Cannon's Matrix multiplication algorithms

Matrix multiplication has very specific characteristics as regards the design and implementation of a parallel algorithm in the parallel algorithms context in general:

- *Computation independency*: each element computed from the result matrix *C*, c_{ij}, is, in principle, independent of all the other elements. This independence is utterly useful because it allows a wide flexibility degree in terms of parallelization.
- *Data independence*: the number and type of operations to be carried out are independent of the data. In this case, the exception is the algorithms of the so-called sparse matrix multiplication, where there exists an attempt to take advantage of the fact that most of the matrices elements to be multiplied (and thus, of the result matrix) are equal to zero.
- *Regularity of data organization and of the operations carried out on data:* data are organized in two-dimensional structures (the same matrices), and the operations basically consist of multiplication and addition.

In computer science, Cannon's algorithm is a distributed algorithm for matrix multiplication for two-dimensional meshes first described in 1969 by Lynn Elliot Cannon. It is also proposed for a two-dimensional array of processing elements interconnected as a mesh and with the edges of each row and column interconnected, i.e. making up a structure called torus for 3×3 processing elements.

Initially, matrices *A, B,* and *C* data distribution is similar to that defined previously in the mesh, i.e. if the processors are numbered according to their position in the two-dimensional array, processor P_{ij} $(0 \le i, j \le P-1)$ has the elements or blocks of position ij $(0 \le i, j \le P-1)$ of matrices *A, B* and *C*. In order to simplify the explanation, matrices elements shall be used instead of blocks. From this data distribution, matrices *A* and *B* data are "realigned" or reassigned so that, if there is a two-dimensional array of $P \times P$ processors, the element or submatrix *A* in row *i* and column $(j+i)$ *mod* P, $a_{i,(j+i)modP}$, and also the element or submatrix of *B* in row $(i+j)$ *mod P* and column *j*, $b_{(i+j)modP,i}$, are assigned to processor P_{ij}. In other words, each data of row i $(0 \le i \le P-1)$ of the elements or submatrices of *A* are transferred or shifted *i* times towards the left processors, and each data or column j $(0 \le j \le P-1)$ of the elements or submatrices of *B* are transferred or shifted j times towards upper processors.

From the initial relocation, the following steps are carried out iteratively:

- Local multiplication of data assigned in each processor for a partial result computation;

- Left rotation of the elements or submatrices of *A*;
- Upwards rotation of the elements or submatrices of *B*, and after *P* of these steps, thoroughly computed values of matrix *C* are finally obtained.

Summarizing, the outstanding characteristics of this way to carry out matrix multiplication are:

- By the way matrices *A* and *B* data are communicated, this is an "initial alignment and rotating" algorithm.
- Load balance, both in terms of computation and communications, is assured only if the processing algorithms are homogeneous.
- As in the processing defined for the processors grid, the running time is minimized if the computation can be overlapped in time with communications.
- Matrices *A* and *B* data distribution is not the initial one when the matrix multiplication computation is finished.

Finally, the **Cannon's Algorithm** consists of the following steps:

1) The **initial step** of the algorithm regards the alignment of the matrixes:
    a. Align the blocks of A and B in such a way that each process can independently start multiplying its local submatrices. This is done by shifting all submatrices $A_{i,j}$ to the left (with wraparound) by **i** steps a and all submatrices $B_{i,j}$ up (with wraparound) by **j** steps.
2) Perform local block multiplication.
3) Each block of *A* moves **one step left** and each block of *B* moves **one step up** (again with wraparound).
4) Perform next block multiplication, add to partial result, repeat until all blocks have been multiplied.

Cannon's Algorithm for 3×3 Matrices and process (1,2) to determine C_{12}

Initial A,B

A(0,0)	A(0,1)	A(0,2)
A(1,0)	A(1,1)	A(1,2)
A(2,0)	A(2,1)	A(2,2)

B(0,0)	B(0,1)	B(0,2)
B(1,0)	B(1,1)	B(1,2)
B(2,0)	B(2,1)	B(2,2)

A,B initial alignment

A(0,0)	A(0,1)	A(0,2)
A(1,1)	A(1,2)	A(1,0)
A(2,0)	A(2,1)	A(2,2)

B(0,0)	B(0,1)	B(2,2)
B(1,0)	B(1,1)	B(0,2)
B(2,0)	B(2,1)	B(1,2)

A,B after shift step 1

A(0,0)	A(0,1)	A(0,2)
A(1,2)	A(1,0)	A(1,1)
A(2,0)	A(2,1)	A(2,2)

B(0,0)	B(0,1)	B(0,2)
B(1,0)	B(1,1)	B(1,2)
B(2,0)	B(2,1)	B(2,2)

A,B after shift step 2

A(0,0)	A(0,1)	A(0,2)
A(1,0)	A(1,1)	A(1,2)
A(2,0)	A(2,1)	A(2,2)

B(0,0)	B(0,1)	B(1,2)
B(1,0)	B(1,1)	B(2,2)
B(2,0)	B(2,1)	B(0,2)

So C(1,2)=A(1,0)*B(0,2)+A(1,1)*B(1,2)+ A(1,2)*B(2,2) and according the Cannon's Algorithm we have:

- $C^1(1,2)= A(1,0)*B(0,2)$ (after A, B initial alignment)
- $C^2(1,2)= C^1(1,2)+A(1,1)*B(1,2)$ (A, B after shift step 1)

- $C(1,2)=C^3(1,2)= C^2(1,2)+A(1,2)*B(2,2)$ (A, B after shift step 2)

Attention! Limitations of Cannon's Algorithm

- P (number of processors) must be a perfect square;
- Matrices A and B must be square, and evenly divisible by √p

Additional explanations when matrices are distributed according to the two-dimensional block-cyclic data layout scheme.

Let **A**=$\begin{pmatrix}1 & 4 & 7 & 10 & 13\\ 3 & 6 & 9 & 12 & 15\\ 5 & 8 & 11 & 14 & 17\\ 7 & 10 & 13 & 16 & 19\\ 9 & 12 & 15 & 18 & 21\end{pmatrix}$ and **X**=$\begin{pmatrix}1\\1\\0\\0\\1\end{pmatrix}$. It is evident that **AX**=$\begin{pmatrix}18\\24\\30\\36\\42\end{pmatrix}$

If the matrix **A** is distributed to the grid with **2x2** processes with block dimensions **2x2** using *2-D cyclic* algorithm, then the processes will possess the following submatrices:

$\mathbf{A_{(0,0)}}=\begin{pmatrix}1 & 4 & 13\\ 3 & 6 & 15\\ 9 & 12 & 21\end{pmatrix}$, $\mathbf{X_{(0,0)}}=\begin{pmatrix}1\\1\\1\end{pmatrix}$; $\mathbf{A_{(0,1)}}=\begin{pmatrix}7 & 10\\ 9 & 12\\ 15 & 18\end{pmatrix}$, $\mathbf{X_{(0,1)}}=\begin{pmatrix}1\\1\\1\end{pmatrix}$;

$\mathbf{A_{(1,0)}}=\begin{pmatrix}5 & 8 & 17\\ 7 & 10 & 19\end{pmatrix}$, $\mathbf{X_{(1,0)}}=\begin{pmatrix}0\\0\end{pmatrix}$; $\mathbf{A_{(1,1)}}=\begin{pmatrix}11 & 14\\ 13 & 16\end{pmatrix}$, $\mathbf{X_{(1,1)}}=\begin{pmatrix}0\\0\end{pmatrix}$;

So we obtain:

	0	**1**
0	$A_{(0,0)}$	$A_{(0,1)}$
1	$A_{(1,0)}$	$A_{(1,1)}$

	0
0	$X_{(0,0)}$
1	$X_{(1,0)}$

The process will be carried out the following calculations:

- Process(0,0): $\boldsymbol{Y_{(0,0)}= A_{(0,0)} X_{(0,0)}+ A_{(0,1)} X_{(1,0)}}$
- Process(1,0): $\boldsymbol{Y_{(1,0)}= A_{(1,0)} X_{(0,0)}+ A_{(1,1)} X_{(1,0)}}$

To determine the product **AX** we use the Cannon algorithm**.** Therefore:

1. It is carried out the alignment of submatrices, i.e. the submatrices $\boldsymbol{A_{(i,j)}}$ (that are in possession of the process *(i,j))* are "moving" (are transmitted on process grid) to the left by *i* times and the submatrices $\boldsymbol{X_{(i,j)}}$ are "moving"" (are transmitted on process grid) to the top by *j* times.

