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 $i$th row $\times$ B’s $j$th column = $C[i, j]$ cell
- E.g.: $5 \times 4 + 8 \times 6 + 3 \times 8 + 8 \times 5 = 132$
### Square Matrix Multiplication

#### Example

| 2 7 3 6 | 8 4 4 3 | 92 96 104 116 |
| 5 8 3 8 | 7 7 6 8 | 127 125 132 147 |
| 6 4 5 6 | 5 3 8 4 | 113 97 118 112 |
| 0 3 9 7 | 2 5 5 7 | 80 83 125 109 |

- $A$’s $i$th row $\times$ $B$’s $j$th 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

$$C[i, j] = \sum_{k=1}^{n} A[i, k] \times B[k, j] \text{ for } i, j \in [1, n]$$
The A, B, C matrices are stored in row-major order

How can we improve cache complexity?

Reorder the loops!
MM Loops

\[ \text{MM-Loop}(A, B, C, n) \]

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. \( C[i, j] \leftarrow C[i, j] + A[i, k] \times B[k, j] \)

<table>
<thead>
<tr>
<th>( T_1(n) )</th>
<th>( T_\infty(n) )</th>
<th>( E_1(n) )</th>
<th>( S_\infty(n) )</th>
<th>( Q_1(n) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Theta(n^3) )</td>
<td>( \Theta(n^3) )</td>
<td>( \Theta(1) )</td>
<td>( \Theta(n^2) )</td>
<td>( \Theta(n^3) )</td>
</tr>
</tbody>
</table>
MM Loops

\[ \text{MM-Loop}(A, B, C, n) \]

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. \( C[i, j] \leftarrow C[i, j] + A[i, k] \times B[k, j] \)

<table>
<thead>
<tr>
<th>( T_1(n) )</th>
<th>( T_\infty(n) )</th>
<th>( E_1(n) )</th>
<th>( S_\infty(n) )</th>
<th>( Q_1(n) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Theta(n^3) )</td>
<td>( \Theta(n^3) )</td>
<td>( \Theta(1) )</td>
<td>( \Theta(n^2) )</td>
<td>( \Theta(n^3) )</td>
</tr>
</tbody>
</table>

- The \( A, B, C \) matrices are stored in row-major order
- How can we improve cache complexity?
**MM Loops**

<table>
<thead>
<tr>
<th>MM-Loop ((A, B, C, n))</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. for (i \leftarrow 1) to (n) do</td>
</tr>
<tr>
<td>2. for (j \leftarrow 1) to (n) do</td>
</tr>
<tr>
<td>3. (C[i, j] \leftarrow 0)</td>
</tr>
<tr>
<td>4. for (k \leftarrow 1) to (n) do</td>
</tr>
<tr>
<td>5. (C[i, j] \leftarrow C[i, j] + A[i, k] \times B[k, j])</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(T_1(n))</th>
<th>(T_\infty(n))</th>
<th>(E_1(n))</th>
<th>(S_\infty(n))</th>
<th>(Q_1(n))</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\Theta(n^3))</td>
<td>(\Theta(n^3))</td>
<td>(\Theta(1))</td>
<td>(\Theta(n^2))</td>
<td>(\Theta(n^3))</td>
</tr>
</tbody>
</table>

- 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)

<table>
<thead>
<tr>
<th>MM-ijk($A, B, C, n$)</th>
<th>MM-jki($A, B, C, n$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. for $i \leftarrow 1$ to $n$ do</td>
<td></td>
</tr>
<tr>
<td>2. for $j \leftarrow 1$ to $n$ do</td>
<td></td>
</tr>
<tr>
<td>3. for $k \leftarrow 1$ to $n$ do</td>
<td></td>
</tr>
<tr>
<td>1. for $j \leftarrow 1$ to $n$ do</td>
<td></td>
</tr>
<tr>
<td>2. for $k \leftarrow 1$ to $n$ do</td>
<td></td>
</tr>
<tr>
<td>3. for $i \leftarrow 1$ to $n$ do</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MM-ikj($A, B, C, n$)</th>
<th>MM-kij($A, B, C, n$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. for $i \leftarrow 1$ to $n$ do</td>
<td></td>
</tr>
<tr>
<td>2. for $k \leftarrow 1$ to $n$ do</td>
<td></td>
</tr>
<tr>
<td>3. for $j \leftarrow 1$ to $n$ do</td>
<td></td>
</tr>
<tr>
<td>1. for $k \leftarrow 1$ to $n$ do</td>
<td></td>
</tr>
<tr>
<td>2. for $i \leftarrow 1$ to $n$ do</td>
<td></td>
</tr>
<tr>
<td>3. for $j \leftarrow 1$ to $n$ do</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MM-jik($A, B, C, n$)</th>
<th>MM-kji($A, B, C, n$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. for $j \leftarrow 1$ to $n$ do</td>
<td></td>
</tr>
<tr>
<td>2. for $i \leftarrow 1$ to $n$ do</td>
<td></td>
</tr>
<tr>
<td>3. for $k \leftarrow 1$ to $n$ do</td>
<td></td>
</tr>
<tr>
<td>1. for $k \leftarrow 1$ to $n$ do</td>
<td></td>
</tr>
<tr>
<td>2. for $j \leftarrow 1$ to $n$ do</td>
<td></td>
</tr>
<tr>
<td>3. for $i \leftarrow 1$ to $n$ do</td>
<td></td>
</tr>
</tbody>
</table>

Which of these algorithms are correct?
Which of the correct algorithms have improved cache locality?
**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-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?
MM Loops (3! = 6 possible ways)

<table>
<thead>
<tr>
<th>MM-ijk$(A, B, C, n)$</th>
<th>MM-jki$(A, B, C, n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. for $i \leftarrow 1$ to $n$ do</td>
<td></td>
</tr>
<tr>
<td>2. for $j \leftarrow 1$ to $n$ do</td>
<td></td>
</tr>
<tr>
<td>3. for $k \leftarrow 1$ to $n$ do</td>
<td></td>
</tr>
<tr>
<td>1. for $j \leftarrow 1$ to $n$ do</td>
<td></td>
</tr>
<tr>
<td>2. for $k \leftarrow 1$ to $n$ do</td>
<td></td>
</tr>
<tr>
<td>3. for $i \leftarrow 1$ to $n$ do</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MM-ikj$(A, B, C, n)$</th>
<th>MM-kij$(A, B, C, n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. for $i \leftarrow 1$ to $n$ do</td>
<td></td>
</tr>
<tr>
<td>2. for $k \leftarrow 1$ to $n$ do</td>
<td></td>
</tr>
<tr>
<td>3. for $j \leftarrow 1$ to $n$ do</td>
<td></td>
</tr>
<tr>
<td>1. for $k \leftarrow 1$ to $n$ do</td>
<td></td>
</tr>
<tr>
<td>2. for $i \leftarrow 1$ to $n$ do</td>
<td></td>
</tr>
<tr>
<td>3. for $j \leftarrow 1$ to $n$ do</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MM-jik$(A, B, C, n)$</th>
<th>MM-kji$(A, B, C, n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. for $j \leftarrow 1$ to $n$ do</td>
<td></td>
</tr>
<tr>
<td>2. for $i \leftarrow 1$ to $n$ do</td>
<td></td>
</tr>
<tr>
<td>3. for $k \leftarrow 1$ to $n$ do</td>
<td></td>
</tr>
<tr>
<td>1. for $k \leftarrow 1$ to $n$ do</td>
<td></td>
</tr>
<tr>
<td>2. for $j \leftarrow 1$ to $n$ do</td>
<td></td>
</tr>
<tr>
<td>3. for $i \leftarrow 1$ to $n$ do</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>$T_1(n)$</th>
<th>$T_\infty(n)$</th>
<th>$E_1(n)$</th>
<th>$S_\infty(n)$</th>
<th>$Q_1(n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MM-ijkl</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(1)$</td>
<td>$\Theta(n^2)$</td>
<td>$\Theta(n^3)$</td>
</tr>
<tr>
<td>MM-ikj</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(1)$</td>
<td>$\Theta(n^2)$</td>
<td>$\Theta\left(\frac{n^3}{B} + n^2\right)$</td>
</tr>
<tr>
<td>MM-jik</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(1)$</td>
<td>$\Theta(n^2)$</td>
<td>$\Theta(n^3)$</td>
</tr>
<tr>
<td>MM-jki</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(1)$</td>
<td>$\Theta(n^2)$</td>
<td>$\Theta(n^3)$</td>
</tr>
<tr>
<td>MM-kij</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(1)$</td>
<td>$\Theta(n^2)$</td>
<td>$\Theta(n^3)$</td>
</tr>
<tr>
<td>MM-kji</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(1)$</td>
<td>$\Theta(n^2)$</td>
<td>$\Theta(n^3)$</td>
</tr>
</tbody>
</table>
### MM Loops: Complexity

