MATRIX MULTIPLY ALGORITHMS

SCALAR FORMULATION

$$\sum_{i,j} a(i,j) = \sum_{i,j} \sum_{k=1}^{n} b(i,k) * c(k,j), \quad i = 1 - - - n, \quad j = 1 - - - n$$

THREE INDICES
SIX PERMUTATIONS

i j k
j i k
k i j
k j i
i k j
j k i

---

i j k AND j i k

1. Compute all intermediate products in parallel - n^3 parallelism.

2. Compute each final result with an invocation of a tree sum.

```
cobegin i : <1 : n>
     cobegin j : <1 : n>
          cobegin k : <1 : n>
                    temp (i, j, k) := b(i, k) * c(k, j)
          coend
          a(i, j) := innersum (temp (i, j, 1 : n))
                  innersum = cascade partial sum
                                       (tree algorithm)
     coend
coend
```

---

k i j AND k j i

(1) Compute all first terms of the inner products;
(2) Compute all second terms of the inner products and accumulate results with the first terms;
(3) Continue

```
cobegin i : <1 : n>
     cobegin j : <1 : n>
            temp (i, j) := 0
            for k := 1 to n do
                temp (i, j) := temp (i, j) + b(i, k) x c(k, j)
            enddo
                a(i, j) := temp(i, j)
     coend
coend
```

---

i k j AND j k i

1. Use n^2 processes.

2. Use broadcast communication.

Let matrix_of_b(i,j) = b(i,k) be a broadcast operator where each element of matrix_of_b(i,j) receives value of b(i,k).

```
cobegin i : <1 : n>
     cobegin j : <1 : n>
             temp (i, j) := 0
             for k := 1 to n do
                 matrix_of_b(i, j) := b(i, k)
                 matrix_of_c(i, j) := c(k, j)
                 temp (i, j) := temp(i, j) +
                                     matrix_of_b(i, j) x matrix_of_c(i, j)
            enddo
            a(i, j) := temp(i, j)
     coend
coend
```

MATRIX MULTIPLICATION

$$c(i, j) = \sum_{k=1}^{n} a(i, k)\, b(k, j)$$

let n = 2 then
$c(1, 1) = a(1, 1)\, b(1, 1) + a(1, 2)\, b(2, 1)$
$c(1, 2) = a(1, 1)\, b(1, 2) + a(1, 2)\, b(2, 2)$
$c(2, 1) = a(2, 1)\, b(1, 1) + a(2, 2)\, b(2, 1)$
$c(2, 2) = a(2, 1)\, b(1, 2) + a(2, 2)\, b(2, 2)$

$$\begin{pmatrix} a(1, 1) & a(1, 2) \\ a(2, 2) & a(2, 1) \end{pmatrix} \otimes \begin{pmatrix} b(1, 1) & b(2, 2) \\ b(2, 1) & b(1, 2) \end{pmatrix}$$

$$\parallel$$

$$c(1) = \begin{pmatrix} a(1, 1)\, b(1, 1) & a(1, 2)\, b(2, 2) \\ a(2, 2)\, b(2, 1) & a(2, 1)\, b(1, 2) \end{pmatrix}$$

$$\begin{pmatrix} a(1, 2) & a(1, 1) \\ a(2, 1) & a(2, 2) \end{pmatrix} \otimes \begin{pmatrix} b(2, 1) & b(1, 2) \\ b(1, 1) & b(2, 2) \end{pmatrix}$$

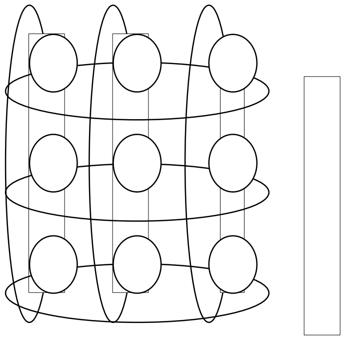
$$\parallel$$

$$c(2) = \begin{pmatrix} a(1, 2)\, b(2, 1) & a(1, 1)\, b(1, 2) \\ a(2, 1)\, b(1, 1) & a(2, 2)\, b(2, 2) \end{pmatrix}$$

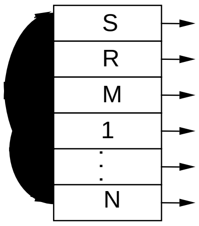
$$c = c(1) + c(2)$$

ARRAY PROCESSOR LANGUAGES

IMPLEMENT MODEL OF COMPUTATION
BASED ON ARRAY OF PROCESSORS



ROTATE AND
SHIFT
OPERATIONS



SHIFT PLANE
REGISTER PLANE
MEMORY PLANE

MEMORY PLANES CONNECTED
TO S, R, M PLANES

MEMORY FOR
EACH CELL

Rotation Algorithm

```
for k := 1 to n do
    cobegin i : <1 : n>
        cobegin j : <1 : n>
            if i > k then b(i, j) := b(i, j Rot 1)
            if j > k then c(i, j) := c(i Rot 1, j)
        coend
    coend
enddo
```

Rot is a column (row) operation
which shifts columns up (or
rows to right)

```
cobegin i : <1 : n>
     cobegin j : <1 : n>
          temp (i, j) := b(i, j) x c(i, j)
          for k := 1 to n do
          begin
              b(i, j) := b(i, j Rot 1)
              c(i, j) := c(i Rot 1, j)
              temp(i, j) := temp(i, j) + b(i, j) x c(i, j)
         enddo
         a(i, j) := temp(i, j)
     coend
coend
```

WHAT HAVE WE DONE?

