}} + \frac{n^2}{B} + n\right)$

- How can we parallelize this algorithm?

MM Parallel D&C

MM (A, B, C, n)

1. if $n = 1$ then **MM-Loop** (A, B, C, n)
2. else
3. **parallel:** **MM** $(A_{11}, B_{11}, C_{11}, n/2)$, **MM** $(A_{11}, B_{12}, C_{12}, n/2)$
   MM $(A_{21}, B_{11}, C_{21}, n/2)$, **MM** $(A_{21}, B_{12}, C_{22}, n/2)$
4. **parallel:** **MM** $(A_{12}, B_{21}, C_{11}, n/2)$, **MM** $(A_{12}, B_{22}, C_{12}, n/2)$
   MM $(A_{22}, B_{21}, C_{21}, n/2)$, **MM** $(A_{22}, B_{22}, C_{22}, n/2)$



MM Parallel D&C

MM(A, B, C, n)

1. if $n = 1$ then **MM-Loop**(A, B, C, n)
2. else
3. **parallel:** **MM**$(A_{11}, B_{11}, C_{11}, n/2)$, **MM**$(A_{11}, B_{12}, C_{12}, n/2)$
   MM$(A_{21}, B_{11}, C_{21}, n/2)$, **MM**$(A_{21}, B_{12}, C_{22}, n/2)$
4. **parallel:** **MM**$(A_{12}, B_{21}, C_{11}, n/2)$, **MM**$(A_{12}, B_{22}, C_{12}, n/2)$
   MM$(A_{22}, B_{21}, C_{21}, n/2)$, **MM**$(A_{22}, B_{22}, C_{22}, n/2)$

Complexity

$$T_1(n) = \begin{cases} \Theta(1) & \text{if } n = 1, \\ 8T_1(\frac{n}{2}) + \Theta(1) & \text{if } n > 1. \end{cases} \quad S_\infty(n) = \begin{cases} \Theta(1) & \text{if } n = 1, \\ 4S_\infty(\frac{n}{2}) + \Theta(1) & \text{if } n > 1. \end{cases}$$

$$T_\infty(n) = \begin{cases} \Theta(1) & \text{if } n = 1, \\ 2T_\infty(\frac{n}{2}) + \Theta(1) & \text{if } n > 1. \end{cases} \quad Q_1(n) = \begin{cases} \Theta\left(\frac{n^2}{B} + n\right) & \text{if } n^2 \leq \alpha M, \\ 8Q_1(\frac{n}{2}) + \Theta(1) & \text{if } n^2 > \alpha M. \end{cases}$$

MM Parallel D&C

MM(A, B, C, n)

1. if $n = 1$ then **MM-Loop**(A, B, C, n)
2. else
3. **parallel:** **MM**$(A_{11}, B_{11}, C_{11}, n/2)$, **MM**$(A_{11}, B_{12}, C_{12}, n/2)$
   MM$(A_{21}, B_{11}, C_{21}, n/2)$, **MM**$(A_{21}, B_{12}, C_{22}, n/2)$
4. **parallel:** **MM**$(A_{12}, B_{21}, C_{11}, n/2)$, **MM**$(A_{12}, B_{22}, C_{12}, n/2)$
   MM$(A_{22}, B_{21}, C_{21}, n/2)$, **MM**$(A_{22}, B_{22}, C_{22}, n/2)$

Complexity

$$T_1(n) = \begin{cases} \Theta(1) & \text{if } n = 1, \\ 8T_1(\frac{n}{2}) + \Theta(1) & \text{if } n > 1. \end{cases} \quad S_\infty(n) = \begin{cases} \Theta(1) & \text{if } n = 1, \\ 4S_\infty(\frac{n}{2}) + \Theta(1) & \text{if } n > 1. \end{cases}$$

$$T_\infty(n) = \begin{cases} \Theta(1) & \text{if } n = 1, \\ 2T_\infty(\frac{n}{2}) + \Theta(1) & \text{if } n > 1. \end{cases} \quad Q_1(n) = \begin{cases} \Theta\left(\frac{n^2}{B} + n\right) & \text{if } n^2 \le \alpha M, \\ 8Q_1(\frac{n}{2}) + \Theta(1) & \text{if } n^2 > \alpha M. \end{cases}$$

$T_1(n)$	$T_\infty(n)$	$E_1(n)$	$S_\infty(n)$	$Q_1(n)$
$\Theta(n^3)$	$\Theta(n)$	$\Theta(\log n)$	$\Theta(n^2)$	$\Theta\left(\frac{n^3}{B\sqrt{M}} + \frac{n^2}{B} + n\right)$

MM Parallel D&C

MM(A, B, C, n)

1. if $n = 1$ then **MM-Loop**(A, B, C, n)
2. else
3. **parallel:** **MM**$(A_{11}, B_{11}, C_{11}, n/2)$, **MM**$(A_{11}, B_{12}, C_{12}, n/2)$
   MM$(A_{21}, B_{11}, C_{21}, n/2)$, **MM**$(A_{21}, B_{12}, C_{22}, n/2)$
4. **parallel:** **MM**$(A_{12}, B_{21}, C_{11}, n/2)$, **MM**$(A_{12}, B_{22}, C_{12}, n/2)$
   MM$(A_{22}, B_{21}, C_{21}, n/2)$, **MM**$(A_{22}, B_{22}, C_{22}, n/2)$