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>$T_1(n)$</th>
<th>$T_\infty(n)$</th>
<th>$E_1(n)$</th>
<th>$S_\infty(n)$</th>
<th>$Q_1(n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MM-ijk</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(1)$</td>
<td>$\Theta(n^2)$</td>
<td>$\Theta(n^3)$</td>
</tr>
<tr>
<td>MM-ikj</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(1)$</td>
<td>$\Theta(n^2)$</td>
<td>$\Theta\left(\frac{n^3}{B} + n^2\right)$</td>
</tr>
<tr>
<td>MM-jik</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(1)$</td>
<td>$\Theta(n^2)$</td>
<td>$\Theta(n^3)$</td>
</tr>
<tr>
<td>MM-jki</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(1)$</td>
<td>$\Theta(n^2)$</td>
<td>$\Theta(n^3)$</td>
</tr>
<tr>
<td>MM-kij</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(1)$</td>
<td>$\Theta(n^2)$</td>
<td>$\Theta(n^3)$</td>
</tr>
<tr>
<td>MM-kji</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(1)$</td>
<td>$\Theta(n^2)$</td>
<td>$\Theta(n^3)$</td>
</tr>
</tbody>
</table>

- The matrices $A, B, C$ are stored in row-major order
- **MM-ikj** exploits spatial cache locality
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?
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-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-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-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. \(C[i, j] \leftarrow C[i, j] + A[i, k] \cdot B[k, j]\)

**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. \(C[i, j] \leftarrow C[i, j] + A[i, k] \cdot B[k, j]\)

▶ All the three loops are parallelized
▶ Are these algorithms correct?
All the three loops are parallelized.

Are these algorithms correct? No! (race conditions)
### MM Parallel Loops

<table>
<thead>
<tr>
<th>MM-ijk$(A, B, C, n)$</th>
<th>MM-jki$(A, B, C, n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. parallel for $i \leftarrow 1$ to $n$ do</td>
<td>1. parallel for $j \leftarrow 1$ to $n$ do</td>
</tr>
<tr>
<td>2. parallel for $j \leftarrow 1$ to $n$ do</td>
<td>2. parallel for $k \leftarrow 1$ to $n$ do</td>
</tr>
<tr>
<td>3. parallel for $k \leftarrow 1$ to $n$ do</td>
<td>3. parallel for $i \leftarrow 1$ to $n$ do</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MM-ikj$(A, B, C, n)$</th>
<th>MM-kij$(A, B, C, n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. parallel for $i \leftarrow 1$ to $n$ do</td>
<td>1. parallel for $k \leftarrow 1$ to $n$ do</td>
</tr>
<tr>
<td>2. parallel for $k \leftarrow 1$ to $n$ do</td>
<td>2. parallel for $i \leftarrow 1$ to $n$ do</td>
</tr>
<tr>
<td>3. parallel for $j \leftarrow 1$ to $n$ do</td>
<td>3. parallel for $j \leftarrow 1$ to $n$ do</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MM-jik$(A, B, C, n)$</th>
<th>MM-kji$(A, B, C, n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. parallel for $j \leftarrow 1$ to $n$ do</td>
<td>1. parallel for $k \leftarrow 1$ to $n$ do</td>
</tr>
<tr>
<td>2. parallel for $i \leftarrow 1$ to $n$ do</td>
<td>2. parallel for $j \leftarrow 1$ to $n$ do</td>
</tr>
<tr>
<td>3. parallel for $k \leftarrow 1$ to $n$ do</td>
<td>3. parallel for $i \leftarrow 1$ to $n$ do</td>
</tr>
</tbody>
</table>

- 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. for \(k \leftarrow 1\) to \(n\) do
2. `parallel` 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-kij** \((A, B, C, n)\)

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-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**\((A, B, C, n)\)

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. \(C[i, j] \leftarrow C[i, j] + A[i, k] \cdot B[k, j]\)

- The \(k\)-loop is serialized
- Are these algorithms correct? Yes! (no race conditions)
# MM Parallel Loops: Complexity

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>$T_1(n)$</th>
<th>$T_\infty(n)$</th>
<th>$E_1(n)$</th>
<th>$S_\infty(n)$</th>
<th>$Q_1(n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MM-ijk</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(n)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(n^2)$</td>
<td>$\Theta(n^3)$</td>
</tr>
<tr>
<td>MM-ikj</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(n)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(n^2)$</td>
<td>$\Theta\left(\frac{n^3}{B} + n^2\right)$</td>
</tr>
<tr>
<td>MM-jik</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(n)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(n^2)$</td>
<td>$\Theta(n^3)$</td>
</tr>
<tr>
<td>MM-jki</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(n)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(n^2)$</td>
<td>$\Theta(n^3)$</td>
</tr>
<tr>
<td>MM-kij</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(n)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(n^2)$</td>
<td>$\Theta(n^3)$</td>
</tr>
<tr>
<td>MM-kji</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(n)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(n^2)$</td>
<td>$\Theta(n^3)$</td>
</tr>
</tbody>
</table>

- How can we compute $S_p(n)$?
MM Parallel Loops: Complexity

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>$T_1(n)$</th>
<th>$T_∞(n)$</th>
<th>$E_1(n)$</th>
<th>$S_∞(n)$</th>
<th>$Q_1(n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MM-ijk</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(n)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(n^2)$</td>
<td>$\Theta(n^3)$</td>
</tr>
<tr>
<td>MM-ikj</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(n)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(n^2)$</td>
<td>$\Theta\left(\frac{n^3}{B} + n^2\right)$</td>
</tr>
<tr>
<td>MM-jik</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(n)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(n^2)$</td>
<td>$\Theta(n^3)$</td>
</tr>
<tr>
<td>MM-jki</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(n)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(n^2)$</td>
<td>$\Theta(n^3)$</td>
</tr>
<tr>
<td>MM-kij</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(n)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(n^2)$</td>
<td>$\Theta(n^3)$</td>
</tr>
<tr>
<td>MM-kji</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(n)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(n^2)$</td>
<td>$\Theta(n^3)$</td>
</tr>
</tbody>
</table>