PRE-SKEWED A BY ROWS	,	PRE-SKEWED B BY COLS
ROTATE JTH ROW BY J-1 POSITIONS TO LEFT		ROTATE JTH COLUMN BY J-1 POSITIONS TO TOP

THEN

1) ELEMENTWISE MULTIPLY

2) ROTATE A LEFT BY ROWS
ROTATE B TOP BY COLUMNS

3) ELEMENTWISE MULTIPLY AND SUM

4) REPEAT 2 AND 3

BLOCK MATRIX MULTIPLY

$A = (N \times N) \quad , \quad B = (N \times N)$

$N = 2k$

$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \qquad B = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}$

$A_{ij} = B_{ij} = (k \times k)$

$C = A \times B = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}$

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

GENERAL CASE

$N = pk$

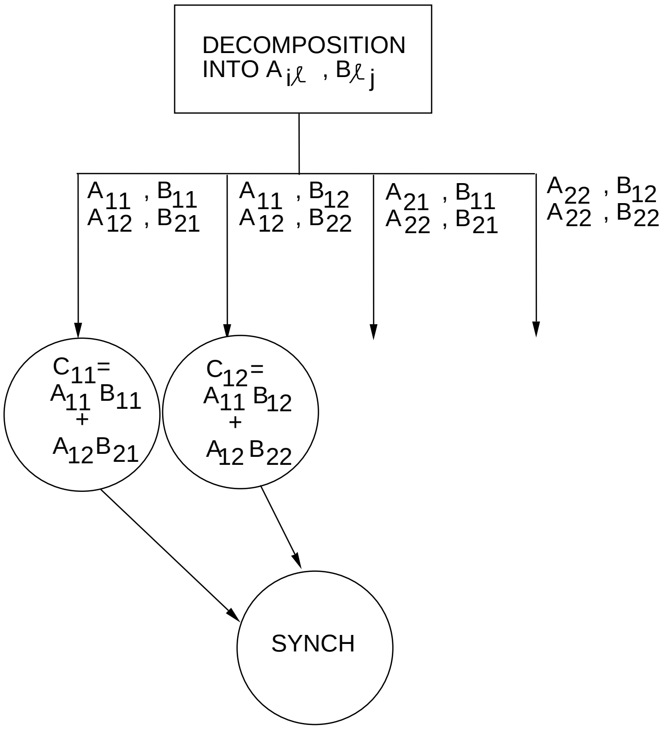
$$C_{ij} = \sum_{\ell = 1}^{p} A_{i\ell} \; B_{\ell j}$$

$A_{11} B_{11} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}$

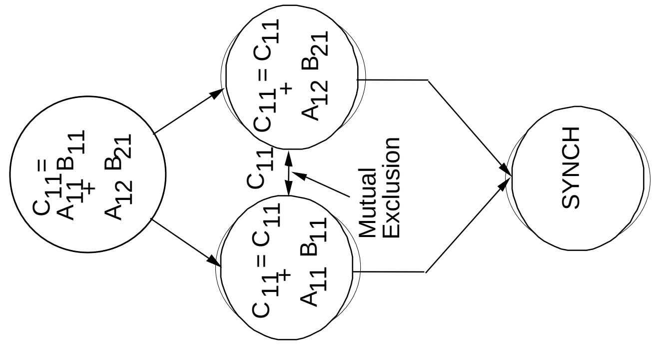
$A_{12} B_{21} = \begin{pmatrix} a_{13} & a_{14} \\ a_{23} & a_{24} \end{pmatrix} \begin{pmatrix} b_{31} & b_{32} \\ b_{41} & b_{42} \end{pmatrix}$

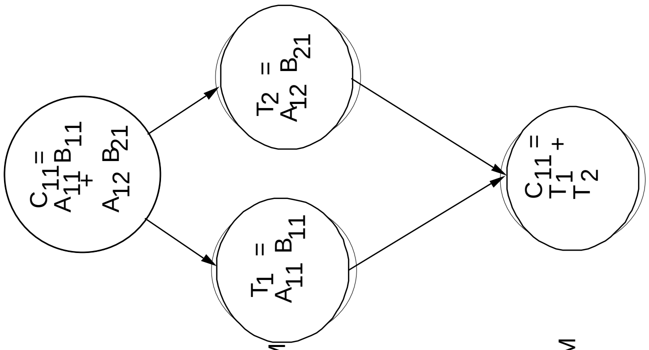
$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} = \begin{pmatrix} a_{11} b_{11} + a_{12} b_{21} & a_{11} b_{12} + a_{12} b_{22} \\ a_{21} b_{11} + a_{22} b_{21} & a_{21} b_{12} + a_{22} b_{22} \end{pmatrix}$

$\begin{pmatrix} a_{13} & a_{14} \\ a_{23} & a_{24} \end{pmatrix} \begin{pmatrix} b_{31} & b_{32} \\ b_{41} & b_{42} \end{pmatrix} = \begin{pmatrix} a_{13} b_{31} + a_{14} b_{41} & a_{13} b_{32} + a_{14} b_{42} \\ a_{23} b_{31} + a_{24} b_{41} & a_{23} b_{32} + a_{24} b_{42} \end{pmatrix}$



PARALLELISM = p = 4





PARALLELISM = 2p = 8

PARALLELISM = p = 4

```
Process row(i= 0 to n)
{       for [j = 0 to n-1]
        {c[i,j] = 0.0;
        for [k= 0 to n-1]
                c[i,j] = c[i,j] + a[i,k] * b[k,j];
        }
}
```

```
Process Worker[w = 1 to p]
{       int first = (w-1)*n/p;
        int last = first +n/p -1;
        for [i = first to last]
        {       for [j = 0 to n-1]
                c[i,j] = 0.0;
                for [k = 0 to n-1]
                {       c[i,j] = c[i,j] +a[i,k]*b[k,j];
                }
        }
}
```