   As a result for each process we get the following:

   - The process (0,0) does nothing (because it can perform the operation of matrix- vector multiplication) and thus it is realized the product $\boldsymbol{Y^0_{(0,0)} =A_{(0,0)}X_{(0,0)}}=\begin{pmatrix}1 & 4 & 13\\ 3 & 6 & 15\\ 9 & 12 & 21\end{pmatrix}\begin{pmatrix}1\\1\\1\end{pmatrix}=\begin{pmatrix}18\\24\\42\end{pmatrix}$
   - The process (1,0). The submatrix $\boldsymbol{A_{(1,0)}}$ is transmitted by *i*=1 to the left, but $\boldsymbol{X_{(1,0)}}$- is not transmitted (*j=0*), so we have:

	0	1
0	$A_{(0,0)}$	$A_{(0,1)}$
1	$A_{(1,1)}$	$A_{(1,0)}$

	0
0	$X_{(0,0)}$
1	$X_{(1,0)}$

So the process (1,0) can calculate $\boldsymbol{Y^0_{(1,0)}} = \boldsymbol{A_{(1,1)}}\ \boldsymbol{X_{(1,0)}} = \begin{pmatrix}11 & 14\\ 13 & 16\end{pmatrix}\begin{pmatrix}0\\ 0\end{pmatrix} = \begin{pmatrix}0\\ 0\end{pmatrix}$.

2. Each submatrices of $\boldsymbol{A_{(i,j)}}$ (that are in possession of the process *(i,j) after the iteration 1.)* moves **one step left** and each submatrices $\boldsymbol{X_{(i,j)}}$ moves **one step up** (again with wraparound).

   As a result for each process we get the following:

   - The process (0,0):

	0	1
0	$A_{(0,1)}$	$A_{(0,0)}$
1	$A_{(1,0)}$	$A_{(1,1)}$

	0
0	$X_{(1,0)}$
1	$X_{(0,0)}$

The process will be carried out the following calculations $\boldsymbol{Y^1_{(0,0)}} = \boldsymbol{Y^0_{(0,0)}} + \boldsymbol{A_{(0,1)}}\ \boldsymbol{X_{(1,0)}} = \begin{pmatrix}18\\ 24\\ 42\end{pmatrix} + \begin{pmatrix}7 & 10\\ 9 & 12\\ 15 & 18\end{pmatrix}\begin{pmatrix}0\\ 0\end{pmatrix} = \begin{pmatrix}18\\ 24\\ 42\end{pmatrix} + \begin{pmatrix}0\\ 0\\ 0\end{pmatrix}$

- The process (1,0):

	0	1
0	$A_{(0,0)}$	$A_{(0,1)}$
1	$A_{(1,0)}$	$A_{(1,1)}$

	0
0	$X_{(1,0)}$
1	$X_{(0,0)}$

The process will be carried out the following calculations $\boldsymbol{Y^1_{(1,0)}} = \boldsymbol{Y^0_{(1,0)}} + \boldsymbol{A_{(1,0)}}\ \boldsymbol{X_{(0,0)}} = \begin{pmatrix}0\\ 0\end{pmatrix} + \begin{pmatrix}5 & 8 & 17\\ 7 & 10 & 19\end{pmatrix}\begin{pmatrix}1\\ 1\\ 1\end{pmatrix} = \begin{pmatrix}30\\ 36\end{pmatrix}$

Therefore, finally we obtain

$$Y=AX=\begin{pmatrix}18\\ 24\\ 30\\ 36\\ 42\end{pmatrix}\begin{matrix}\to process(0,0)\\ \to process(0,0)\\ \to process(1,0)\\ \to process(1,0)\\ \to process(0,0)\end{matrix}$$

Scalable Universal Matrix Multiply Algorithm (SUMMA),

An alternative for Cannon algorithm represents the SUMMA algorithm:

- SUMMA = Scalable Universal Matrix Multiply Algorithm.
- Slightly less efficient, but simpler and easier to generalize.

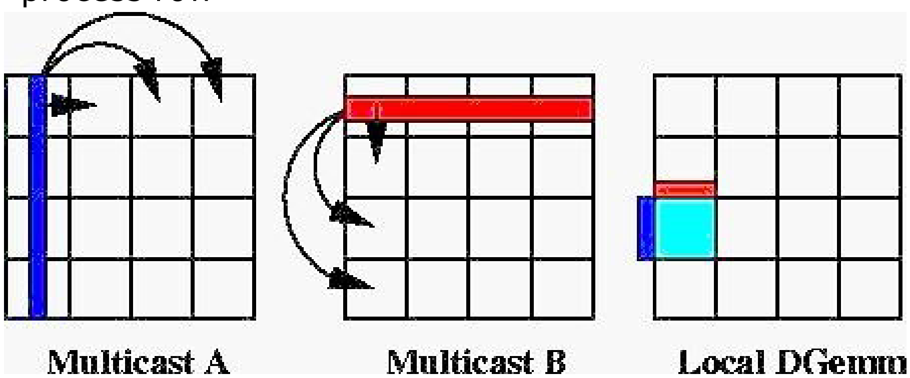
Naive matrix multiply

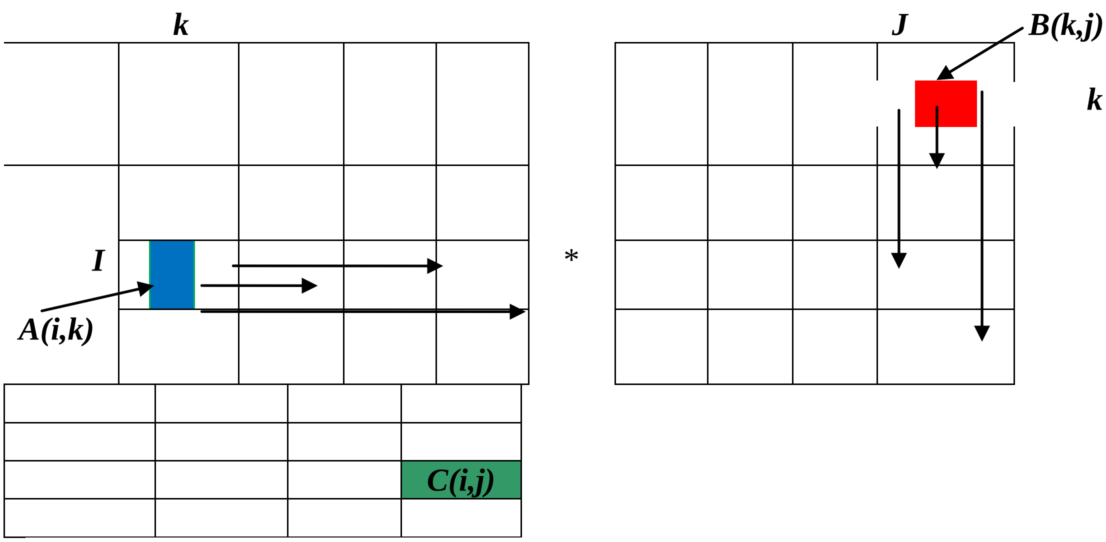
  - *For i = 0 to n*
  - *For j = 0 to n*
  - *For k = 0 to n*
  - *C[i,j] += A[i,k]*B[k,j]*

Calculates n^2 dot products (inner products) *C[i,j] = A[i,:]*B[:,j]*

The main steps of the algorithm:

- For each *k* (between 0 and n-1),
  - Owner of partial row *k* broadcasts that row along its process column;
  - Owner of partial column *k* broadcasts that column along its process row





- ***I, J*** represent all rows, columns owned by a processor
- ***k*** is a single row or column (or a block of ***b*** rows or columns)
- ***C(I,J) = C(I,J) +*** $\boldsymbol{\Sigma_k}$ ***A(I,k)*B(k,J)***

Assume a p_r by p_c processor grid ($p_r = p_c = 4$ above)
Complete pseudo code of the algorithm.