Complexity

$$T_1(n) = \begin{cases} \Theta(1) & \text{if } n = 1, \\ 8T_1(\frac{n}{2}) + \Theta(1) & \text{if } n > 1. \end{cases} \quad S_\infty(n) = \begin{cases} \Theta(1) & \text{if } n = 1, \\ 4S_\infty(\frac{n}{2}) + \Theta(1) & \text{if } n > 1. \end{cases}$$

$$T_\infty(n) = \begin{cases} \Theta(1) & \text{if } n = 1, \\ 2T_\infty(\frac{n}{2}) + \Theta(1) & \text{if } n > 1. \end{cases} \quad Q_1(n) = \begin{cases} \Theta\left(\frac{n^2}{B} + n\right) & \text{if } n^2 \le \alpha M, \\ 8Q_1(\frac{n}{2}) + \Theta(1) & \text{if } n^2 > \alpha M. \end{cases}$$

$T_1(n)$	$T_\infty(n)$	$E_1(n)$	$S_\infty(n)$	$Q_1(n)$
$\Theta(n^3)$	$\Theta(n)$	$\Theta(\log n)$	$\Theta(n^2)$	$\Theta\left(\frac{n^3}{B\sqrt{M}} + \frac{n^2}{B} + n\right)$

- How can we further improve parallelism?

MM Parallel D&C

MM(A, B, C, n)

1. if $n = 1$ then **MM-Loop**(A, B, C, n)
2. else
3. **parallel:** **MM**$(A_{11}, B_{11}, C_{11}, n/2)$, **MM**$(A_{11}, B_{12}, C_{12}, n/2)$
   MM$(A_{21}, B_{11}, C_{21}, n/2)$, **MM**$(A_{21}, B_{12}, C_{22}, n/2)$
4. **parallel:** **MM**$(A_{12}, B_{21}, C_{11}, n/2)$, **MM**$(A_{12}, B_{22}, C_{12}, n/2)$
   MM$(A_{22}, B_{21}, C_{21}, n/2)$, **MM**$(A_{22}, B_{22}, C_{22}, n/2)$

Complexity

$$T_1(n) = \begin{cases} \Theta(1) & \text{if } n = 1, \\ 8T_1(\frac{n}{2}) + \Theta(1) & \text{if } n > 1. \end{cases} \quad S_\infty(n) = \begin{cases} \Theta(1) & \text{if } n = 1, \\ 4S_\infty(\frac{n}{2}) + \Theta(1) & \text{if } n > 1. \end{cases}$$

$$T_\infty(n) = \begin{cases} \Theta(1) & \text{if } n = 1, \\ 2T_\infty(\frac{n}{2}) + \Theta(1) & \text{if } n > 1. \end{cases} \quad Q_1(n) = \begin{cases} \Theta\left(\frac{n^2}{B} + n\right) & \text{if } n^2 \le \alpha M, \\ 8Q_1(\frac{n}{2}) + \Theta(1) & \text{if } n^2 > \alpha M. \end{cases}$$

$T_1(n)$	$T_\infty(n)$	$E_1(n)$	$S_\infty(n)$	$Q_1(n)$
$\Theta(n^3)$	$\Theta(n)$	$\Theta(\log n)$	$\Theta(n^2)$	$\Theta\left(\frac{n^3}{B\sqrt{M}} + \frac{n^2}{B} + n\right)$

- How can we further improve parallelism?
  Reduction! i.e., using extra space

MM Parallel Not-In-Place D&C

MM(A, B, C, n)

1. if $n = 1$ then **MM-Loop**(A, B, C, n)
2. else
3. **parallel:** **MM**$(A_{11}, B_{11}, C_{11}, n/2)$, **MM**$(A_{11}, B_{12}, C_{12}, n/2)$
   MM$(A_{21}, B_{11}, C_{21}, n/2)$, **MM**$(A_{21}, B_{12}, C_{22}, n/2)$
   MM$(A_{12}, B_{21}, D_{11}, n/2)$, **MM**$(A_{12}, B_{22}, D_{12}, n/2)$
   MM$(A_{22}, B_{21}, D_{21}, n/2)$, **MM**$(A_{22}, B_{22}, D_{22}, n/2)$
4. $C \leftarrow$ **Parallel-Matrix-Sum**(C, D) $\triangleright$ $C \leftarrow C + D$

MM Parallel Not-In-Place D&C

MM(A, B, C, n)

1. if $n = 1$ then **MM-Loop**(A, B, C, n)
2. else
3. **parallel:** **MM**$(A_{11}, B_{11}, C_{11}, n/2)$, **MM**$(A_{11}, B_{12}, C_{12}, n/2)$
   MM$(A_{21}, B_{11}, C_{21}, n/2)$, **MM**$(A_{21}, B_{12}, C_{22}, n/2)$
   MM$(A_{12}, B_{21}, D_{11}, n/2)$, **MM**$(A_{12}, B_{22}, D_{12}, n/2)$
   MM$(A_{22}, B_{21}, D_{21}, n/2)$, **MM**$(A_{22}, B_{22}, D_{22}, n/2)$
4. $C \leftarrow$ **Parallel-Matrix-Sum**(C, D) $\triangleright$ $C \leftarrow C + D$

Complexity

$$T_1(n) = \begin{cases} \Theta(1) & \text{if } n = 1, \\ 8T_1(\frac{n}{2}) + \Theta(1) & \text{if } n > 1. \end{cases}$$

$$S_\infty(n) = \begin{cases} \Theta(1) & \text{if } n = 1, \\ 8S_\infty(\frac{n}{2}) + \Theta(n^2) & \text{if } n > 1. \end{cases}$$

$$T_\infty(n) = \begin{cases} \Theta(1) & \text{if } n = 1, \\ T_\infty(\frac{n}{2}) + \Theta(\log n) & \text{if } n > 1. \end{cases}$$

$$Q_1(n) = \Theta\left(n^3\right)$$

MM Parallel Not-In-Place D&C

MM(A, B, C, n)