▶ How can we compute $S_p(n)$?
▶ How can we improve parallelism?
## MM Parallel Loops: Complexity

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>$T_1(n)$</th>
<th>$T_\infty(n)$</th>
<th>$E_1(n)$</th>
<th>$S_\infty(n)$</th>
<th>$Q_1(n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MM-ijk</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(n)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(n^2)$</td>
<td>$\Theta(n^3)$</td>
</tr>
<tr>
<td>MM-ikj</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(n)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(n^2)$</td>
<td>$\Theta\left(\frac{n^3}{B} + n^2\right)$</td>
</tr>
<tr>
<td>MM-jik</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(n)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(n^2)$</td>
<td>$\Theta(n^3)$</td>
</tr>
<tr>
<td>MM-jki</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(n)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(n^2)$</td>
<td>$\Theta(n^3)$</td>
</tr>
<tr>
<td>MM-kij</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(n)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(n^2)$</td>
<td>$\Theta(n^3)$</td>
</tr>
<tr>
<td>MM-kji</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(n)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(n^2)$</td>
<td>$\Theta(n^3)$</td>
</tr>
</tbody>
</table>

- 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

<table>
<thead>
<tr>
<th>$\text{MM-ijk}(A, B, C, n)$</th>
<th>$\text{MM-jki}(A, B, C, n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.  parallel for $i \leftarrow 1$ to $n$ do</td>
<td></td>
</tr>
<tr>
<td>2.  parallel for $j \leftarrow 1$ to $n$ do</td>
<td></td>
</tr>
<tr>
<td>3.  reduce for $k \leftarrow 1$ to $n$ do</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\text{MM-ikj}(A, B, C, n)$</th>
<th>$\text{MM-kij}(A, B, C, n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.  parallel for $i \leftarrow 1$ to $n$ do</td>
<td></td>
</tr>
<tr>
<td>2.  reduce for $k \leftarrow 1$ to $n$ do</td>
<td></td>
</tr>
<tr>
<td>3.  parallel for $j \leftarrow 1$ to $n$ do</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\text{MM-jik}(A, B, C, n)$</th>
<th>$\text{MM-kji}(A, B, C, n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.  parallel for $j \leftarrow 1$ to $n$ do</td>
<td></td>
</tr>
<tr>
<td>2.  parallel for $i \leftarrow 1$ to $n$ do</td>
<td></td>
</tr>
<tr>
<td>3.  reduce for $k \leftarrow 1$ to $n$ do</td>
<td></td>
</tr>
</tbody>
</table>

- The $k$-loop is reduced

- Are these algorithms correct?
MM Parallel Loops with Reduction

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

$\textbf{MM-ijk}(A, B, C, n)$

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

$\textbf{MM-jki}(A, B, C, n)$

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

$\textbf{MM-ikj}(A, B, C, n)$

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

$\textbf{MM-kij}(A, B, C, n)$

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

$\textbf{MM-kji}(A, B, C, n)$

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

- The $k$-loop is reduced
- Are these algorithms correct? Yes! (no race conditions)
## MM Parallel Loops with Reduction: Complexity

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>(T_1(n))</th>
<th>(T_\infty(n))</th>
<th>(E_1(n))</th>
<th>(S_\infty(n))</th>
<th>(Q_1(n))</th>
</tr>
</thead>
<tbody>
<tr>
<td>MM-ijk</td>
<td>(\Theta(n^3))</td>
<td>(\Theta(\log n))</td>
<td>(\Theta(\log n))</td>
<td>(\Theta(n^3))</td>
<td>(\Theta(n^3))</td>
</tr>
<tr>
<td>MM-ikj</td>
<td>(\Theta(n^3))</td>
<td>(\Theta(\log n))</td>
<td>(\Theta(\log n))</td>
<td>(\Theta(n^3))</td>
<td>(\Theta\left(\frac{n^3}{B} + n^2\right))</td>
</tr>
<tr>
<td>MM-jik</td>
<td>(\Theta(n^3))</td>
<td>(\Theta(\log n))</td>
<td>(\Theta(\log n))</td>
<td>(\Theta(n^3))</td>
<td>(\Theta(n^3))</td>
</tr>
<tr>
<td>MM-jki</td>
<td>(\Theta(n^3))</td>
<td>(\Theta(\log n))</td>
<td>(\Theta(\log n))</td>
<td>(\Theta(n^3))</td>
<td>(\Theta(n^3))</td>
</tr>
<tr>
<td>MM-kij</td>
<td>(\Theta(n^3))</td>
<td>(\Theta(\log n))</td>
<td>(\Theta(\log n))</td>
<td>(\Theta(n^3))</td>
<td>(\Theta(n^3))</td>
</tr>
<tr>
<td>MM-kji</td>
<td>(\Theta(n^3))</td>
<td>(\Theta(\log n))</td>
<td>(\Theta(\log n))</td>
<td>(\Theta(n^3))</td>
<td>(\Theta(n^3))</td>
</tr>
</tbody>
</table>
### MM Parallel Loops with Reduction: Complexity

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>$T_1(n)$</th>
<th>$T_\infty(n)$</th>
<th>$E_1(n)$</th>
<th>$S_\infty(n)$</th>
<th>$Q_1(n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MM-ijk</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(n^3)$</td>
</tr>
<tr>
<td>MM-ikj</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta\left(\frac{n^3}{B} + n^2\right)$</td>
</tr>
<tr>
<td>MM-jik</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(n^3)$</td>
</tr>
<tr>
<td>MM-jki</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(n^3)$</td>
</tr>
<tr>
<td>MM-kij</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(n^3)$</td>
</tr>
<tr>
<td>MM-kji</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(n^3)$</td>
</tr>
</tbody>
</table>

### Advantages
- Minimum depth
- Easy to implement

### Disadvantages
- Maximum space
- Lack temporal cache locality
### MM Parallel Loops with Reduction: Complexity

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>$T_1(n)$</th>
<th>$T_\infty(n)$</th>
<th>$E_1(n)$</th>
<th>$S_\infty(n)$</th>
<th>$Q_1(n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MM-ijk</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(n^3)$</td>
</tr>
<tr>
<td>MM-ikj</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta\left(\frac{n^3}{B} + n^2\right)$</td>
</tr>
<tr>
<td>MM-jik</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(n^3)$</td>
</tr>
<tr>
<td>MM-jki</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(n^3)$</td>
</tr>
<tr>
<td>MM-kij</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(n^3)$</td>
</tr>
<tr>
<td>MM-kji</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(n^3)$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum depth</td>
<td>Maximum space</td>
</tr>
<tr>
<td>Easy to implement</td>
<td>Lack temporal cache locality</td>
</tr>
</tbody>
</table>

> How can we exploit the temporal data locality?
## MM Parallel Loops with Reduction: Complexity

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>$T_1(n)$</th>
<th>$T_\infty(n)$</th>
<th>$E_1(n)$</th>
<th>$S_\infty(n)$</th>
<th>$Q_1(n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MM-ijk</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(n^3)$</td>
</tr>
<tr>
<td>MM-ikj</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta\left(\frac{n^3}{B} + n^2\right)$</td>
</tr>
<tr>
<td>MM-jik</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(n^3)$</td>
</tr>
<tr>
<td>MM-jki</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(n^3)$</td>
</tr>
<tr>
<td>MM-kij</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(n^3)$</td>
</tr>
<tr>
<td>MM-kji</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(\log n)$</td>
<td>$\Theta(n^3)$</td>
<td>$\Theta(n^3)$</td>
</tr>
</tbody>
</table>