// On each process P(i,j):

- **For** *k=0* to *n-1* … or *n/b-1* where b is the block size
  …= # cols in A(I,k) and # rows in B(k,J)
  - for all *I = 1* to p_r … in parallel
    - owner of *A(I,k)* broadcasts it to whole processor row
  - for all J = 1 to p_c … in parallel
    - owner of *B(k,J)* broadcasts it to whole processor column
  - Receive A(I,k) into Acol
  - Receive B(k,J) into Brow
  - C(myproc,myproc) = C(myproc,myproc)+Acol*Brow
- **Endfor**

We consider the following two matrices

A_{11}	A_{12}	A_{13}	A_{14}	A_{15}	A_{16}	A_{17}	A_{18}	A_{19}
A_{21}	A_{22}	A_{23}	A_{24}	A_{25}	A_{26}	A_{27}	A_{28}	A_{29}
A_{31}	A_{32}	A_{33}	A_{34}	A_{35}	A_{36}	A_{37}	A_{38}	A_{39}
A_{41}	A_{42}	A_{43}	A_{44}	A_{45}	A_{46}	A_{47}	A_{48}	A_{49}
A_{51}	A_{52}	A_{53}	A_{54}	A_{55}	A_{56}	A_{57}	A_{58}	A_{59}
A_{61}	A_{62}	A_{63}	A_{64}	A_{65}	A_{66}	A_{67}	A_{68}	A_{69}
A_{71}	A_{72}	A_{73}	A_{74}	A_{75}	A_{76}	A_{77}	A_{78}	A_{79}
A_{81}	A_{82}	A_{83}	A_{84}	A_{85}	A_{86}	A_{87}	A_{88}	A_{89}
A_{91}	A_{92}	A_{93}	A_{94}	A_{95}	A_{96}	A_{97}	A_{98}	A_{99}

B_{11}	B_{12}	B_{13}	B_{14}	B_{15}	B_{16}	B_{17}	B_{18}	B_{19}
B_{21}	B_{22}	B_{23}	B_{24}	B_{25}	B_{26}	B_{27}	B_{28}	B_{29}
B_{31}	B_{32}	B_{33}	B_{34}	B_{35}	B_{36}	B_{37}	B_{38}	B_{39}
B_{41}	B_{42}	B_{43}	B_{44}	B_{45}	B_{46}	B_{47}	B_{48}	B_{49}
B_{51}	B_{52}	B_{53}	B_{54}	B_{55}	B_{56}	B_{57}	B_{58}	B_{59}
B_{61}	B_{62}	B_{63}	B_{64}	B_{65}	B_{66}	B_{67}	B_{68}	B_{69}
B_{71}	B_{72}	B_{73}	B_{74}	B_{75}	B_{76}	B_{77}	B_{78}	B_{79}
B_{81}	B_{82}	B_{83}	B_{84}	B_{85}	B_{86}	B_{87}	B_{88}	B_{89}
B_{91}	B_{92}	B_{93}	B_{94}	B_{95}	B_{96}	B_{97}	B_{98}	B_{99}

As a result of distribution of these matrices using “2Dciclic” algorithm on the grid of 2x3 processes (2 lines and 3 columns) with block dimensions 2x2, we obtain:

Matrix ***A***	Matrix ***B***
$\mathbf{A}_{(0,0)}=\begin{pmatrix} A_{11} & A_{12} & A_{17} & A_{18} \\ A_{21} & A_{22} & A_{27} & A_{28} \\ A_{51} & A_{52} & A_{57} & A_{58} \\ A_{61} & A_{62} & A_{67} & A_{68} \\ A_{91} & A_{92} & A_{97} & A_{98} \end{pmatrix}$	$\mathbf{B}_{(0,0)}=\begin{pmatrix} B_{11} & B_{12} & B_{17} & B_{18} \\ B_{21} & B_{22} & B_{27} & B_{28} \\ B_{51} & B_{52} & B_{57} & B_{58} \\ B_{61} & B_{62} & B_{67} & B_{68} \\ B_{91} & B_{92} & B_{97} & B_{98} \end{pmatrix}$
$\mathbf{A}_{(0,1)}=\begin{pmatrix} A_{13} & A_{14} & A_{19} \\ A_{23} & A_{24} & A_{29} \\ A_{53} & A_{54} & A_{59} \\ A_{63} & A_{64} & A_{69} \\ A_{93} & A_{94} & A_{99} \end{pmatrix}$	$\mathbf{B}_{(0,1)}=\begin{pmatrix} B_{13} & B_{14} & B_{19} \\ B_{23} & B_{24} & B_{29} \\ B_{53} & B_{54} & B_{59} \\ B_{63} & B_{64} & B_{69} \\ B_{93} & B_{94} & B_{99} \end{pmatrix}$
$\mathbf{A}_{(0,2)}=\begin{pmatrix} A_{15} & A_{16} \\ A_{25} & A_{26} \\ A_{55} & A_{56} \\ A_{65} & A_{66} \\ A_{95} & A_{96} \end{pmatrix}$	$\mathbf{B}_{(0,2)}=\begin{pmatrix} B_{15} & B_{16} \\ B_{25} & B_{26} \\ B_{55} & B_{56} \\ B_{65} & B_{66} \\ B_{95} & B_{96} \end{pmatrix}$
$\mathbf{A}_{(1,0)}=\begin{pmatrix} A_{31} & A_{32} & A_{37} & A_{38} \\ A_{41} & A_{42} & A_{47} & A_{48} \\ A_{71} & A_{72} & A_{77} & A_{78} \\ A_{81} & A_{82} & A_{87} & A_{88} \end{pmatrix}$	$\mathbf{B}_{(1,0)}=\begin{pmatrix} B_{31} & B_{32} & B_{37} & B_{38} \\ B_{41} & B_{42} & B_{47} & B_{48} \\ B_{71} & B_{72} & B_{77} & B_{78} \\ B_{81} & B_{82} & B_{87} & B_{88} \end{pmatrix}$
$\mathbf{A}_{(1,1)}=\begin{pmatrix} A_{33} & A_{34} & A_{39} \\ A_{43} & A_{44} & A_{49} \\ A_{73} & A_{74} & A_{79} \\ A_{83} & A_{84} & A_{89} \end{pmatrix}$	$\mathbf{B}_{(1,1)}=\begin{pmatrix} B_{33} & B_{34} & B_{39} \\ B_{43} & B_{44} & B_{49} \\ B_{73} & B_{74} & B_{79} \\ B_{83} & B_{84} & B_{89} \end{pmatrix}$

$\mathbf{A_{(1,2)}}=\begin{pmatrix} A_{35} & A_{36} \\ A_{45} & A_{46} \\ A_{75} & A_{76} \\ A_{85} & A_{86} \end{pmatrix}$	$\mathbf{B_{(1,2)}}=\begin{pmatrix} B_{35} & B_{36} \\ B_{45} & B_{46} \\ B_{75} & B_{76} \\ B_{85} & B_{86} \end{pmatrix}$

So in a compact form we have

	0	**1**	**2**
0	$\boldsymbol{A_{(0,0)}}$	$\boldsymbol{A_{(0,1)}}$	$\boldsymbol{A_{(0,2)}}$
1	$\boldsymbol{A_{(1,0)}}$	$\boldsymbol{A_{(1,1)}}$	$\boldsymbol{A_{(1,2)}}$

	0	**1**	**2**
0	$\boldsymbol{B_{(0,0)}}$	$\boldsymbol{B_{(0,1)}}$	$\boldsymbol{B_{(0,2)}}$
1	$\boldsymbol{B_{(1,0)}}$	$\boldsymbol{B_{(1,1)}}$	$\boldsymbol{B_{(1,2)}}$

To determine the product **C=AB** we use the algorithm SUMMA**.** Processes will determine the following submatrices of the matrix **C:**

$$\mathbf{C_{(0,0)}}=\begin{pmatrix} C_{11} & C_{12} & C_{17} & C_{18} \\ C_{21} & C_{22} & C_{27} & C_{28} \\ C_{51} & C_{52} & C_{57} & C_{68} \\ C_{61} & C_{62} & C_{67} & C_{68} \\ C_{91} & C_{92} & C_{97} & C_{98} \end{pmatrix} \quad \mathbf{C_{(0,1)}}=\begin{pmatrix} C_{13} & C_{14} & C_{19} \\ C_{23} & C_{24} & C_{29} \\ C_{53} & C_{54} & C_{59} \\ C_{63} & C_{64} & C_{69} \\ C_{93} & C_{94} & C_{99} \end{pmatrix} \quad \mathbf{C_{(0,2)}}=\begin{pmatrix} C_{15} & C_{16} \\ C_{25} & C_{26} \\ C_{55} & C_{56} \\ C_{65} & C_{66} \\ C_{95} & C_{96} \end{pmatrix}$$

$$\mathbf{C_{(1,0)}}=\begin{pmatrix} C_{31} & C_{32} & C_{37} & C_{38} \\ C_{41} & C_{42} & C_{47} & C_{48} \\ C_{71} & C_{72} & C_{77} & C_{78} \\ C_{81} & C_{82} & C_{87} & C_{88} \end{pmatrix} \quad \mathbf{C_{(1,1)}}=\begin{pmatrix} C_{33} & C_{34} & C_{39} \\ C_{43} & C_{44} & C_{49} \\ C_{73} & C_{74} & C_{79} \\ C_{83} & C_{84} & C_{89} \end{pmatrix} \quad \mathbf{C_{(1,2)}}=\begin{pmatrix} C_{35} & C_{36} \\ C_{45} & C_{46} \\ C_{75} & C_{76} \\ C_{85} & C_{86} \end{pmatrix}$$