1. if $n = 1$ then **MM-Loop**(A, B, C, n)
2. else
3. **parallel:** **MM**$(A_{11}, B_{11}, C_{11}, n/2)$, **MM**$(A_{11}, B_{12}, C_{12}, n/2)$
   MM$(A_{21}, B_{11}, C_{21}, n/2)$, **MM**$(A_{21}, B_{12}, C_{22}, n/2)$
   MM$(A_{12}, B_{21}, D_{11}, n/2)$, **MM**$(A_{12}, B_{22}, D_{12}, n/2)$
   MM$(A_{22}, B_{21}, D_{21}, n/2)$, **MM**$(A_{22}, B_{22}, D_{22}, n/2)$
4. $C \leftarrow$ **Parallel-Matrix-Sum**(C, D) $\qquad \triangleright\ C \leftarrow C + D$

Complexity

$$T_1(n) = \begin{cases} \Theta(1) & \text{if } n = 1, \\ 8T_1(\frac{n}{2}) + \Theta(1) & \text{if } n > 1. \end{cases}$$

$$S_\infty(n) = \begin{cases} \Theta(1) & \text{if } n = 1, \\ 8S_\infty(\frac{n}{2}) + \Theta(n^2) & \text{if } n > 1. \end{cases}$$

$$T_\infty(n) = \begin{cases} \Theta(1) & \text{if } n = 1, \\ T_\infty(\frac{n}{2}) + \Theta(\log n) & \text{if } n > 1. \end{cases}$$

$$Q_1(n) = \Theta\left(n^3\right)$$

$T_1(n)$	$T_\infty(n)$	$E_1(n)$	$S_\infty(n)$	$Q_1(n)$
$\Theta(n^3)$	$\Theta\left(\log^2 n\right)$	$\Theta(n^2)$	$\Theta(n^3)$	$\Theta(n^3)$

MM Parallel Not-In-Place D&C

MM(A, B, C, n)

1. if $n = 1$ then **MM-Loop**(A, B, C, n)
2. else
3. **parallel:** **MM**$(A_{11}, B_{11}, C_{11}, n/2)$, **MM**$(A_{11}, B_{12}, C_{12}, n/2)$
   MM$(A_{21}, B_{11}, C_{21}, n/2)$, **MM**$(A_{21}, B_{12}, C_{22}, n/2)$
   MM$(A_{12}, B_{21}, D_{11}, n/2)$, **MM**$(A_{12}, B_{22}, D_{12}, n/2)$
   MM$(A_{22}, B_{21}, D_{21}, n/2)$, **MM**$(A_{22}, B_{22}, D_{22}, n/2)$
4. $C \leftarrow$ **Parallel-Matrix-Sum**(C, D) $\triangleright\ C \leftarrow C + D$

Complexity

$$T_1(n) = \begin{cases} \Theta(1) & \text{if } n = 1, \\ 8T_1(\frac{n}{2}) + \Theta(1) & \text{if } n > 1. \end{cases}$$

$$S_\infty(n) = \begin{cases} \Theta(1) & \text{if } n = 1, \\ 8S_\infty(\frac{n}{2}) + \Theta(n^2) & \text{if } n > 1. \end{cases}$$

$$T_\infty(n) = \begin{cases} \Theta(1) & \text{if } n = 1, \\ T_\infty(\frac{n}{2}) + \Theta(\log n) & \text{if } n > 1. \end{cases}$$

$$Q_1(n) = \Theta\left(n^3\right)$$

$T_1(n)$	$T_\infty(n)$	$E_1(n)$	$S_\infty(n)$	$Q_1(n)$
$\Theta(n^3)$	$\Theta\left(\log^2 n\right)$	$\Theta(n^2)$	$\Theta(n^3)$	$\Theta(n^3)$

- How can we improve cache complexity?

MM Parallel Not-In-Place D&C

MM(A, B, C, n)

1. if $n = 1$ then **MM-Loop**(A, B, C, n)
2. else
3. **parallel:** **MM**$(A_{11}, B_{11}, C_{11}, n/2)$, **MM**$(A_{11}, B_{12}, C_{12}, n/2)$
   MM$(A_{21}, B_{11}, C_{21}, n/2)$, **MM**$(A_{21}, B_{12}, C_{22}, n/2)$
   MM$(A_{12}, B_{21}, D_{11}, n/2)$, **MM**$(A_{12}, B_{22}, D_{12}, n/2)$
   MM$(A_{22}, B_{21}, D_{21}, n/2)$, **MM**$(A_{22}, B_{22}, D_{22}, n/2)$
4. $C \leftarrow$ **Parallel-Matrix-Sum**(C, D) $\triangleright$ $C \leftarrow C + D$

Complexity

$$T_1(n) = \begin{cases} \Theta(1) & \text{if } n = 1, \\ 8T_1(\frac{n}{2}) + \Theta(1) & \text{if } n > 1. \end{cases} \qquad S_\infty(n) = \begin{cases} \Theta(1) & \text{if } n = 1, \\ 8S_\infty(\frac{n}{2}) + \Theta(n^2) & \text{if } n > 1. \end{cases}$$

$$T_\infty(n) = \begin{cases} \Theta(1) & \text{if } n = 1, \\ T_\infty(\frac{n}{2}) + \Theta(\log n) & \text{if } n > 1. \end{cases} \qquad Q_1(n) = \Theta\left(n^3\right)$$

$T_1(n)$	$T_\infty(n)$	$E_1(n)$	$S_\infty(n)$	$Q_1(n)$
$\Theta(n^3)$	$\Theta\left(\log^2 n\right)$	$\Theta(n^2)$	$\Theta(n^3)$	$\Theta(n^3)$

- How can we improve cache complexity?
  In-place tiled basecase!