### Advantages
- Minimum depth
- Easy to implement

### Disadvantages
- Maximum space
- 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

\[ MM-\text{Tiled}(A, B, C, n, M) \]

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. \( \text{MM-Loop}(A[I, K], B[K, J], C[I, J], n/r) \)
## 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. \( \text{MM-Loop}(A[I, K], B[K, J], C[I, J], n/r) \)

### Correctness

- Similar to our previous proofs
MM Tiled Loops

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

1. \(r \leftarrow \frac{n}{\sqrt{M}}\)
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. \(\textbf{MM-Loop}(A[I, K], B[K, J], C[I, J], n/r)\)
   

**Correctness**

- Similar to our previous proofs

<table>
<thead>
<tr>
<th>(T_1(n))</th>
<th>(T_\infty(n))</th>
<th>(E_1(n))</th>
<th>(S_\infty(n))</th>
<th>(Q_1(n))</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\Theta(n^3))</td>
<td>(\Theta(n^3))</td>
<td>(\Theta(1))</td>
<td>(\Theta(n^2))</td>
<td>(\Theta\left(\frac{n^3}{B\sqrt{M}} + \frac{n^2}{B} + n\right))</td>
</tr>
</tbody>
</table>
MM Tiled Loops

\[ \text{MM-Tiled}(A, B, C, n, M) \]

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. \( \text{MM-Loop}(A[I, K], B[K, J], C[I, J], n/r) \)

Correctness

▸ Similar to our previous proofs

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

▸ How can we parallelize this algorithm?
MM Parallel Tiled Loops

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

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($A[I, K], B[K, J], C[I, J], n/r$)

The $⟨I,J,K⟩$ and $⟨i,j,k⟩$ 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 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**\((A[I, K], B[K, J], C[I, J], n/r)\)

<table>
<thead>
<tr>
<th>( T_1(n) )</th>
<th>( T_\infty(n) )</th>
<th>( E_1(n) )</th>
<th>( S_\infty(n) )</th>
<th>( Q_1(n) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Theta (n^3) )</td>
<td>( \Theta (n) )</td>
<td>( \Theta (1) )</td>
<td>( \Theta (n^2) )</td>
<td>( \Theta \left( \frac{n^3}{B \sqrt{M}} + \frac{n^2}{B} + n \right) )</td>
</tr>
</tbody>
</table>
MM Parallel Tiled Loops

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

1. \(r \leftarrow n/\sqrt{M}\)
2. \textbf{parallel} for \(I \leftarrow 1\) to \(r\) do
3. \textbf{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**\((A[I, K], B[K, J], C[I, J], n/r)\)


\[
\begin{array}{cccccc}
T_1(n) & T_\infty(n) & E_1(n) & S_\infty(n) & Q_1(n) \\
\Theta (n^3) & \Theta (n) & \Theta (1) & \Theta (n^2) & \Theta \left(\frac{n^3}{B\sqrt{M}} + \frac{n^2}{B} + n\right)
\end{array}
\]

- 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**

\[ \text{MM-Tiled-Parallel}(A, B, C, n, M) \]

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

<table>
<thead>
<tr>
<th>( T_1(n) )</th>
<th>( T_\infty(n) )</th>
<th>( E_1(n) )</th>
<th>( S_\infty(n) )</th>
<th>( Q_1(n) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Theta(n^3) )</td>
<td>( \Theta(n) )</td>
<td>( \Theta(1) )</td>
<td>( \Theta(n^2) )</td>
<td>( \Theta\left(\frac{n^3}{B \sqrt{M}} + \frac{n^2}{B} + n\right) )</td>
</tr>
</tbody>
</table>

- 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**($A[I, K], B[K, J], C[I, J], n/r$)
   

<table>
<thead>
<tr>
<th>$T_1(n)$</th>
<th>$T_\infty(n)$</th>
<th>$E_1(n)$</th>
<th>$S_\infty(n)$</th>
<th>$Q_1(n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Theta(n^3)$</td>
<td>$\Theta(n)$</td>
<td>$\Theta(1)$</td>
<td>$\Theta(n^2)$</td>
<td>$\Theta\left(\frac{n^3}{B \sqrt{M}} + \frac{n^2}{B} + n\right)$</td>
</tr>
</tbody>
</table>

- 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!
\[
\begin{array}{cc}
C_{11} & C_{12} \\
C_{21} & C_{22}
\end{array}
\]
\[=\]
\[
\begin{array}{cc}
A_{11} & A_{12} \\
A_{21} & A_{22}
\end{array}
\times
\begin{array}{cc}
B_{11} & B_{12} \\
B_{21} & B_{22}
\end{array}
\]
\[=\]
\[
\begin{array}{cc}
A_{11}B_{11} + A_{12}B_{21} & A_{11}B_{12} + A_{12}B_{22} \\
A_{21}B_{11} + A_{22}B_{21} & A_{21}B_{12} + A_{22}B_{22}
\end{array}
\]
**MM D&C**

<table>
<thead>
<tr>
<th>MM$(A, B, C, n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. if $n = 1$ then</td>
</tr>
<tr>
<td>2. <strong>MM-Loop</strong>(A, B, C, n)</td>
</tr>
<tr>
<td>3. else</td>
</tr>
<tr>
<td>4. MM$(A_{11}, B_{11}, C_{11}, n/2)$</td>
</tr>
<tr>
<td>5. MM$(A_{12}, B_{21}, C_{11}, n/2)$</td>
</tr>
<tr>
<td>6. MM$(A_{11}, B_{12}, C_{12}, n/2)$</td>
</tr>
<tr>
<td>7. MM$(A_{12}, B_{22}, C_{12}, n/2)$</td>
</tr>
<tr>
<td>8. MM$(A_{21}, B_{11}, C_{21}, n/2)$</td>
</tr>
<tr>
<td>9. MM$(A_{22}, B_{21}, C_{21}, n/2)$</td>
</tr>
<tr>
<td>10. MM$(A_{21}, B_{12}, C_{22}, n/2)$</td>
</tr>
<tr>
<td>11. MM$(A_{22}, B_{22}, C_{22}, n/2)$</td>
</tr>
</tbody>
</table>

**Complexity**

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

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

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

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

▶ How can we parallelize this algorithm?
1. if \( n = 1 \) then
2. \textbf{MM-Loop}(A, B, C, n)
3. else
4. \textbf{MM}(A_{11}, B_{11}, C_{11}, n/2)
5. \textbf{MM}(A_{12}, B_{21}, C_{11}, n/2)
6. \textbf{MM}(A_{11}, B_{12}, C_{12}, n/2)
7. \textbf{MM}(A_{12}, B_{22}, C_{12}, n/2)
8. \textbf{MM}(A_{21}, B_{11}, C_{21}, n/2)
9. \textbf{MM}(A_{22}, B_{21}, C_{21}, n/2)
10. \textbf{MM}(A_{21}, B_{12}, C_{22}, n/2)
11. \textbf{MM}(A_{22}, B_{22}, C_{22}, n/2)

**Complexity**

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

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

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

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

\[
8Q_1\left(\frac{n}{2}\right) + \Theta(1) & \text{if } n^2 > \alpha M.
\]