Here for example, the elements of the submatrix $\mathbf{C_{(0,0)}}$ are:

$C_{11}=A_{11}B_{11} + A_{12}B_{21}+ A_{13}B_{31} +A_{14}B_{41} +A_{15}B_{51}+ A_{16}B_{61}+ A_{17}B_{71} +A_{18}B_{81} +A_{19}B_{91}$

$C_{12}=A_{11}B_{12} + A_{12}B_{22}+ A_{13}B_{32} +A_{14}B_{42} +A_{15}B_{52}+ A_{16}B_{62}+ A_{17}B_{72} +A_{18}B_{82} +A_{19}B_{92}$

$C_{17}=A_{11}B_{17} + A_{12}B_{27}+ A_{13}B_{37} +A_{14}B_{47} +A_{15}B_{57}+ A_{16}B_{67}+ A_{17}B_{77} +A_{18}B_{87} +A_{19}B_{97}$

$C_{18}=A_{11}B_{18} + A_{12}B_{28}+ A_{13}B_{38} +A_{14}B_{48} +A_{15}B_{58}+ A_{16}B_{68}+ A_{17}B_{78} +A_{18}B_{88} +A_{19}B_{98}$

$C_{21}=A_{21}B_{11} + A_{22}B_{21}+ A_{23}B_{31} +A_{24}B_{41} +A_{25}B_{51}+ A_{26}B_{61}+ A_{27}B_{71} +A_{28}B_{81} +A_{29}B_{91}$

$C_{22}=A_{21}B_{12} + A_{22}B_{22}+ A_{23}B_{32} +A_{24}B_{42} +A_{25}B_{52}+ A_{26}B_{62}+ A_{27}B_{72} +A_{28}B_{82} +A_{29}B_{92}$

$C_{27}=A_{21}B_{17} + A_{22}B_{27}+ A_{23}B_{37} +A_{24}B_{47} +A_{25}B_{57}+ A_{26}B_{67}+ A_{27}B_{77} +A_{28}B_{87} +A_{29}B_{97}$

$C_{28}=A_{21}B_{18}+ A_{22}B_{28}+ A_{23}B_{38}+A_{24}B_{48} +A_{25}B_{58}+ A_{26}B_{68}+ A_{27}B_{77} +A_{28}B_{88} +A_{29}B_{98}$

$C_{51}=A_{51}B_{11} + A_{52}B_{21}+ A_{53}B_{31} +A_{54}B_{41} +A_{55}B_{51}+ A_{56}B_{61}+ A_{57}B_{71} +A_{58}B_{81} +A_{59}B_{91}$

$C_{52}=A_{51}B_{12} + A_{52}B_{22}+ A_{53}B_{32} +A_{54}B_{42} +A_{55}B_{52}+ A_{56}B_{62}+ A_{57}B_{72} +A_{58}B_{82} +A_{59}B_{92}$
$C_{57}=A_{51}B_{17} + A_{52}B_{27}+ A_{53}B_{37} +A_{54}B_{47} +A_{55}B_{57}+ A_{56}B_{67}+ A_{57}B_{77} +A_{58}B_{87} +A_{59}B_{97}$
$C_{58}=A_{51}B_{18}+ A_{52}B_{28}+ A_{53}B_{38}+A_{54}B_{48} +A_{55}B_{58}+ A_{56}B_{68}+ A_{57}B_{78} +A_{58}B_{88} +A_{59}B_{98}$

$C_{61}=A_{61}B_{11} + A_{62}B_{21}+ A_{63}B_{31} +A_{64}B_{41} +A_{65}B_{51}+ A_{66}B_{61}+ A_{67}B_{71} +A_{68}B_{81} +A_{69}B_{91}$
$C_{62}=A_{61}B_{12} + A_{62}B_{22}+ A_{63}B_{32} +A_{64}B_{42} +A_{65}B_{52}+ A_{66}B_{62}+ A_{67}B_{72} +A_{68}B_{82} +A_{69}B_{92}$
$C_{67}=A_{61}B_{17} + A_{62}B_{27}+ A_{63}B_{37} +A_{64}B_{47} +A_{65}B_{57}+ A_{66}B_{67}+ A_{67}B_{77} +A_{68}B_{87} +A_{69}B_{97}$
$C_{68}=A_{61}B_{18}+ A_{62}B_{28}+ A_{63}B_{38}+A_{64}B_{48} +A_{65}B_{58}+ A_{66}B_{68}+ A_{67}B_{77} +A_{68}B_{88} +A_{69}B_{98}$

$C_{91}=A_{91}B_{11} + A_{92}B_{21}+ A_{93}B_{31} +A_{94}B_{41} +A_{95}B_{51}+ A_{96}B_{61}+ A_{97}B_{71} +A_{98}B_{81} +A_{99}B_{91}$
$C_{92}=A_{91}B_{12} + A_{92}B_{22}+ A_{93}B_{32} +A_{94}B_{42} +A_{95}B_{52}+ A_{96}B_{62}+ A_{97}B_{72} +A_{98}B_{82} +A_{99}B_{92}$
$C_{97}=A_{91}B_{17} + A_{92}B_{27}+ A_{93}B_{37} +A_{94}B_{47} +A_{95}B_{57}+ A_{96}B_{67}+ A_{97}B_{77} +A_{98}B_{87} +A_{99}B_{97}$
$C_{98}=A_{91}B_{18}+ A_{92}B_{28}+ A_{93}B_{38}+A_{94}B_{48} +A_{95}B_{58}+ A_{96}B_{68}+ A_{97}B_{78} +A_{98}B_{88} +A_{99}B_{98}$

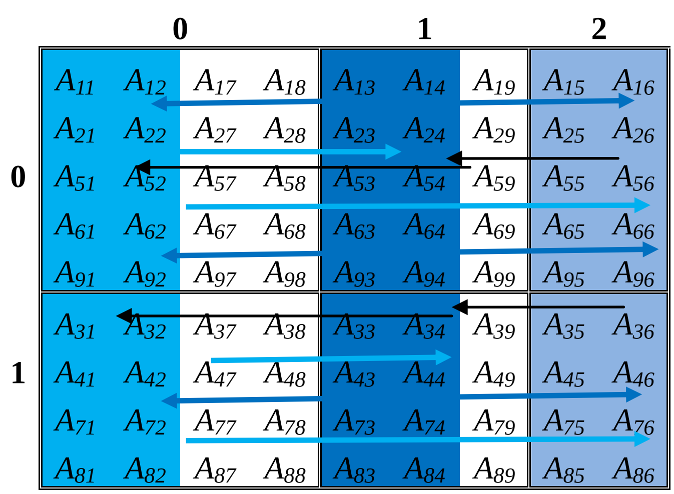
We apply the algorithm SUMMA.
The main iterations are:

- ***I, J*** represent all rows, columns owned by a processor;
- ***k*** is a single row or column (or a block of ***b*** rows or columns);
- ***C(i,j) = C(i,j) +*** $\sum_k A(i,k)\times B(k,j)$ **.**

Assume a p_r by p_c processor grid (p_r = p_c = *4* above). Need not be a square.
Complete algorithm. On each process *P(i,j):*

- For k=0 to n-1 … or n/b-1 where b is the block size
  … = # cols in A(I,k) and # rows in B(k,J)
- for all *I* = *1* to p_r … in parallel
  ✓owner of *A(I,k)* broadcasts it to whole processor row
- for all J = 1 to p_c … in parallel
  ✓owner of B(k,J) broadcasts it to whole processor column
- Receive A(I,k) into Acol
- Receive B(k,J) into Brow
- C(myproc , myproc) = C(myproc , myproc) + Acol * Brow

The initial situation is:



	0				1			2	
	B_{11}	B_{12}	B_{17}	B_{18}	B_{13}	B_{14}	B_{19}	B_{15}	B_{16}
	B_{21}	B_{22}	B_{27}	B_{28}	B_{23}	B_{24}	B_{29}	B_{25}	B_{26}
0	B_{51}	B_{52}	B_{57}	B_{58}	B_{53}	B_{54}	B_{59}	B_{55}	B_{56}
	B_{61}	B_{62}	B_{67}	B_{68}	B_{63}	B_{64}	B_{69}	B_{65}	B_{66}
	B_{91}	B_{92}	B_{97}	B_{98}	B_{93}	B_{94}	B_{99}	B_{95}	B_{96}
	B_{31}	B_{32}	B_{37}	B_{38}	B_{33}	B_{34}	B_{39}	B_{35}	B_{36}
1	B_{41}	B_{42}	B_{47}	B_{48}	B_{43}	B_{44}	B_{49}	B_{45}	B_{46}
	B_{71}	B_{72}	B_{77}	B_{78}	B_{73}	B_{74}	B_{79}	B_{75}	B_{76}
	B_{81}	B_{82}	B_{87}	B_{88}	B_{83}	B_{84}	B_{89}	B_{85}	B_{86}

The process (0,0)

On the base of algorithm SUMMA we obtain:

Let k=2 (the block length, i.e. there are transmitted two rows and two columns).