MM Parallel Not-In-Place D&C

MM(A, B, C, n)

1. if $n = \sqrt{M}$ then **MM-Base**(A, B, C, n) ▷ Parallel in-place MM
2. else
3. **parallel:** **MM**$(A_{11}, B_{11}, C_{11}, n/2)$, **MM**$(A_{11}, B_{12}, C_{12}, n/2)$
   MM$(A_{21}, B_{11}, C_{21}, n/2)$, **MM**$(A_{21}, B_{12}, C_{22}, n/2)$
   MM$(A_{12}, B_{21}, D_{11}, n/2)$, **MM**$(A_{12}, B_{22}, D_{12}, n/2)$
   MM$(A_{22}, B_{21}, D_{21}, n/2)$, **MM**$(A_{22}, B_{22}, D_{22}, n/2)$
4. $C \leftarrow$ **Parallel-Matrix-Sum**(C, D) ▷ $C \leftarrow C + D$

MM Parallel Not-In-Place D&C

MM(A, B, C, n)

1. if $n = \sqrt{M}$ then **MM-Base**(A, B, C, n) ▷ Parallel in-place MM
2. else
3. **parallel:** **MM**$(A_{11}, B_{11}, C_{11}, n/2)$, **MM**$(A_{11}, B_{12}, C_{12}, n/2)$
   MM$(A_{21}, B_{11}, C_{21}, n/2)$, **MM**$(A_{21}, B_{12}, C_{22}, n/2)$
   MM$(A_{12}, B_{21}, D_{11}, n/2)$, **MM**$(A_{12}, B_{22}, D_{12}, n/2)$
   MM$(A_{22}, B_{21}, D_{21}, n/2)$, **MM**$(A_{22}, B_{22}, D_{22}, n/2)$
4. $C \leftarrow$ **Parallel-Matrix-Sum**(C, D) ▷ $C \leftarrow C + D$

Complexity

$$T_1(n) = \begin{cases} \Theta\left(n^3\right) & \text{if } n \leq \alpha\sqrt{M}, \\ 8T_1\left(\frac{n}{2}\right) + \Theta\left(1\right) & \text{if } n > \alpha\sqrt{M}. \end{cases}$$

$$S_\infty(n) = \begin{cases} \Theta\left(n^2\right) & \text{if } n^2 \leq \alpha M, \\ 8S_\infty\left(\frac{n}{2}\right) + \Theta\left(n^2\right) & \text{if } n^2 > \alpha M. \end{cases}$$

$$T_\infty(n) = \begin{cases} \Theta\left(n\right) & \text{if } n \leq \alpha\sqrt{M}, \\ T_\infty\left(\frac{n}{2}\right) + \Theta\left(\log n\right) & \text{if } n > \alpha\sqrt{M}. \end{cases}$$

$$Q_1(n) = \begin{cases} \Theta\left(\frac{n^2}{B} + n\right) & \text{if } n^2 \leq \alpha M, \\ 8Q_1\left(\frac{n}{2}\right) + \Theta\left(\frac{n^2}{B} + n\right) & \text{if } n^2 > \alpha M. \end{cases}$$

MM Parallel Not-In-Place D&C

MM(A, B, C, n)

1. if $n = \sqrt{M}$ then **MM-Base**(A, B, C, n) ▷ Parallel in-place MM
2. else
3. **parallel:** **MM**$(A_{11}, B_{11}, C_{11}, n/2)$, **MM**$(A_{11}, B_{12}, C_{12}, n/2)$
   MM$(A_{21}, B_{11}, C_{21}, n/2)$, **MM**$(A_{21}, B_{12}, C_{22}, n/2)$
   MM$(A_{12}, B_{21}, D_{11}, n/2)$, **MM**$(A_{12}, B_{22}, D_{12}, n/2)$
   MM$(A_{22}, B_{21}, D_{21}, n/2)$, **MM**$(A_{22}, B_{22}, D_{22}, n/2)$
4. $C \leftarrow$ **Parallel-Matrix-Sum**(C, D) ▷ $C \leftarrow C + D$

Complexity

$$T_1(n) = \begin{cases} \Theta\left(n^3\right) & \text{if } n \le \alpha\sqrt{M}, \\ 8T_1\left(\frac{n}{2}\right) + \Theta\left(1\right) & \text{if } n > \alpha\sqrt{M}. \end{cases}$$

$$S_\infty(n) = \begin{cases} \Theta\left(n^2\right) & \text{if } n^2 \le \alpha M, \\ 8S_\infty\left(\frac{n}{2}\right) + \Theta\left(n^2\right) & \text{if } n^2 > \alpha M. \end{cases}$$

$$T_\infty(n) = \begin{cases} \Theta\left(n\right) & \text{if } n \le \alpha\sqrt{M}, \\ T_\infty\left(\frac{n}{2}\right) + \Theta\left(\log n\right) & \text{if } n > \alpha\sqrt{M}. \end{cases}$$

$$Q_1(n) = \begin{cases} \Theta\left(\frac{n^2}{B} + n\right) & \text{if } n^2 \le \alpha M, \\ 8Q_1\left(\frac{n}{2}\right) + \Theta\left(\frac{n^2}{B} + n\right) & \text{if } n^2 > \alpha M. \end{cases}$$

$T_1(n)$	$T_\infty(n)$	$E_1(n)$	$S_\infty(n)$	$Q_1(n)$
$\Theta\left(n^3\right)$	$\Theta\left(\sqrt{M} + \log n\right)$	$\Theta\left(n^2\right)$	$\Theta\left(\frac{n^3}{\sqrt{M}}\right)$	$\Theta\left(\frac{n^3}{B\sqrt{M}} + \frac{n^2}{B} + n\right)$

MM Parallel Not-In-Place D&C

MM(A, B, C, n)