<table>
<thead>
<tr>
<th>( T_1(n) )</th>
<th>( T_\infty(n) )</th>
<th>( E_1(n) )</th>
<th>( S_\infty(n) )</th>
<th>( Q_1(n) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Theta(n^3) )</td>
<td>( \Theta(n^3) )</td>
<td>( \Theta(\log n) )</td>
<td>( \Theta(n^2) )</td>
<td>( \Theta\left(\frac{n^3}{B\sqrt{M}} + \frac{n^2}{B} + n\right) )</td>
</tr>
</tbody>
</table>
How can we parallelize this algorithm?
**MM Parallel D&C**

<table>
<thead>
<tr>
<th>MM((A, B, C, n))</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. if (n = 1) then <strong>MM-Loop</strong>((A, B, C, n))</td>
</tr>
<tr>
<td>2. else</td>
</tr>
<tr>
<td>3. <strong>parallel:</strong> MM((A_{11}, B_{11}, C_{11}, n/2)), MM((A_{11}, B_{12}, C_{12}, n/2))</td>
</tr>
<tr>
<td>MM((A_{21}, B_{11}, C_{21}, n/2)), MM((A_{21}, B_{12}, C_{22}, n/2))</td>
</tr>
<tr>
<td>4. <strong>parallel:</strong> MM((A_{12}, B_{21}, C_{11}, n/2)), MM((A_{12}, B_{22}, C_{12}, n/2))</td>
</tr>
<tr>
<td>MM((A_{22}, B_{21}, C_{21}, n/2)), MM((A_{22}, B_{22}, C_{22}, n/2))</td>
</tr>
</tbody>
</table>

\[
\begin{array}{cc}
C_{11} & C_{12} \\
C_{21} & C_{22}
\end{array}
= \begin{array}{cc}
A_{11} & A_{12} \\
A_{21} & A_{22}
\end{array}
\times \begin{array}{cc}
B_{11} & B_{12} \\
B_{21} & B_{22}
\end{array}
= \begin{array}{cc}
A_{11}B_{11} + A_{12}B_{21} & A_{11}B_{12} + A_{12}B_{22} \\
A_{21}B_{11} + A_{22}B_{21} & A_{21}B_{12} + A_{22}B_{22}
\end{array}
\]
**MM Parallel D&C**

<table>
<thead>
<tr>
<th>MM$(A, B, C, n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. if $n = 1$ then MM-Loop$(A, B, C, n)$</td>
</tr>
<tr>
<td>2. else</td>
</tr>
<tr>
<td>3. parallel: MM$(A_{11}, B_{11}, C_{11}, n/2)$, MM$(A_{11}, B_{12}, C_{12}, n/2)$</td>
</tr>
<tr>
<td>MM$(A_{21}, B_{11}, C_{21}, n/2)$, MM$(A_{21}, B_{12}, C_{22}, n/2)$</td>
</tr>
<tr>
<td>4. parallel: MM$(A_{12}, B_{21}, C_{11}, n/2)$, MM$(A_{12}, B_{22}, C_{12}, n/2)$</td>
</tr>
<tr>
<td>MM$(A_{22}, B_{21}, C_{21}, n/2)$, MM$(A_{22}, B_{22}, C_{22}, n/2)$</td>
</tr>
</tbody>
</table>

**Complexity**

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

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

\[
T_\infty(n) = \begin{cases} 
\Theta(1) & \text{if } n = 1, \\
2T_\infty\left(\frac{n}{2}\right) + \Theta(1) & \text{if } n > 1. 
\end{cases}
\]

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

How can we further improve parallelism? Reduction! i.e., using extra space
MM Parallel D&C

\[\text{MM}(A, B, C, n)\]

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

**Complexity**

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

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

\[T_\infty(n) = \begin{cases} 
\Theta(1) & \text{if } n = 1, \\
2T_\infty\left(\frac{n}{2}\right) + \Theta(1) & \text{if } n > 1.
\end{cases}\]

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

<table>
<thead>
<tr>
<th>(T_1(n))</th>
<th>(T_\infty(n))</th>
<th>(E_1(n))</th>
<th>(S_\infty(n))</th>
<th>(Q_1(n))</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\Theta(n^3))</td>
<td>(\Theta(n))</td>
<td>(\Theta(\log n))</td>
<td>(\Theta(n^2))</td>
<td>(\Theta\left(\frac{n^3}{B\sqrt{M}} + \frac{n^2}{B} + n\right))</td>
</tr>
</tbody>
</table>
**MM Parallel D&C**

<table>
<thead>
<tr>
<th>MM(A, B, C, n)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. if ( n = 1 ) then ( \text{MM-Loop}(A, B, C, n) )</td>
</tr>
<tr>
<td>2. else</td>
</tr>
<tr>
<td>3. parallel: ( \text{MM}(A_{11}, B_{11}, C_{11}, n/2), \text{MM}(A_{11}, B_{12}, C_{12}, n/2) )</td>
</tr>
<tr>
<td>( \text{MM}(A_{21}, B_{11}, C_{21}, n/2), \text{MM}(A_{21}, B_{12}, C_{22}, n/2) )</td>
</tr>
<tr>
<td>4. parallel: ( \text{MM}(A_{12}, B_{21}, C_{11}, n/2), \text{MM}(A_{12}, B_{22}, C_{12}, n/2) )</td>
</tr>
<tr>
<td>( \text{MM}(A_{22}, B_{21}, C_{21}, n/2), \text{MM}(A_{22}, B_{22}, C_{22}, n/2) )</td>
</tr>
</tbody>
</table>

**Complexity**

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

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

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

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

<table>
<thead>
<tr>
<th>( T_1(n) )</th>
<th>( T_\infty(n) )</th>
<th>( E_1(n) )</th>
<th>( S_\infty(n) )</th>
<th>( Q_1(n) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Theta(n^3) )</td>
<td>( \Theta(n) )</td>
<td>( \Theta(\log n) )</td>
<td>( \Theta(n^2) )</td>
<td>( \Theta\left(\frac{n^3}{B\sqrt{M}} + \frac{n^2}{B} + n\right) )</td>
</tr>
</tbody>
</table>

▶ How can we further improve parallelism?
**MM Parallel D&C**

<table>
<thead>
<tr>
<th><strong>MM</strong>(<em>A, B, C, n</em>)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. if <em>n</em> = 1 then <strong>MM-Loop</strong>(<em>A, B, C, n</em>)</td>
</tr>
<tr>
<td>2. else</td>
</tr>
<tr>
<td>3. parallel: <strong>MM</strong>(<em>A_11, B_11, C_11, n/2</em>), <strong>MM</strong>(<em>A_11, B_12, C_12, n/2</em>), <strong>MM</strong>(<em>A_21, B_11, C_21, n/2</em>), <strong>MM</strong>(<em>A_21, B_12, C_22, n/2</em>)</td>
</tr>
<tr>
<td>4. parallel: <strong>MM</strong>(<em>A_12, B_21, C_11, n/2</em>), <strong>MM</strong>(<em>A_12, B_22, C_12, n/2</em>), <strong>MM</strong>(<em>A_22, B_21, C_21, n/2</em>), <strong>MM</strong>(<em>A_22, B_22, C_22, n/2</em>)</td>
</tr>
</tbody>
</table>

**Complexity**

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

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

\[
T_\infty(n) = \begin{cases} 
  \Theta(1) & \text{if } n = 1, \\
  2T_\infty\left(\frac{n}{2}\right) + \Theta(1) & \text{if } n > 1. 
\end{cases}
\]

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

<table>
<thead>
<tr>
<th><strong>T_1(n)</strong></th>
<th><strong>T_\infty(n)</strong></th>
<th><strong>E_1(n)</strong></th>
<th><strong>S_\infty(n)</strong></th>
<th><strong>Q_1(n)</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>(\Theta(n^3))</td>
<td>(\Theta(n))</td>
<td>(\Theta(\log n))</td>
<td>(\Theta(n^2))</td>
<td>(\Theta\left(\frac{n^3}{B\sqrt{M}} + \frac{n^2}{B} + n\right))</td>
</tr>
</tbody>
</table>

▶ How can we further improve parallelism?
Reduction! i.e., using extra space
**MM Parallel Not-In-Place D&C**

\[
\text{MM}(A, B, C, n)
\]

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

**Complexity**

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

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

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

\[
Q_1(n) = \Theta(n^3)
\]

How can we improve cache complexity?

