Matrix Multiplication on the Connection Machine
Matrix Multiply: \( c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj} \)

- Systolic "Cannon's" Algorithm
  - align and rotate phases

- "Blocked" Data Movement
  - minimizes communication costs

- Optimum use of Floating Point Units
  - micro-coded local matrix multiply
Systolic Algorithm
Align Phase

- In row i moves i to left - In column j moves j up
- Aligns rows of A and columns of B so inner indices match
Systolic Algorithm
Rotate Phase

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>00 01 02 03</td>
<td>00 01 02 03</td>
</tr>
<tr>
<td>10 11 12 13</td>
<td>10 11 12 13</td>
</tr>
<tr>
<td>20 21 22 23</td>
<td>20 21 22 23</td>
</tr>
<tr>
<td>30 31 32 33</td>
<td>30 31 32 33</td>
</tr>
</tbody>
</table>

\[ X = C \times A \]

- In A all rows move 1 to left - In B all columns move 1 up
- Multiply, add and rotate n times
Systolic - Blocked Algorithm
Conforming blocks

\[
\begin{array}{cccc}
00 & 01 & 02 & 03 \\
10 & 11 & 12 & 13 \\
20 & 21 & 22 & 23 \\
30 & 31 & 32 & 33 \\
\end{array}
\times
\begin{array}{cccc}
00 & 01 & 02 & 03 \\
10 & 11 & 12 & 13 \\
20 & 21 & 22 & 23 \\
30 & 31 & 32 & 33 \\
\end{array}
= \begin{array}{cccc}
00 & 01 & 02 & 03 \\
10 & 11 & 12 & 13 \\
20 & 21 & 22 & 23 \\
30 & 31 & 32 & 33 \\
\end{array}
\times
\begin{array}{cccc}
00 & 01 \\
10 & 11 \\
20 & 21 \\
30 & 31 \\
\end{array}
= \begin{array}{cccc}
00 & 01 \\
10 & 11 \\
\end{array}

- Perform local matrix multiply on entire block
- Local matrix multiply uses micro-coded SAXPY
### Systolic - Blocked Algorithm

#### Non-conforming blocks

- Operate only on conforming part of blocks
- Rotate A and B unequally to allow use of all data
Performance of the matrix multiplication routine for square matrices of size $N$ as a function of the size of the Connection Machine.
Performance of the matrix multiplication routine for square matrices of size $N$ as a function of the size of the Connection Machine.
Peak Performance of the sprint node matrix multiplication kernel.

![Graph showing GFlops/s performance over Log(r) for a 64x256 matrix multiplication kernel.](image-url)