1. if $n = \sqrt{M}$ then **MM-Base**(A, B, C, n) ▷ Parallel in-place MM
2. else
3. **parallel:** **MM**$(A_{11}, B_{11}, C_{11}, n/2)$, **MM**$(A_{11}, B_{12}, C_{12}, n/2)$
   MM$(A_{21}, B_{11}, C_{21}, n/2)$, **MM**$(A_{21}, B_{12}, C_{22}, n/2)$
   MM$(A_{12}, B_{21}, D_{11}, n/2)$, **MM**$(A_{12}, B_{22}, D_{12}, n/2)$
   MM$(A_{22}, B_{21}, D_{21}, n/2)$, **MM**$(A_{22}, B_{22}, D_{22}, n/2)$
4. $C \leftarrow$ **Parallel-Matrix-Sum**(C, D) ▷ $C \leftarrow C + D$

Complexity

$$T_1(n) = \begin{cases} \Theta\left(n^3\right) & \text{if } n \le \alpha\sqrt{M}, \\ 8T_1\left(\frac{n}{2}\right) + \Theta\left(1\right) & \text{if } n > \alpha\sqrt{M}. \end{cases} \qquad S_\infty(n) = \begin{cases} \Theta\left(n^2\right) & \text{if } n^2 \le \alpha M, \\ 8S_\infty\left(\frac{n}{2}\right) + \Theta\left(n^2\right) & \text{if } n^2 > \alpha M. \end{cases}$$

$$T_\infty(n) = \begin{cases} \Theta\left(n\right) & \text{if } n \le \alpha\sqrt{M}, \\ T_\infty\left(\frac{n}{2}\right) + \Theta\left(\log n\right) & \text{if } n > \alpha\sqrt{M}. \end{cases} \qquad Q_1(n) = \begin{cases} \Theta\left(\frac{n^2}{B} + n\right) & \text{if } n^2 \le \alpha M, \\ 8Q_1\left(\frac{n}{2}\right) + \Theta\left(\frac{n^2}{B} + n\right) & \text{if } n^2 > \alpha M. \end{cases}$$

$T_1(n)$	$T_\infty(n)$	$E_1(n)$	$S_\infty(n)$	$Q_1(n)$
$\Theta\left(n^3\right)$	$\Theta\left(\sqrt{M} + \log n\right)$	$\Theta\left(n^2\right)$	$\Theta\left(\frac{n^3}{\sqrt{M}}\right)$	$\Theta\left(\frac{n^3}{B\sqrt{M}} + \frac{n^2}{B} + n\right)$

- How can we reduce work?

MM Parallel Not-In-Place D&C

MM(A, B, C, n)

1. if $n = \sqrt{M}$ then **MM-Base**(A, B, C, n) ▷ Parallel in-place MM
2. else
3. **parallel:** **MM**$(A_{11}, B_{11}, C_{11}, n/2)$, **MM**$(A_{11}, B_{12}, C_{12}, n/2)$
   MM$(A_{21}, B_{11}, C_{21}, n/2)$, **MM**$(A_{21}, B_{12}, C_{22}, n/2)$
   MM$(A_{12}, B_{21}, D_{11}, n/2)$, **MM**$(A_{12}, B_{22}, D_{12}, n/2)$
   MM$(A_{22}, B_{21}, D_{21}, n/2)$, **MM**$(A_{22}, B_{22}, D_{22}, n/2)$
4. $C \leftarrow$ **Parallel-Matrix-Sum**(C, D) ▷ $C \leftarrow C + D$

Complexity

$$T_1(n) = \begin{cases} \Theta\left(n^3\right) & \text{if } n \le \alpha\sqrt{M}, \\ 8T_1\left(\frac{n}{2}\right) + \Theta\left(1\right) & \text{if } n > \alpha\sqrt{M}. \end{cases}$$

$$S_\infty(n) = \begin{cases} \Theta\left(n^2\right) & \text{if } n^2 \le \alpha M, \\ 8S_\infty\left(\frac{n}{2}\right) + \Theta\left(n^2\right) & \text{if } n^2 > \alpha M. \end{cases}$$

$$T_\infty(n) = \begin{cases} \Theta\left(n\right) & \text{if } n \le \alpha\sqrt{M}, \\ T_\infty\left(\frac{n}{2}\right) + \Theta\left(\log n\right) & \text{if } n > \alpha\sqrt{M}. \end{cases}$$

$$Q_1(n) = \begin{cases} \Theta\left(\frac{n^2}{B} + n\right) & \text{if } n^2 \le \alpha M, \\ 8Q_1\left(\frac{n}{2}\right) + \Theta\left(\frac{n^2}{B} + n\right) & \text{if } n^2 > \alpha M. \end{cases}$$

$T_1(n)$	$T_\infty(n)$	$E_1(n)$	$S_\infty(n)$	$Q_1(n)$
$\Theta\left(n^3\right)$	$\Theta\left(\sqrt{M} + \log n\right)$	$\Theta\left(n^2\right)$	$\Theta\left(\frac{n^3}{\sqrt{M}}\right)$	$\Theta\left(\frac{n^3}{B\sqrt{M}} + \frac{n^2}{B} + n\right)$

- How can we reduce work? Strassen's algorithm!

Volker Strassen's MM D&C: Core Idea

Problem

Is there is a strategy to perform multiplication of two complex numbers with only 3 multiplications?

$(a + ib)(c + id) = (ac - bd) + i(bc + ad)$

Volker Strassen's MM D&C: Core Idea

Problem

Is there is a strategy to perform multiplication of two complex numbers with only 3 multiplications?
$(a + ib)(c + id) = (ac - bd) + i(bc + ad)$

Solution 1

Let $x = bd$, $y = ac$, and $z = (a + b)(c + d)$.
Then, real part $= y - x$ and imaginary part $= z - x - y$.