In-place tiled basecase!
\textbf{MM Parallel Not-In-Place D\&C}

\begin{table}[h]
\begin{tabular}{|c|}
\hline
\textbf{MM}(A, B, C, n) \\
\hline
1. if \( n = 1 \) then \textbf{MM-Loop}(A, B, C, n) \\
2. else \\
3. \textbf{parallel}: \textbf{MM}(A_{11}, B_{11}, C_{11}, n/2), \textbf{MM}(A_{11}, B_{12}, C_{12}, n/2) \\
    \textbf{MM}(A_{21}, B_{11}, C_{21}, n/2), \textbf{MM}(A_{21}, B_{12}, C_{22}, n/2) \\
    \textbf{MM}(A_{12}, B_{21}, D_{11}, n/2), \textbf{MM}(A_{12}, B_{22}, D_{12}, n/2) \\
    \textbf{MM}(A_{22}, B_{21}, D_{21}, n/2), \textbf{MM}(A_{22}, B_{22}, D_{22}, n/2) \\
4. \( C \leftarrow \text{Parallel-Matrix-Sum}(C, D) \quad \triangleright \ C \leftarrow C + D \) \\
\hline
\end{tabular}
\end{table}

\textbf{Complexity}

\begin{tabular}{|l|}
\hline
\textbf{\( T_1(n) = \)} & \text{if} \( n = 1 \), \\
\hline
\( \Theta (1) \) & \( \Theta (1) \) \\
\hline
\( 8T_1(n/2) + \Theta (1) \) & \( 8T_1(n/2) + \Theta (1) \) \\
\hline
\end{tabular}

\begin{tabular}{|l|}
\hline
\textbf{\( S_\infty(n) = \)} & \text{if} \( n = 1 \), \\
\hline
\( \Theta (1) \) & \( \Theta (1) \) \\
\hline
\( 8S_\infty(n/2) + \Theta (n^2) \) & \( 8S_\infty(n/2) + \Theta (n^2) \) \\
\hline
\end{tabular}

\begin{tabular}{|l|}
\hline
\textbf{\( T_\infty(n) = \)} & \text{if} \( n = 1 \), \\
\hline
\( \Theta (1) \) & \( \Theta (1) \) \\
\hline
\( T_\infty(n/2) + \Theta (\log n) \) & \( T_\infty(n/2) + \Theta (\log n) \) \\
\hline
\end{tabular}

\begin{tabular}{|l|}
\hline
\textbf{\( Q_1(n) = \)} & \text{if} \( n = 1 \), \\
\hline
\( \Theta (n^3) \) & \( \Theta (n^3) \) \\
\hline
\end{tabular}
MM Parallel Not-In-Place D&C

\[ \text{MM}(A, B, C, n) \]

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

**Complexity**

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

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

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

\[ Q_1(n) = \Theta\left(n^3\right) \]

<table>
<thead>
<tr>
<th>( T_1(n) )</th>
<th>( T_\infty(n) )</th>
<th>( E_1(n) )</th>
<th>( S_\infty(n) )</th>
<th>( Q_1(n) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Theta\left(n^3\right) )</td>
<td>( \Theta\left(\log^2 n\right) )</td>
<td>( \Theta\left(n^2\right) )</td>
<td>( \Theta\left(n^3\right) )</td>
<td>( \Theta\left(n^3\right) )</td>
</tr>
</tbody>
</table>
MM Parallel Not-In-Place D&C

\[ \text{MM}(A, B, C, n) \]

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

**Complexity**

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

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

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

\[
Q_1(n) = \Theta(n^3)
\]

<table>
<thead>
<tr>
<th>( T_1(n) )</th>
<th>( T_\infty(n) )</th>
<th>( E_1(n) )</th>
<th>( S_{\infty}(n) )</th>
<th>( Q_1(n) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Theta(n^3) )</td>
<td>( \Theta(\log^2 n) )</td>
<td>( \Theta(n^2) )</td>
<td>( \Theta(n^3) )</td>
<td>( \Theta(n^3) )</td>
</tr>
</tbody>
</table>

▶ How can we improve cache complexity?
MM Parallel Not-In-Place D&C

\[ MM(A, B, C, n) \]

1. if \( n = 1 \) then \( MM\text{-Loop}(A, B, C, n) \)
2. else
3. \textbf{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. \( C \leftarrow \text{Parallel-Matrix-Sum}(C, D) \) \( \triangleright C \leftarrow C + D \)

\[ \text{Complexity} \]

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

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

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

\[
Q_1(n) = \Theta(n^3)
\]

\[
\begin{array}{|c|c|c|c|c|}
\hline
T_1(n) & T_\infty(n) & E_1(n) & S_\infty(n) & Q_1(n) \\
\hline
\Theta(n^3) & \Theta(\log^2 n) & \Theta(n^2) & \Theta(n^3) & \Theta(n^3) \\
\hline
\end{array}
\]

▶ How can we improve cache complexity?
In-place tiled basecase!
MM Parallel Not-In-Place D&C

<table>
<thead>
<tr>
<th>MM$(A, B, C, n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. if $n = \sqrt{M}$ then MM-Base$(A, B, C, n)$ $\triangleright$ Parallel in-place MM</td>
</tr>
<tr>
<td>2. else</td>
</tr>
<tr>
<td>3. parallel: MM$(A_{11}, B_{11}, C_{11}, n/2)$, MM$(A_{11}, B_{12}, C_{12}, n/2)$</td>
</tr>
<tr>
<td>MM$(A_{21}, B_{11}, C_{21}, n/2)$, MM$(A_{21}, B_{12}, C_{22}, n/2)$</td>
</tr>
<tr>
<td>MM$(A_{12}, B_{21}, D_{11}, n/2)$, MM$(A_{12}, B_{22}, D_{12}, n/2)$</td>
</tr>
<tr>
<td>MM$(A_{22}, B_{21}, D_{21}, n/2)$, MM$(A_{22}, B_{22}, D_{22}, n/2)$</td>
</tr>
<tr>
<td>4. $C \leftarrow$ Parallel-Matrix-Sum$(C, D)$ $\triangleright$ $C \leftarrow C + D$</td>
</tr>
</tbody>
</table>
MM Parallel Not-In-Place D&C

\[ \text{MM}(A, B, C, n) \]

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

**Complexity**

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

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

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

\[ Q_1(n) = \begin{cases} 
\Theta\left(\frac{n^2}{B} + n\right) & \text{if } n^2 \leq \alpha M, \\
8Q_1\left(\frac{n}{2}\right) + \Theta\left(\frac{n^2}{B} + n\right) & \text{if } n^2 > \alpha M.
\end{cases}
\]
**, MM Parallel Not-In-Place D&C**