- **Iteration 0**. There is no data transmission

0

Acol *←(0,0)*

0	A_{11}	A_{12}
	A_{21}	A_{22}
	A_{51}	A_{52}
	A_{61}	A_{62}
	A_{91}	A_{92}

0

Brow

0 ↑ *(0,0)*

B_{11}	B_{12}	B_{17}	B_{18}
B_{21}	B_{22}	B_{27}	B_{28}

Then it is possible to calculate the product $C_{(0,0)}= C_{(0,0)}+Acol*Brow$. So we obtain

$C^0_{(0,0)}=$	$C^0_{1,1}=A_{11}B_{11}+A_{12}B_{21}$	$C^0_{1,2}=A_{11}B_{12}+A_{12}B_{22}$	$C^0_{1,7}=A_{11}B_{17}+A_{12}B_{27}$	$C^0_{1,8}=A_{11}B_{18}+A_{12}B_{28}$
	$C^0_{2,1}=A_{21}B_{11}+A_{22}B_{21}$	$C^0_{22}=A_{21}B_{12}+A_{22}B_{22}$	$C^0_{2,7}=A_{21}B_{17}+A_{22}B_{27}$	$C^0_{2,8}=A_{21}B_{18}+A_{22}B_{28}$
	$C^0_{5,1}=A_{51}B_{11}+A_{52}B_{21}$	$C^0_{52}=A_{51}B_{12}+A_{52}B_{22}$	$C^0_{57}=A_{51}B_{17}+A_{52}B_{27}$	$C^0_{58}=A_{51}B_{18}+A_{52}B_{28}$
	$C^0_{61}=A_{61}B_{11}+A_{62}B_{21}$	$C^0_{62}=A_{61}B_{12}+A_{62}B_{22}$	$C^0_{67}=A_{61}B_{17}+A_{62}B_{27}$	$C^0_{68}=A_{61}B_{18}+A_{62}B_{28}$
	$C^0_{91}=A_{91}B_{11}+A_{92}B_{21}$	$C^0_{92}=A_{91}B_{12}+A_{92}B_{22}$	$C^0_{97}=A_{91}B_{17}+A_{92}B_{27}$	$C^0_{98}=A_{91}B_{18}+A_{92}B_{28}$

Finally, $C_{(0,0)}= C^0_{(0,0)}$

Therefore, at the end of this iteration the process (0,0) already has calculated

$C_{11}=A_{11}B_{11} + A_{12}B_{21}$
$C_{12}=A_{11}B_{12} + A_{12}B_{22}$
$C_{17}=A_{11}B_{17} + A_{12}B_{27}$
$C_{18}=A_{11}B_{18} + A_{12}B_{28}$

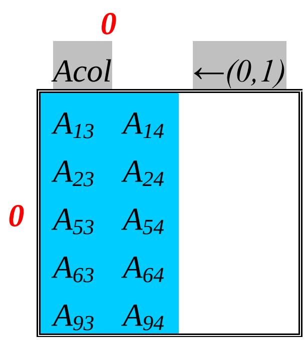
$C_{21}=A_{21}B_{11} + A_{22}B_{21}$
$C_{22}=A_{21}B_{12} + A_{22}B_{22}$
$C_{27}=A_{21}B_{17} + A_{22}B_{27}$
$C_{28}=A_{21}B_{18}+ A_{22}B_{28}$

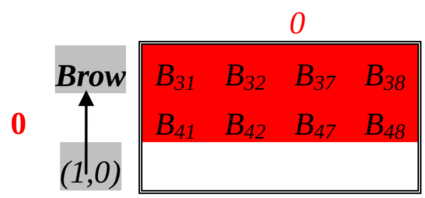
$C_{51}=A_{51}B_{11} + A_{52}B_{21}$
$C_{52}=A_{51}B_{12} + A_{52}B_{22}$
$C_{57}=A_{51}B_{17} + A_{52}B_{27}$
$C_{58}=A_{51}B_{18}+ A_{52}B_{28}$

$C_{61}=A_{61}B_{11} + A_{62}B_{21}$
$C_{62}=A_{61}B_{12} + A_{62}B_{22}$
$C_{67}=A_{61}B_{17} + A_{62}B_{27}$
$C_{68}=A_{61}B_{18}+ A_{62}B_{28}$

$C_{91}=A_{91}B_{11} + A_{92}B_{21}$
$C_{92}=A_{91}B_{12} + A_{92}B_{22}$
$C_{97}=A_{91}B_{17} + A_{92}B_{27}$
$C_{98}=A_{91}B_{18}+ A_{92}B_{28}$

- **Iteration 1.** Here we have for *Acol* and *Bcol* the following:





Then it is possible to calculate the product $C_{(0,0)}= C_{(0,0)}$+Acol*Brow. So we obtain

$C^1_{(0,0)}=$	$\boldsymbol{C^1_{1,1}}=A_{13}B_{31}+A_{14}B_{41}$ $\boldsymbol{C^1_{2,1}}=A_{23}B_{31}+A_{24}B_{41}$ $\boldsymbol{C^1_{5,1=}}A_{53}B_{31}+A_{54}B_{41}$ $\boldsymbol{C^1_{61}}=A_{63}B_{31}+A_{64}B_{41}$ $\boldsymbol{C^1_{91}}=A_{93}B_{31}+A_{94}B_{41}$	$\boldsymbol{C^1_{1,2}}=$ $A_{13}B_{32}+A_{14}B_{42}$ $\boldsymbol{C^1_{22}}=$ $A_{23}B_{32}+A_{24}B_{42}$ $\boldsymbol{C^1_{52}}=$ $A_{53}B_{32}+A_{54}B_{42}$ $\boldsymbol{C^1_{62}}=$ $A_{63}B_{32}+A_{64}B_{42}$ $\boldsymbol{C^1_{92}}=$ $A_{93}B_{32}+A_{94}B_{42}$	$\boldsymbol{C^1_{1,7}}=$ $A_{13}B_{37}+A_{14}B_{47}$ $\boldsymbol{C^1_{2,7}}=$ $A_{23}B_{37}+A_{24}B_{47}$ $\boldsymbol{C^1_{57}}=$ $A_{53}B_{37}+A_{54}B_{47}$ $\boldsymbol{C^1_{67}}=$ $A_{63}B_{37}+A_{64}B_{47}$ $\boldsymbol{C^1_{97}}=$ $A_{93}B_{37}+A_{94}B_{47}$	$\boldsymbol{C^1_{1,8}}=$ $A_{13}B_{38}+A_{14}B_{48}$ $\boldsymbol{C^1_{2,8}}=$ $A_{23}B_{38}+A_{24}B_{48}$ $\boldsymbol{C^1_{58}}=$ $A_{53}B_{38}+A_{54}B_{48}$ $\boldsymbol{C^1_{68}}=$ $A_{63}B_{38}+A_{64}B_{48}$ $\boldsymbol{C^1_{98}}=$ $A_{93}B_{38}+A_{94}B_{48}$

Therefore, at the end of this iteration the process (0,0) already has calculated

$C_{11}=A_{11}B_{11} + A_{12}B_{21}+ A_{13}B_{31} +A_{14}B_{41}$
$C_{12}=A_{11}B_{12} + A_{12}B_{22}+ A_{13}B_{32} +A_{14}B_{42}$
$C_{17}=A_{11}B_{17} + A_{12}B_{27}+ A_{13}B_{37} +A_{14}B_{47}$
$C_{18}=A_{11}B_{18} + A_{12}B_{28}+ A_{13}B_{38} +A_{14}B_{48}$