Volker Strassen's MM D&C: Core Idea

Problem

Is there is a strategy to perform multiplication of two complex numbers with only 3 multiplications?
$(a+ib)(c+id) = (ac-bd) + i(bc+ad)$

Solution 1

Let $x = bd$, $y = ac$, and $z = (a+b)(c+d)$.
Then, real part $= y - x$ and imaginary part $= z - x - y$.

Solution 2

Let $x = c(a+b)$, $y = a(d-c)$, and $z = b(c+d)$.
Then, real part $= x - z$ and imaginary part $= x + y$.

Volker Strassen's MM D&C: Core Idea

Problem

Is there is a strategy to perform multiplication of two complex numbers with only 3 multiplications?
$(a + ib)(c + id) = (ac - bd) + i(bc + ad)$

Solution 1

Let $x = bd$, $y = ac$, and $z = (a + b)(c + d)$.
Then, real part $= y - x$ and imaginary part $= z - x - y$.

Solution 2

Let $x = c(a + b)$, $y = a(d - c)$, and $z = b(c + d)$.
Then, real part $= x - z$ and imaginary part $= x + y$.

Solution 3

Let $x = c(a + b)$, $y = a(c - d)$, and $z = d(a - b)$.
Then, real part $= y + z$ and imaginary part $= x - y$.

Volker Strassen's MM D&C: Core Idea

Problem	Traditional	Solution 1	Solution 2	Solution 3
Complex number mult.	4 mults 2 adds	3 mults 5 adds	3 mults 5 adds	3 mults 5 adds

Volker Strassen's MM D&C: Core Idea

Problem	Traditional	Solution 1	Solution 2	Solution 3
Complex number mult.	4 mults 2 adds	3 mults 5 adds	3 mults 5 adds	3 mults 5 adds

(block size $n/2 \times n/2$)

$$\begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \times \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}$$

$$= \begin{bmatrix} A_{11}B_{11} + A_{12}B_{21} & A_{11}B_{12} + A_{12}B_{22} \\ A_{21}B_{11} + A_{22}B_{21} & A_{21}B_{12} + A_{22}B_{22} \end{bmatrix}$$

Problem	Traditional	Strassen	Winograd
2×2 MM	8 mults 4 adds	7 mults 18 adds	7 mults 15 adds
$n \times n$ MM	n^3 mults $(n^3 - n^2)$ adds	$n^{\log_2 7}$ mults $(6n^{\log_2 7} - 6n^2)$ adds	$n^{\log_2 7}$ mults $(5n^{\log_2 7} - 5n^2)$ adds

Volker Strassen's MM D&C

MM-Strassen(A, B, C, n)

1. $P_1 \leftarrow (A_{11} + A_{22}) \times (B_{11} + B_{22})$
2. $P_2 \leftarrow (A_{21} + A_{22}) \times B_{11}$
3. $P_3 \leftarrow A_{11} \times (B_{12} - B_{22})$
4. $P_4 \leftarrow A_{22} \times (B_{21} - B_{11})$
5. $P_5 \leftarrow (A_{11} + A_{12}) \times B_{22}$
6. $P_6 \leftarrow (A_{21} - A_{11}) \times (B_{11} + B_{12})$
7. $P_7 \leftarrow (A_{12} - A_{22}) \times (B_{21} + B_{22})$
8. $C_{11} \leftarrow P_1 + P_4 - P_5 + P_7$
9. $C_{12} \leftarrow P_3 + P_5$
10. $C_{21} \leftarrow P_2 + P_4$
11. $C_{22} \leftarrow P_1 - P_2 + P_3 + P_6$

Volker Strassen's MM D&C

MM-Strassen(A, B, C, n)

1. $P_1 \leftarrow (A_{11} + A_{22}) \times (B_{11} + B_{22})$
2. $P_2 \leftarrow (A_{21} + A_{22}) \times B_{11}$
3. $P_3 \leftarrow A_{11} \times (B_{12} - B_{22})$
4. $P_4 \leftarrow A_{22} \times (B_{21} - B_{11})$
5. $P_5 \leftarrow (A_{11} + A_{12}) \times B_{22}$
6. $P_6 \leftarrow (A_{21} - A_{11}) \times (B_{11} + B_{12})$
7. $P_7 \leftarrow (A_{12} - A_{22}) \times (B_{21} + B_{22})$
8. $C_{11} \leftarrow P_1 + P_4 - P_5 + P_7$
9. $C_{12} \leftarrow P_3 + P_5$
10. $C_{21} \leftarrow P_2 + P_4$
11. $C_{22} \leftarrow P_1 - P_2 + P_3 + P_6$

Complexity

$$T_1(n) = \begin{cases} \Theta(1) & \text{if } n = 1, \\ 7T_1(\frac{n}{2}) + \Theta\left(n^2\right) & \text{if } n > 1. \end{cases}$$

$$= \Theta\left(n^{\log_2 7}\right)$$

Volker Strassen's MM D&C

MM-Strassen(A, B, C, n)

1. $P_1 \leftarrow (A_{11} + A_{22}) \times (B_{11} + B_{22})$
2. $P_2 \leftarrow (A_{21} + A_{22}) \times B_{11}$
3. $P_3 \leftarrow A_{11} \times (B_{12} - B_{22})$
4. $P_4 \leftarrow A_{22} \times (B_{21} - B_{11})$
5. $P_5 \leftarrow (A_{11} + A_{12}) \times B_{22}$
6. $P_6 \leftarrow (A_{21} - A_{11}) \times (B_{11} + B_{12})$
7. $P_7 \leftarrow (A_{12} - A_{22}) \times (B_{21} + B_{22})$
8. $C_{11} \leftarrow P_1 + P_4 - P_5 + P_7$
9. $C_{12} \leftarrow P_3 + P_5$
10. $C_{21} \leftarrow P_2 + P_4$
11. $C_{22} \leftarrow P_1 - P_2 + P_3 + P_6$

Complexity

$$T_1(n) = \begin{cases} \Theta(1) & \text{if } n = 1, \\ 7T_1(\frac{n}{2}) + \Theta\left(n^2\right) & \text{if } n > 1. \end{cases}$$
$$= \Theta\left(n^{\log_2 7}\right)$$

- How can we parallelize the algorithm?
- What are the complexities of the parallel algorithm?