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

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

**Complexity**

<table>
<thead>
<tr>
<th>( T_1(n) )</th>
<th>( T_\infty(n) )</th>
<th>( E_1(n) )</th>
<th>( S_\infty(n) )</th>
<th>( Q_1(n) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Theta(n^3) )</td>
<td>( \Theta(\sqrt{M} + \log n) )</td>
<td>( \Theta(n^2) )</td>
<td>( \Theta\left(\frac{n^3}{\sqrt{M}}\right) )</td>
<td>( \Theta\left(\frac{n^3}{B \sqrt{M}} + \frac{n^2}{B} + n\right) )</td>
</tr>
</tbody>
</table>
### MM Parallel Not-In-Place D&C

**MM** \((A, B, C, n)\)

1. if \(n = \sqrt{M}\) then **MM-Base** \((A, B, C, n)\)  \(\triangleright\) 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. \(C \leftarrow \text{Parallel-Matrix-Sum}(C, D)\)  \(\triangleright\) \(C \leftarrow C + D\)

### Complexity

**T_1(n)** = \[
\begin{cases}
  \Theta(n^3) & \text{if } n \leq \alpha \sqrt{M}, \\
  8T_1(n/2) + \Theta(1) & \text{if } n > \alpha \sqrt{M}.
\end{cases}
\]

**S_\infty(n)** = \[
\begin{cases}
  \Theta(n^2) & \text{if } n^2 \leq \alpha M, \\
  8S_\infty(n/2) + \Theta(n^2) & \text{if } n^2 > \alpha M.
\end{cases}
\]

**T_\infty(n)** = \[
\begin{cases}
  \Theta(n) & \text{if } n \leq \alpha \sqrt{M}, \\
  T_\infty(n/2) + \Theta(\log n) & \text{if } n > \alpha \sqrt{M}.
\end{cases}
\]

**Q_1(n)** = \[
\begin{cases}
  \Theta(n^2_B + n) & \text{if } n^2 \leq \alpha M, \\
  8Q_1(n/2) + \Theta(n^2_B + n) & \text{if } n^2 > \alpha M.
\end{cases}
\]

<table>
<thead>
<tr>
<th>(T_1(n))</th>
<th>(T_\infty(n))</th>
<th>(E_1(n))</th>
<th>(S_\infty(n))</th>
<th>(Q_1(n))</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\Theta(n^3))</td>
<td>(\Theta(\sqrt{M} + \log n))</td>
<td>(\Theta(n^2))</td>
<td>(\Theta(n^3_M))</td>
<td>(\Theta(n^3_B \sqrt{M} + n^2_B + n))</td>
</tr>
</tbody>
</table>

**How can we reduce work?**
MM Parallel Not-In-Place D&C

\[ \text{MM}(A, B, C, n) \]

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

**Complexity**

\[
\begin{align*}
T_1(n) &= \begin{cases} 
\Theta(n^3) & \text{if } n \leq \alpha \sqrt{M}, \\
8T_1\left(\frac{n}{2}\right) + \Theta(1) & \text{if } n > \alpha \sqrt{M}.
\end{cases} \\
S_\infty(n) &= \begin{cases} 
\Theta(n^2) & \text{if } n^2 \leq \alpha M, \\
8S_\infty\left(\frac{n}{2}\right) + \Theta(n^2) & \text{if } n^2 > \alpha M.
\end{cases} \\
T_\infty(n) &= \begin{cases} 
\Theta(n) & \text{if } n \leq \alpha \sqrt{M}, \\
T_\infty\left(\frac{n}{2}\right) + \Theta(\log n) & \text{if } n > \alpha \sqrt{M}.
\end{cases} \\
Q_1(n) &= \begin{cases} 
\Theta\left(\frac{n^2}{B} + n\right) & \text{if } n^2 \leq \alpha M, \\
8Q_1\left(\frac{n}{2}\right) + \Theta\left(\frac{n^2}{B} + n\right) & \text{if } n^2 > \alpha M.
\end{cases}
\end{align*}
\]

---

\[ \begin{array}{|c|c|c|c|c|}
\hline
T_1(n) & T_\infty(n) & E_1(n) & S_\infty(n) & Q_1(n) \\
\hline
\Theta(n^3) & \Theta\left(\sqrt{M} + \log n\right) & \Theta(n^2) & \Theta\left(\frac{n^3}{\sqrt{M}}\right) & \Theta\left(\frac{n^3}{B \sqrt{M}} + \frac{n^2}{B} + n\right) \\
\hline
\end{array} \]

\( \triangleright \) How can we reduce work? Strassen’s algorithm!
### Volker Strassen’s MM D&C: Core Idea

<table>
<thead>
<tr>
<th>Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is there a strategy to perform multiplication of two complex numbers with only 3 multiplications?</td>
</tr>
<tr>
<td>((a + ib)(c + id) = (ac - bd) + i(bc + ad))</td>
</tr>
</tbody>
</table>
Volker Strassen’s MM D&C: Core Idea

<table>
<thead>
<tr>
<th>Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is there a strategy to perform multiplication of two complex numbers with only 3 multiplications?</td>
</tr>
<tr>
<td>[(a + ib)(c + id) = (ac - bd) + i(bc + ad)]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Solution 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Let (x = bd), (y = ac), and (z = (a + b)(c + d)).</td>
</tr>
<tr>
<td>Then, real part = (y - x) and imaginary part = (z - x - y).</td>
</tr>
</tbody>
</table>
**Volker Strassen’s MM D&C: Core Idea**

<table>
<thead>
<tr>
<th>Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is there a strategy to perform multiplication of two complex numbers with only 3 multiplications?</td>
</tr>
<tr>
<td>((a + ib)(c + id) = (ac - bd) + i(bc + ad))</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Solution 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Let (x = bd,) (y = ac,) and (z = (a + b)(c + d)).</td>
</tr>
<tr>
<td>Then, real part = (y - x) and imaginary part = (z - x - y).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Solution 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Let (x = c(a + b),) (y = a(d - c),) and (z = b(c + d)).</td>
</tr>
<tr>
<td>Then, real part = (x - z) and imaginary part = (x + y).</td>
</tr>
</tbody>
</table>
**Volker Strassen’s MM D&C: Core Idea**

<table>
<thead>
<tr>
<th>Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is there is a strategy to perform multiplication of two complex numbers with only 3 multiplications?</td>
</tr>
</tbody>
</table>

\[(a + ib)(c + id) = (ac - bd) + i(bc + ad)\]

<table>
<thead>
<tr>
<th>Solution 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Let (x = bd, y = ac, ) and (z = (a + b)(c + d).) Then, real part = (y - x) and imaginary part = (z - x - y.)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Solution 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Let (x = c(a + b), y = a(d - c), ) and (z = b(c + d).) Then, real part = (x - z) and imaginary part = (x + y.)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Solution 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Let (x = c(a + b), y = a(c - d), ) and (z = d(a - b).) Then, real part = (y + z) and imaginary part = (x - y.)</td>
</tr>
</tbody>
</table>
Volker Strassen’s MM D&C: Core Idea

<table>
<thead>
<tr>
<th>Problem</th>
<th>Traditional</th>
<th>Solution 1</th>
<th>Solution 2</th>
<th>Solution 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complex number mult.</td>
<td>4 mults</td>
<td>3 mults</td>
<td>3 mults</td>
<td>3 mults</td>
</tr>
<tr>
<td></td>
<td>2 adds</td>
<td>5 adds</td>
<td>5 adds</td>
<td>5 adds</td>
</tr>
</tbody>
</table>
### Volker Strassen’s MM D&C: Core Idea