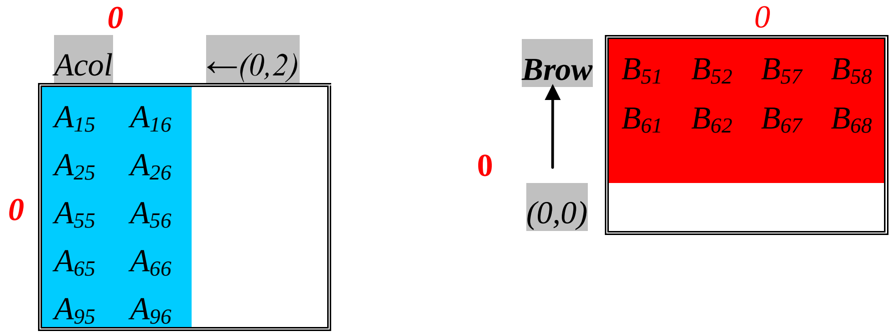
$C_{21}=A_{21}B_{11} + A_{22}B_{21}+ A_{23}B_{31} +A_{24}B_{41}$
$C_{22}=A_{21}B_{12} + A_{22}B_{22}+ A_{23}B_{32} +A_{24}B_{42}$
$C_{27}=A_{21}B_{17} + A_{22}B_{27}+ A_{23}B_{37} +A_{24}B_{47}$
$C_{28}=A_{21}B_{18}+ A_{22}B_{28}+ A_{23}B_{38}+A_{24}B_{48}$

$C_{51}=A_{51}B_{11} + A_{52}B_{21} + A_{53}B_{31} + A_{54}B_{41}$
$C_{52}=A_{51}B_{12} + A_{52}B_{22} + A_{53}B_{32} + A_{54}B_{42}$
$C_{57}=A_{51}B_{17} + A_{52}B_{27} + A_{53}B_{37} + A_{54}B_{47}$
$C_{58}=A_{51}B_{18} + A_{52}B_{28} + A_{53}B_{38} + A_{54}B_{48}$

$C_{61}=A_{61}B_{11} + A_{62}B_{21} + A_{63}B_{31} + A_{64}B_{41}$
$C_{62}=A_{61}B_{12} + A_{62}B_{22} + A_{63}B_{32} + A_{64}B_{42}$
$C_{67}=A_{61}B_{17} + A_{62}B_{27} + A_{63}B_{37} + A_{64}B_{47}$
$C_{68}=A_{61}B_{18} + A_{62}B_{28} + A_{63}B_{38} + A_{64}B_{48}$

$C_{91}=A_{91}B_{11} + A_{92}B_{21} + A_{93}B_{31} + A_{94}B_{41}$
$C_{92}=A_{91}B_{12} + A_{92}B_{22} + A_{93}B_{32} + A_{94}B_{42}$
$C_{97}=A_{91}B_{17} + A_{92}B_{27} + A_{93}B_{37} + A_{94}B_{47}$
$C_{98}=A_{91}B_{18} + A_{92}B_{28} + A_{93}B_{38} + A_{94}B_{48}$

- **Iteration 2.** Here we have for *Acol* and *Bcol* the following:



Then it is possible to calculate the product $C_{(0,0)}= C_{(0,0)}$+Acol*Brow. So we obtain

$C^2_{(0,0)}=$	$C^2_{1,1}= A_{15}B_{51}+ A_{16}B_{61}$ $C^2_{2,1}=A_{25}B_{51}+A_{26}B_{61}$ $C^2_{5,1}=A_{55}B_{51}+A_{56}B_{61}$ $C^2_{61}=A_{65}B_{51}+A_{66}B_{61}$ $C^2_{91}=A_{95}B_{51}+A_{96}B_{61}$	$C^2_{1,2} = A_{15}B_{52}+ A_{16}B_{62}$ $C^2_{22}= A_{25}B_{52}+A_{26}B_{62}$ $C^2_{52}= A_{55}B_{52}+A_{56}B_{62}$ $C^2_{62}= A_{65}B_{52}+A_{66}B_{62}$ $C^2_{92}= A_{95}B_{52}+A_{96}B_{62}$	$C^2_{1,7}= A_{15}B_{57}+ A_{16}B_{67}$ $C^2_{2,7}= A_{25}B_{57}+A_{26}B_{67}$ $C^2_{57}= A_{55}B_{57}+A_{56}B_{67}$ $C^2_{67}= A_{65}B_{57}+A_{66}B_{67}$ $C^2_{97}= A_{95}B_{57}+A_{96}B_{67}$	$C^2_{1,8}= A_{15}B_{58}+ A_{16}B_{68}$ $C^2_{2,8}= A_{25}B_{58}+A_{26}B_{68}$ $C^2_{58}= A_{55}B_{58}+A_{56}B_{68}$ $C^2_{68}= A_{65}B_{58}+A_{66}B_{68}$ $C^2_{98}= A_{95}B_{58}+A_{96}B_{68}$

Therefore, at the end of this iteration the process (0,0) already has calculated

$$C_{11}=A_{11}B_{11} + A_{12}B_{21}+ A_{13}B_{31} +A_{14}B_{41} +A_{15}B_{51}+ A_{16}B_{61}$$
$$C_{12}=A_{11}B_{12} + A_{12}B_{22}+ A_{13}B_{32} +A_{14}B_{42} +A_{15}B_{52}+ A_{16}B_{62}$$
$$C_{17}=A_{11}B_{17} + A_{12}B_{27}+ A_{13}B_{37} +A_{14}B_{47} +A_{15}B_{57}+ A_{16}B_{67}$$
$$C_{18}=A_{11}B_{18} + A_{12}B_{28}+ A_{13}B_{38} +A_{14}B_{48} +A_{15}B_{58}+ A_{16}B_{68}$$

$$C_{21}=A_{21}B_{11} + A_{22}B_{21}+ A_{23}B_{31} +A_{24}B_{41} +A_{25}B_{51}+ A_{26}B_{61}$$
$$C_{22}=A_{21}B_{12} + A_{22}B_{22}+ A_{23}B_{32} +A_{24}B_{42} +A_{25}B_{52}+ A_{26}B_{62}$$
$$C_{27}=A_{21}B_{17} + A_{22}B_{27}+ A_{23}B_{37} +A_{24}B_{47} +A_{25}B_{57}+ A_{26}B_{67}$$
$$C_{28}=A_{21}B_{18}+ A_{22}B_{28}+ A_{23}B_{38}+A_{24}B_{48} +A_{25}B_{58}+ A_{26}B_{68}$$

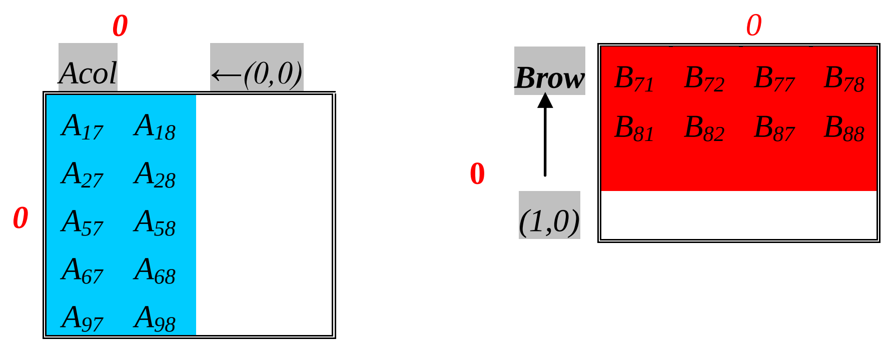
$$C_{51}=A_{51}B_{11} + A_{52}B_{21}+ A_{53}B_{31} +A_{54}B_{41} +A_{55}B_{51}+ A_{56}B_{61}$$
$$C_{52}=A_{51}B_{12} + A_{52}B_{22}+ A_{53}B_{32} +A_{54}B_{42} +A_{55}B_{52}+ A_{56}B_{62}$$
$$C_{57}=A_{51}B_{17} + A_{52}B_{27}+ A_{53}B_{37} +A_{54}B_{47} +A_{55}B_{57}+ A_{56}B_{67}$$
$$C_{58}=A_{51}B_{18}+ A_{52}B_{28}+ A_{53}B_{38}+A_{54}B_{48} +A_{55}B_{58}+ A_{56}B_{68}$$

$$C_{61}=A_{61}B_{11} + A_{62}B_{21}+ A_{63}B_{31} +A_{64}B_{41} +A_{65}B_{51}+ A_{66}B_{61}$$

$C_{62}=A_{61}B_{12} + A_{62}B_{22} + A_{63}B_{32} + A_{64}B_{42} + A_{65}B_{52} + A_{66}B_{62}$
$C_{67}=A_{61}B_{17} + A_{62}B_{27} + A_{63}B_{37} + A_{64}B_{47} + A_{65}B_{57} + A_{66}B_{67}$
$C_{68}=A_{61}B_{18} + A_{62}B_{28} + A_{63}B_{38} + A_{64}B_{48} + A_{65}B_{58} + A_{66}B_{68}$