Shmuel Winograd's MM D&C

MM-Winograd(A, B, C, n)

1. $R_1 \leftarrow A_{21} + A_{22}$
2. $R_2 \leftarrow R_1 - A_{11}$
3. $R_3 \leftarrow A_{11} - A_{21}$
4. $R_4 \leftarrow A_{12} - R_2$
5. $R_5 \leftarrow B_{12} - B_{11}$
6. $R_6 \leftarrow B_{22} - R_5$
7. $R_7 \leftarrow B_{22} - B_{12}$
8. $R_8 \leftarrow R_6 - B_{21}$
9. $P_1 \leftarrow R_2 \times R_6$
10. $P_2 \leftarrow A_{11} \times B_{11}$
11. $P_3 \leftarrow A_{12} \times B_{21}$
12. $P_4 \leftarrow R_3 \times R_7$
13. $P_5 \leftarrow R_1 \times R_5$
14. $P_6 \leftarrow R_4 \times B_{22}$
15. $P_7 \leftarrow A_{22} \times R_8$
16. $V_1 \leftarrow P_1 + P_2$
17. $V_2 \leftarrow V_1 + P_4$
18. $C_{11} \leftarrow P_2 + P_3$
19. $C_{12} \leftarrow V_1 + P_5 + P_6$
20. $C_{21} \leftarrow V_2 - P_7$
21. $C_{22} \leftarrow V_2 + P_5$

Shmuel Winograd's MM D&C

MM-Winograd(A, B, C, n)

1. $R_1 \leftarrow A_{21} + A_{22}$
2. $R_2 \leftarrow R_1 - A_{11}$
3. $R_3 \leftarrow A_{11} - A_{21}$
4. $R_4 \leftarrow A_{12} - R_2$
5. $R_5 \leftarrow B_{12} - B_{11}$
6. $R_6 \leftarrow B_{22} - R_5$
7. $R_7 \leftarrow B_{22} - B_{12}$
8. $R_8 \leftarrow R_6 - B_{21}$
9. $P_1 \leftarrow R_2 \times R_6$
10. $P_2 \leftarrow A_{11} \times B_{11}$
11. $P_3 \leftarrow A_{12} \times B_{21}$
12. $P_4 \leftarrow R_3 \times R_7$
13. $P_5 \leftarrow R_1 \times R_5$
14. $P_6 \leftarrow R_4 \times B_{22}$
15. $P_7 \leftarrow A_{22} \times R_8$
16. $V_1 \leftarrow P_1 + P_2$
17. $V_2 \leftarrow V_1 + P_4$
18. $C_{11} \leftarrow P_2 + P_3$
19. $C_{12} \leftarrow V_1 + P_5 + P_6$
20. $C_{21} \leftarrow V_2 - P_7$
21. $C_{22} \leftarrow V_2 + P_5$

Complexity

$$T_1(n) = \begin{cases} \Theta(1) & \text{if } n = 1, \\ 7T_1(\frac{n}{2}) + \Theta\left(n^2\right) & \text{if } n > 1. \end{cases}$$
$$= \Theta\left(n^{\log_2 7}\right)$$

Shmuel Winograd's MM D&C

MM-Winograd(A, B, C, n)

1. $R_1 \leftarrow A_{21} + A_{22}$
2. $R_2 \leftarrow R_1 - A_{11}$
3. $R_3 \leftarrow A_{11} - A_{21}$
4. $R_4 \leftarrow A_{12} - R_2$
5. $R_5 \leftarrow B_{12} - B_{11}$
6. $R_6 \leftarrow B_{22} - R_5$
7. $R_7 \leftarrow B_{22} - B_{12}$
8. $R_8 \leftarrow R_6 - B_{21}$
9. $P_1 \leftarrow R_2 \times R_6$
10. $P_2 \leftarrow A_{11} \times B_{11}$
11. $P_3 \leftarrow A_{12} \times B_{21}$
12. $P_4 \leftarrow R_3 \times R_7$
13. $P_5 \leftarrow R_1 \times R_5$
14. $P_6 \leftarrow R_4 \times B_{22}$
15. $P_7 \leftarrow A_{22} \times R_8$
16. $V_1 \leftarrow P_1 + P_2$
17. $V_2 \leftarrow V_1 + P_4$
18. $C_{11} \leftarrow P_2 + P_3$
19. $C_{12} \leftarrow V_1 + P_5 + P_6$
20. $C_{21} \leftarrow V_2 - P_7$
21. $C_{22} \leftarrow V_2 + P_5$

Complexity

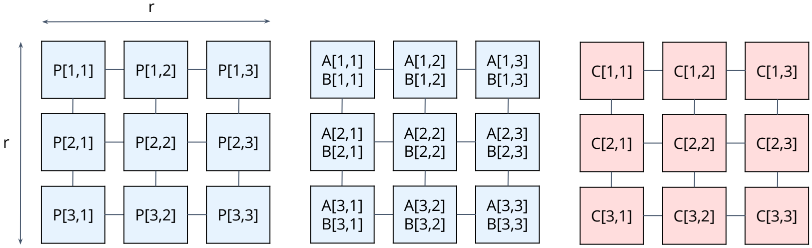
$$T_1(n) = \begin{cases} \Theta\left(1\right) & \text{if } n = 1, \\ 7T_1(\frac{n}{2}) + \Theta\left(n^2\right) & \text{if } n > 1. \end{cases}$$
$$= \Theta\left(n^{\log_2 7}\right)$$

- How can we parallelize the algorithm?