<table>
<thead>
<tr>
<th>Problem</th>
<th>Traditional</th>
<th>Solution 1</th>
<th>Solution 2</th>
<th>Solution 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complex number mult.</td>
<td>4 mults</td>
<td>3 mults</td>
<td>3 mults</td>
<td>3 mults</td>
</tr>
<tr>
<td></td>
<td>2 adds</td>
<td>5 adds</td>
<td>5 adds</td>
<td>5 adds</td>
</tr>
</tbody>
</table>

\[
\begin{array}{cccc}
A_{11} & A_{12} & & \\
A_{21} & A_{22} & & \\
& & B_{11} & B_{12} \\
& & B_{21} & B_{22} \\
\end{array}
\times
\begin{array}{cccc}
A_{11}B_{11} + A_{12}B_{21} & A_{11}B_{12} + A_{12}B_{22} \\
A_{21}B_{11} + A_{22}B_{21} & A_{21}B_{12} + A_{22}B_{22} \\
\end{array}
= \begin{array}{cccc}
C_{11} & C_{12} & & \\
C_{21} & C_{22} & & \\
\end{array}
\]

<table>
<thead>
<tr>
<th>Problem</th>
<th>Traditional</th>
<th>Strassen</th>
<th>Winograd</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2 \times 2$ MM</td>
<td>8 mults</td>
<td>7 mults</td>
<td>7 mults</td>
</tr>
<tr>
<td></td>
<td>4 adds</td>
<td>18 adds</td>
<td>15 adds</td>
</tr>
<tr>
<td>$n \times n$ MM</td>
<td>$n^3$ mults</td>
<td>$n^{\log_2 7}$ mults</td>
<td>$n^{\log_2 7}$ mults</td>
</tr>
<tr>
<td></td>
<td>$(n^3 - n^2)$ adds</td>
<td>$(6n^{\log_2 7} - 6n^2)$ adds</td>
<td>$(5n^{\log_2 7} - 5n^2)$ adds</td>
</tr>
</tbody>
</table>
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

\[
\text{MM-Strassen}(A, B, C, n)
\]

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**

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

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

\(= \Theta(n^{\log_2 7})\)

- How can we parallelize the algorithm?
- What are the complexities of the parallel algorithm?
Shmuel Winograd’s MM D&C

<table>
<thead>
<tr>
<th>MM-Winograd((A, B, C, n))</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. (R_1 \leftarrow A_{21} + A_{22})</td>
</tr>
<tr>
<td>2. (R_2 \leftarrow R_1 - A_{11})</td>
</tr>
<tr>
<td>3. (R_3 \leftarrow A_{11} - A_{21})</td>
</tr>
<tr>
<td>4. (R_4 \leftarrow A_{12} - R_2)</td>
</tr>
<tr>
<td>5. (R_5 \leftarrow B_{12} - B_{11})</td>
</tr>
<tr>
<td>6. (R_6 \leftarrow B_{22} - R_5)</td>
</tr>
<tr>
<td>7. (R_7 \leftarrow B_{22} - B_{12})</td>
</tr>
<tr>
<td>8. (R_8 \leftarrow R_6 - B_{21})</td>
</tr>
<tr>
<td>9. (P_1 \leftarrow R_2 \times R_6)</td>
</tr>
<tr>
<td>10. (P_2 \leftarrow A_{11} \times B_{11})</td>
</tr>
<tr>
<td>11. (P_3 \leftarrow A_{12} \times B_{21})</td>
</tr>
<tr>
<td>12. (P_4 \leftarrow R_3 \times R_7)</td>
</tr>
<tr>
<td>13. (P_5 \leftarrow R_1 \times R_5)</td>
</tr>
<tr>
<td>14. (P_6 \leftarrow R_4 \times B_{22})</td>
</tr>
<tr>
<td>15. (P_7 \leftarrow A_{22} \times R_8)</td>
</tr>
<tr>
<td>16. (V_1 \leftarrow P_1 + P_2)</td>
</tr>
<tr>
<td>17. (V_2 \leftarrow V_1 + P_4)</td>
</tr>
<tr>
<td>18. (C_{11} \leftarrow P_2 + P_3)</td>
</tr>
<tr>
<td>19. (C_{12} \leftarrow V_1 + P_5 + P_6)</td>
</tr>
<tr>
<td>20. (C_{21} \leftarrow V_2 - P_7)</td>
</tr>
<tr>
<td>21. (C_{22} \leftarrow V_2 + P_5)</td>
</tr>
</tbody>
</table>

Source: computer.org

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

How can we parallelize the 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. $C_{22} \leftarrow V_2 + P_5$

**Complexity**

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

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

Source: computer.org
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

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

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

> 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 $C[i, j]$

![Diagram of distributed setup with $r \times r$ processor grid, initial input distribution, and final output.]
**MM Distributed**

\[ \text{MM}(A, B, C, n, p) \]

1. \( r \leftarrow \sqrt{p}, \text{ where } p \leq n^2 \)
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 \( \sum_{k=1}^{r} A[i, k] \times B[k, j] \)

![Diagram](image-url)
**MM Distributed**

[Diagram showing the initial input distribution and broadcast rowwise process]

- **Initial input distribution**
  - Initial values: $A[1,1]$, $B[1,1]$

- **Broadcast rowwise**

  - $A[1,2]$, $B[1,2]$

  - $A[1,3]$, $B[1,3]$
MM Distributed

Broadcast columnwise

Final output
**MM Distributed: Complexity**

\[\text{MM}(A, B, C, n, p)\]

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 \( \sum_{k=1}^{r} A[i, k] \times B[k, j] \)

<table>
<thead>
<tr>
<th>Step</th>
<th>( T_{\text{comp}}(n) )</th>
<th>( T_{\text{comm}}(n) )</th>
<th>( S_{\max}(n) )</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BC-Row</strong> ((r))</td>
<td>(-)</td>
<td>( t_s \log r + t_w(r - 1) \left( \frac{n}{r} \cdot \frac{n}{r} \right) )</td>
<td>( r \left( \frac{n}{r} \cdot \frac{n}{r} \right) )</td>
</tr>
<tr>
<td><strong>BC-Col</strong> ((r))</td>
<td>(-)</td>
<td>( t_s \log r + t_w(r - 1) \left( \frac{n}{r} \cdot \frac{n}{r} \right) )</td>
<td>( r \left( \frac{n}{r} \cdot \frac{n}{r} \right) )</td>
</tr>
<tr>
<td>Compute ( C[i, j] )</td>
<td>( r \left( \frac{n}{r} \right)^3 )</td>
<td>(-)</td>
<td>( \left( \frac{n}{r} \cdot \frac{n}{r} \right) )</td>
</tr>
<tr>
<td><strong>Total</strong> ((r))</td>
<td>( \frac{n^3}{r^2} )</td>
<td>( 2 \left( t_s \log r + t_w(r - 1) \left( \frac{n}{r} \right)^2 \right) )</td>
<td>( (2r + 1) \left( \frac{n}{r} \right)^2 )</td>
</tr>
<tr>
<td><strong>Total</strong> ((r = \sqrt{p}))</td>
<td>( \frac{n^3}{p} )</td>
<td>( t_s \log p + 2t_w(\sqrt{p} - 1) \left( \frac{n^2}{p} \right) )</td>
<td>( (2 \sqrt{p} + 1) \left( \frac{n^2}{p} \right) )</td>
</tr>
</tbody>
</table>

➤ How can we improve space complexity?