$C_{91}=A_{91}B_{11} + A_{92}B_{21} + A_{93}B_{31} + A_{94}B_{41} + A_{95}B_{51} + A_{96}B_{61}$
$C_{92}=A_{91}B_{12} + A_{92}B_{22} + A_{93}B_{32} + A_{94}B_{42} + A_{95}B_{52} + A_{96}B_{62}$
$C_{97}=A_{91}B_{17} + A_{92}B_{27} + A_{93}B_{37} + A_{94}B_{47} + A_{95}B_{57} + A_{96}B_{67}$
$C_{98}=A_{91}B_{18} + A_{92}B_{28} + A_{93}B_{38} + A_{94}B_{48} + A_{95}B_{58} + A_{96}B_{68}$

- **Iteration 3.** Here we have for Acol and Bcol the following:



Then it is possible to calculate the product $C_{(0,0)}= C_{(0,0)}$+Acol*Brow. So we obtain

$C^3_{(0,0)}=$	$C^3_{1,1}=A_{17}B_{71}+A_{18}B_{81}$ $C^3_{2,1}=A_{27}B_{71}+A_{28}B_{81}$ $C^3_{5,1}=A_{57}B_{71}+A_{58}B_{81}$ $C^3_{61}=A_{67}B_{71}+A_{68}B_{81}$ $C^3_{91}=A_{97}B_{71}+A_{98}B_{81}$	$C^3_{1,2}=A_{17}B_{72}+A_{18}B_{82}$ $C^3_{22}= A_{27}B_{72}+A_{28}B_{82}$ $C^3_{52}= A_{57}B_{72}+A_{58}B_{82}$ $C^3_{62}= A_{67}B_{72}+A_{68}B_{82}$ $C^3_{92}= A_{97}B_{72}+A_{98}B_{82}$	$C^3_{1,7}=A_{17}B_{77}+ A_{18}B_{87}$ $C^3_{2,7}= A_{27}B_{77}+A_{28}B_{87}$ $C^3_{57}= A_{57}B_{77}+A_{58}B_{87}$ $C^3_{67}= A_{67}B_{77}+A_{68}B_{87}$ $C^3_{97}= A_{97}B_{77}+A_{98}B_{87}$	$C^3_{1,8}= A_{17}B_{78}+ A_{18}B_{88}$ $C^3_{2,8}= A_{27}B_{78}+A_{28}B_{88}$ $C^3_{58}= A_{57}B_{78}+A_{58}B_{88}$ $C^3_{68}= A_{67}B_{78}+A_{68}B_{88}$ $C^3_{98}= A_{97}B_{78}+A_{98}B_{88}$

Therefore, at the end of this iteration the process (0,0) already has calculated

$C_{11}=A_{11}B_{11} + A_{12}B_{21} + A_{13}B_{31} + A_{14}B_{41} + A_{15}B_{51} + A_{16}B_{61} + A_{17}B_{71} + A_{18}B_{81}$
$C_{12}=A_{11}B_{12} + A_{12}B_{22} + A_{13}B_{32} + A_{14}B_{42} + A_{15}B_{52} + A_{16}B_{62} + A_{17}B_{72} + A_{18}B_{82}$
$C_{17}=A_{11}B_{17} + A_{12}B_{27} + A_{13}B_{37} + A_{14}B_{47} + A_{15}B_{57} + A_{16}B_{67} + A_{17}B_{77} + A_{18}B_{87}$
$C_{18}=A_{11}B_{18} + A_{12}B_{28} + A_{13}B_{38} + A_{14}B_{48} + A_{15}B_{58} + A_{16}B_{68} + A_{17}B_{78} + A_{18}B_{88}$

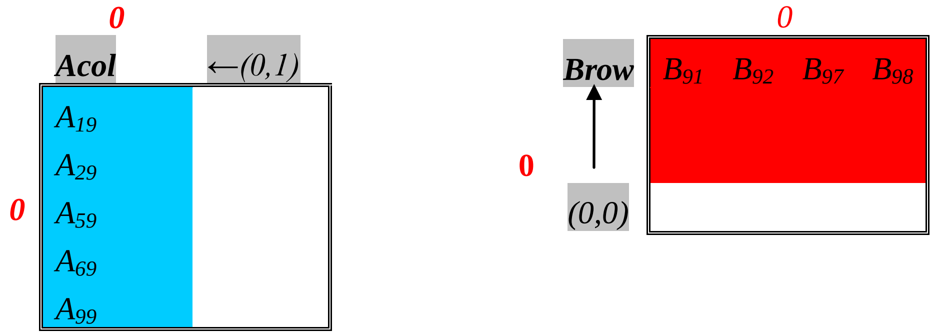
$C_{21}=A_{21}B_{11} + A_{22}B_{21}+ A_{23}B_{31} +A_{24}B_{41} +A_{25}B_{51}+ A_{26}B_{61}+ A_{27}B_{71} +A_{28}B_{81}$
$C_{22}=A_{21}B_{12} + A_{22}B_{22}+ A_{23}B_{32} +A_{24}B_{42} +A_{25}B_{52}+ A_{26}B_{62}+ A_{27}B_{72} +A_{28}B_{82}$
$C_{27}=A_{21}B_{17} + A_{22}B_{27}+ A_{23}B_{37} +A_{24}B_{47} +A_{25}B_{57}+ A_{26}B_{67}+ A_{27}B_{77} +A_{28}B_{87}$
$C_{28}=A_{21}B_{18}+ A_{22}B_{28}+ A_{23}B_{38}+A_{24}B_{48} +A_{25}B_{58}+ A_{26}B_{68}+ A_{27}B_{77} +A_{28}B_{88}$

$C_{51}=A_{51}B_{11} + A_{52}B_{21}+ A_{53}B_{31} +A_{54}B_{41} +A_{55}B_{51}+ A_{56}B_{61}+ A_{57}B_{71} +A_{58}B_{81}$
$C_{52}=A_{51}B_{12} + A_{52}B_{22}+ A_{53}B_{32} +A_{54}B_{42} +A_{55}B_{52}+ A_{56}B_{62}+ A_{57}B_{72} +A_{58}B_{82}$
$C_{57}=A_{51}B_{17} + A_{52}B_{27}+ A_{53}B_{37} +A_{54}B_{47} +A_{55}B_{57}+ A_{56}B_{67}+ A_{57}B_{77} +A_{58}B_{87}$
$C_{58}=A_{51}B_{18}+ A_{52}B_{28}+ A_{53}B_{38}+A_{54}B_{48} +A_{55}B_{58}+ A_{56}B_{68}+ A_{57}B_{78} +A_{58}B_{88}$

$C_{61}=A_{61}B_{11} + A_{62}B_{21}+ A_{63}B_{31} +A_{64}B_{41} +A_{65}B_{51}+ A_{66}B_{61}+ A_{67}B_{71} +A_{68}B_{81}$
$C_{62}=A_{61}B_{12} + A_{62}B_{22}+ A_{63}B_{32} +A_{64}B_{42} +A_{65}B_{52}+ A_{66}B_{62}+ A_{67}B_{72} +A_{68}B_{82}$
$C_{67}=A_{61}B_{17} + A_{62}B_{27}+ A_{63}B_{37} +A_{64}B_{47} +A_{65}B_{57}+ A_{66}B_{67}+ A_{67}B_{77} +A_{68}B_{87}$
$C_{68}=A_{61}B_{18}+ A_{62}B_{28}+ A_{63}B_{38}+A_{64}B_{48} +A_{65}B_{58}+ A_{66}B_{68}+ A_{67}B_{78} +A_{68}B_{88}$

