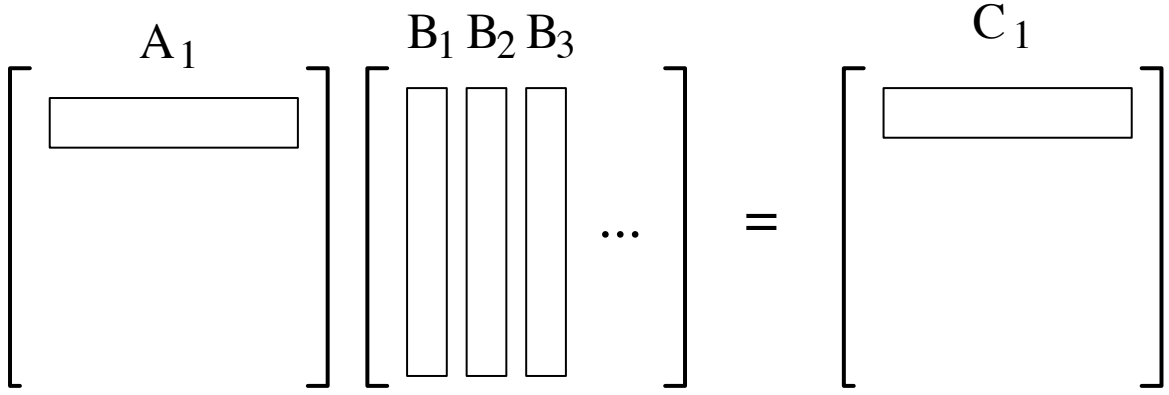
Matrix-Matrix Multiplication

Problem: $C = A * B$ where A and B are $n \times n$ matrices.

Sequential code:

```
for i = 1 to n do
    for j = 1 to n do
        sum = 0;
        for k = 1 to n do
              sum = sum + a[i,k] * b[k,j];
        c[i,j] = sum;
      endfor
   endfor
endfor
```

An example of $A * B$



$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} * \begin{pmatrix} 5 & 7 \\ 6 & 8 \end{pmatrix}$$

$$= \begin{pmatrix} 1*5+2*6 & 1*7+2*8 \\ 3*5+4*6 & 3*7+4*8 \end{pmatrix}$$

$$= \begin{pmatrix} 17 & 23 \\ 39 & 53 \end{pmatrix}$$

Task graph of $C = A * B$

Partitioned code:

for $i = 1$ **to** n **do**

T_i :

for $j = 1$ **to** n **do**

$sum = 0;$

for $k = 1$ to n **do**

$sum = sum + a[i,k] * b[k,j];$

endfor

$c[i,j] = sum;$

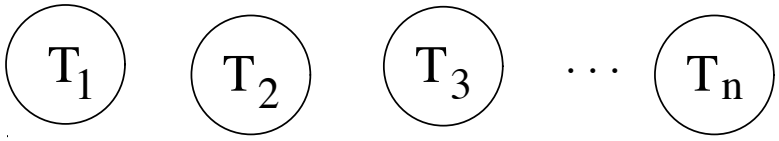
endfor

endfor

T_i : Read row A_i and matrix B.

Write row C_i

Task graph:



Task and data mapping for $C = A * B$

SPMD code:

for $i = 1$ **to** n
    if proc_map(i)=me do T_i

Data mapping:

A is partitioned using rowwise block mapping

C is partitioned using rowwise block mapping

B is duplicated to all processors

Changes in T_i's code:

$$a_{ik} \longrightarrow a_{local(i)k}$$

$$c_{ij} \longrightarrow c_{local(i)j}$$

Parallel SPMD code of $C = A * B$

for $i = 1$ **to** n **do**
  if proc_map(i)=me do
    for $j = 1$ **to** n **do**
      $sum = 0$;
      for $k = 1$ to n **do**
        $sum = sum + a[local(i), k] * b[k, j]$;
      endfor
      $c[local(i), j] = sum$;
    endfor
  endif
endfor

Parallel algorithm with 1D partitioning

Partitioned code:

for $i = 1$ **to** n **do**

  for $j = 1$ **to** n **do**

  $T_{i,j}$:

$sum = 0;$
for $k = 1$ to n **do**
$sum = sum + a(i,k) * b(k,j);$
Endfor
$c(i,j) = sum;$

  Endfor

Endfor

Data access: Each task $T_{i,j}$ reads row A_i and column B_j to write data element $c_{i,j}$.

Task graph: n^2 independent tasks:

$T_{1,1}$ $T_{1,2}$ $\cdots T_{1,n}$

$T_{2,1}$ $T_{2,2}$ $\cdots T_{2,n}$

$\cdots$

$T_{n,1}$ $T_{n,2}$ $\cdots T_{n,n}$

Mapping.

- Matrix A is partitioned using row-wise block mapping
- Matrix C is partitioned using row-wise block mapping
- Matrix B is partitioned using column-wise block mapping
- Task $T_{i,j}$ is mapped to the processor of row i in matrix A.

  Cluster 1: $\boxed{T_{1,1}\ T_{1,2}\ \cdots T_{1,n}}$

  Cluster 2: $\boxed{T_{2,1}\ T_{2,2}\ \cdots T_{2,n}}$

  $\cdots$

  Cluster n: $\boxed{T_{n,1}\ T_{n,2}\ \cdots T_{n,n}}$

Parallel algorithm:

For $j = 1$ **to** n

Broadcast column B_j to all processors

Do tasks $T_{1,j}, T_{2,j}, \cdots, T_{n,j}$ in parallel.

Endfor

Evaluation:

- Each multiplication or addition counts one time unit ω.
- Each task $T_{i,j}$ costs $2n\omega$.
- Assume that each broadcast costs $(\alpha + \beta n) \log p$.

$$PT = \sum_{j=1}^{n} ((\alpha + \beta n) \log p + \frac{n}{p} 2n\omega)$$

$$= n(\alpha + \beta n) \log p + \frac{2n^3\omega}{p}.$$