MM Distributed

Distributed setup

- Let $r = \sqrt{p}$. The machine architecture is an $r \times r$ processor grid/mesh/hypercube
- Split the $n \times n$ sized A, B matrices into $r \times r$ blocks/tiles
- Initially, processor $P[i,j]$ holds $A[i,j]$ and $B[i,j]$ input blocks
- After the algorithm completes, processor $P[i,j]$ holds $C[i,j]$

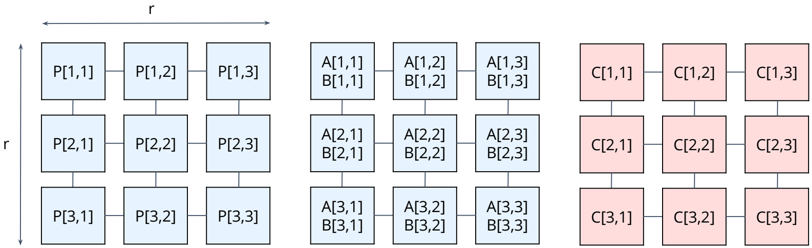


r x r processor grid — Initial input distribution — Final output

MM Distributed

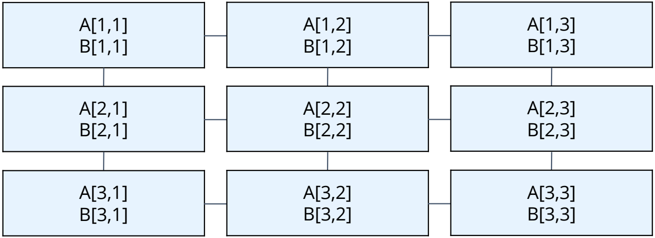
MM(A, B, C, n, p)

1. $r \leftarrow \sqrt{p}$, where $p \leq n^2$
2. A, B matrices are split into $r \times r$ blocks/tiles
3. $P[i, j]$ initially stores $A[i, j]$ and $B[i, j]$ blocks/tiles
4. **Broadcast-Rowwise**(A's blocks)
5. **Broadcast-Columnwise**(B's blocks)
6. $P[i, j]$ computes $\sum_{k=1}^{r} A[i, k] \times B[k, j]$

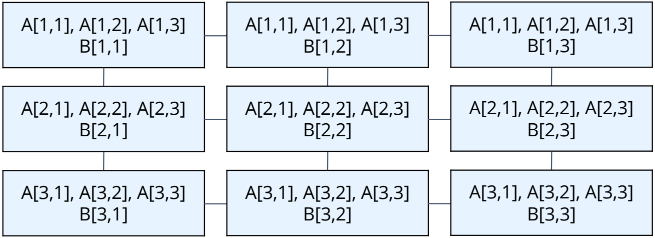


r x r processor grid — Initial input distribution — Final output

MM Distributed

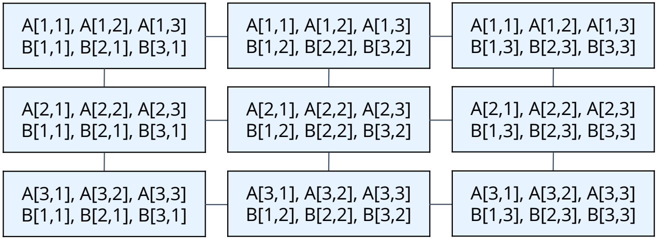


Initial input distribution

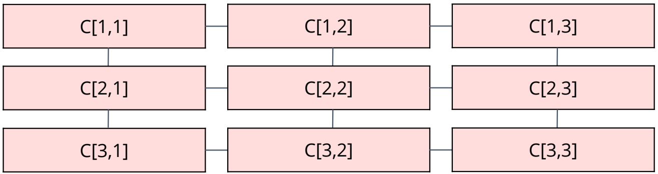


Broadcast rowwise

MM Distributed



Broadcast columnwise



Final output

MM Distributed: Complexity

MM(A, B, C, n, p)

1. $r \leftarrow \sqrt{p}$, where $p \leq n^2$
2. A, B matrices are split into $r \times r$ blocks/tiles
3. $P[i,j]$ initially stores $A[i,j]$ and $B[i,j]$ blocks/tiles
4. **Broadcast-Rowwise**(A's blocks)
5. **Broadcast-Columnwise**(B's blocks)
6. $P[i,j]$ computes $\sum_{k=1}^{r} A[i,k] \times B[k,j]$

Step	$T_{\text{comp}}(n)$	$T_{\text{comm}}(n)$	$S_{\text{max}}(n)$
BC-Row (r)	$-$	$t_s \log r + t_w(r-1)\left(\frac{n}{r} \cdot \frac{n}{r}\right)$	$r\left(\frac{n}{r} \cdot \frac{n}{r}\right)$
BC-Col (r)	$-$	$t_s \log r + t_w(r-1)\left(\frac{n}{r} \cdot \frac{n}{r}\right)$	$r\left(\frac{n}{r} \cdot \frac{n}{r}\right)$
Compute $C[i,j]$	$r\left(\frac{n}{r}\right)^3$	$-$	$\left(\frac{n}{r} \cdot \frac{n}{r}\right)$
Total (r)	$\frac{n^3}{r^2}$	$2\left(t_s \log r + t_w(r-1)\left(\frac{n}{r}\right)^2\right)$	$(2r+1)\left(\frac{n}{r}\right)^2$
Total $(r = \sqrt{p})$	$\frac{n^3}{p}$	$t_s \log p + 2t_w(\sqrt{p}-1)\frac{n^2}{p}$	$(2\sqrt{p}+1)\frac{n^2}{p}$

- How can we improve space complexity??