$C_{91}=A_{91}B_{11} + A_{92}B_{21}+ A_{93}B_{31} +A_{94}B_{41} +A_{95}B_{51}+ A_{96}B_{61}+ A_{97}B_{71} +A_{98}B_{81}$
$C_{92}=A_{91}B_{12} + A_{92}B_{22}+ A_{93}B_{32} +A_{94}B_{42} +A_{95}B_{52}+ A_{96}B_{62}+ A_{97}B_{72} +A_{98}B_{82}$
$C_{97}=A_{91}B_{17} + A_{92}B_{27}+ A_{93}B_{37} +A_{94}B_{47} +A_{95}B_{57}+ A_{96}B_{67}+ A_{97}B_{77} +A_{98}B_{87}$
$C_{98}=A_{91}B_{18}+ A_{92}B_{28}+ A_{93}B_{38}+A_{94}B_{48} +A_{95}B_{58}+ A_{96}B_{68}+ A_{97}B_{78} +A_{98}B_{88}$

- **Iteration 4.** Here we have for *Acol* and *Bcol* the following:



Then it is possible to calculate the product C(0,0)= C(0,0)+Acol*Brow. So we obtain

	$C^4_{1,1}= A_{19}B_{91}$	$C^4_{1,2} = A_{19}B_{92}$	$C^4_{1,7}= A_{19}B_{97}$	$C^4_{1,8}= A_{19}B_{98}$
	$C^4_{2,1}=A_{29}B_{91}$	$C^4_{22}= A_{29}B_{92}$	$C^4_{2,7}= A_{29}B_{97}$	$C^4_{2,8}= A_{29}B_{98}$
$C^4_{(0,0)}=$	$C^4_{5,1}=A_{59}B_{91}$	$C^4_{52}= A_{59}B_{92}$	$C^4_{57}= A_{59}B_{97}$	$C^4_{58}= A_{59}B_{98}$
	$C^4_{61}=A_{69}B_{91}$	$C^4_{62}= A_{69}B_{92}$	$C^4_{67}= A_{69}B_{97}$	$C^4_{68}= A_{69}B_{98}$
	$C^4_{91}=A_{99}B_{91}$	$C^4_{92}= A_{99}B_{92}$	$C^4_{97}= A_{99}B_{97}$	$C^4_{98}= A_{99}B_{98}$

Therefore, at the end of this iteration the process (0,0) already has calculated

$C_{11}=A_{11}B_{11} + A_{12}B_{21}+ A_{13}B_{31} +A_{14}B_{41} +A_{15}B_{51}+ A_{16}B_{61}+ A_{17}B_{71} +A_{18}B_{81} +A_{19}B_{91}$
$C_{12}=A_{11}B_{12} + A_{12}B_{22}+ A_{13}B_{32} +A_{14}B_{42} +A_{15}B_{52}+ A_{16}B_{62}+ A_{17}B_{72} +A_{18}B_{82} +A_{19}B_{92}$
$C_{17}=A_{11}B_{17} + A_{12}B_{27}+ A_{13}B_{37} +A_{14}B_{47} +A_{15}B_{57}+ A_{16}B_{67}+ A_{17}B_{77} +A_{18}B_{87} +A_{19}B_{97}$
$C_{18}=A_{11}B_{18} + A_{12}B_{28}+ A_{13}B_{38} +A_{14}B_{48} +A_{15}B_{58}+ A_{16}B_{68}+ A_{17}B_{78} +A_{18}B_{88} +A_{19}B_{98}$

$C_{21}=A_{21}B_{11} + A_{22}B_{21}+ A_{23}B_{31} +A_{24}B_{41} +A_{25}B_{51}+ A_{26}B_{61}+ A_{27}B_{71} +A_{28}B_{81} +A_{29}B_{91}$
$C_{22}=A_{21}B_{12} + A_{22}B_{22}+ A_{23}B_{32} +A_{24}B_{42} +A_{25}B_{52}+ A_{26}B_{62}+ A_{27}B_{72} +A_{28}B_{82} +A_{29}B_{92}$
$C_{27}=A_{21}B_{17} + A_{22}B_{27}+ A_{23}B_{37} +A_{24}B_{47} +A_{25}B_{57}+ A_{26}B_{67}+ A_{27}B_{77} +A_{28}B_{87} +A_{29}B_{97}$
$C_{28}=A_{21}B_{18}+ A_{22}B_{28}+ A_{23}B_{38}+A_{24}B_{48} +A_{25}B_{58}+ A_{26}B_{68}+ A_{27}B_{77} +A_{28}B_{88} +A_{29}B_{98}$

$C_{51}=A_{51}B_{11} + A_{52}B_{21}+ A_{53}B_{31} +A_{54}B_{41} +A_{55}B_{51}+ A_{56}B_{61}+ A_{57}B_{71} +A_{58}B_{81} +A_{59}B_{91}$
$C_{52}=A_{51}B_{12} + A_{52}B_{22}+ A_{53}B_{32} +A_{54}B_{42} +A_{55}B_{52}+ A_{56}B_{62}+ A_{57}B_{72} +A_{58}B_{82} +A_{59}B_{92}$

$C_{57}=A_{51}B_{17} + A_{52}B_{27}+ A_{53}B_{37} +A_{54}B_{47} +A_{55}B_{57}+ A_{56}B_{67}+ A_{57}B_{77} +A_{58}B_{87} +A_{59}B_{97}$

$C_{58}=A_{51}B_{18}+ A_{52}B_{28}+ A_{53}B_{38}+A_{54}B_{48} +A_{55}B_{58}+ A_{56}B_{68}+ A_{57}B_{78} +A_{58}B_{88} +A_{59}B_{98}$

$C_{61}=A_{61}B_{11} + A_{62}B_{21}+ A_{63}B_{31} +A_{64}B_{41} +A_{65}B_{51}+ A_{66}B_{61}+ A_{67}B_{71} +A_{68}B_{81} +A_{69}B_{91}$

$C_{62}=A_{61}B_{12} + A_{62}B_{22}+ A_{63}B_{32} +A_{64}B_{42} +A_{65}B_{52}+ A_{66}B_{62}+ A_{67}B_{72} +A_{68}B_{82} +A_{69}B_{92}$

$C_{67}=A_{61}B_{17} + A_{62}B_{27}+ A_{63}B_{37} +A_{64}B_{47} +A_{65}B_{57}+ A_{66}B_{67}+ A_{67}B_{77} +A_{68}B_{87} +A_{69}B_{97}$

$C_{68}=A_{61}B_{18}+ A_{62}B_{28}+ A_{63}B_{38}+A_{64}B_{48} +A_{65}B_{58}+ A_{66}B_{68}+ A_{67}B_{77} +A_{68}B_{88} +A_{69}B_{98}$

$C_{91}=A_{91}B_{11} + A_{92}B_{21}+ A_{93}B_{31} +A_{94}B_{41} +A_{95}B_{51}+ A_{96}B_{61}+ A_{97}B_{71} +A_{98}B_{81} +A_{99}B_{91}$

$C_{92}=A_{91}B_{12} + A_{92}B_{22}+ A_{93}B_{32} +A_{94}B_{42} +A_{95}B_{52}+ A_{96}B_{62}+ A_{97}B_{72} +A_{98}B_{82} +A_{99}B_{92}$

$C_{97}=A_{91}B_{17} + A_{92}B_{27}+ A_{93}B_{37} +A_{94}B_{47} +A_{95}B_{57}+ A_{96}B_{67}+ A_{97}B_{77} +A_{98}B_{87} +A_{99}B_{97}$

$C_{98}=A_{91}B_{18}+ A_{92}B_{28}+ A_{93}B_{38}+A_{94}B_{48} +A_{95}B_{58}+ A_{96}B_{68}+ A_{97}B_{78} +A_{98}B_{88} +A_{99}B_{98}$

Some results are the following

```
[MI_gr_TPS1@hpc]$./mpiCC_ScL -o Example2.9.1.exe Example2.9.1.cpp
[MI_gr_TPS1@hpc]$/opt/openmpi/bin/mpirun -n 4 -host compute-0-0,compute-0-1
                 Example2.9.1.exe
[MI_gr_TPS1@hpc]$./mpiCC_ScL -o pmm.exe pmm.cpp
[MI_gr_TPS1@hpc]$/opt/openmpi/bin/mpirun -n 4 -host compute-0-0,compute-0-1
                 pmm.exe

Global matrix AA:
  0   1   2   3   4
  1   2   3   4   5
  2   3   4   5   6
  3   4   5   6   7
  4   5   6   7   8
Global matrix BB:
  0   1   2   3   4
  1   2   3   4   5
  2   3   4   5   6
  3   4   5   6   7
  4   5   6   7   8
C_loc on node 0
 30  40  70
 40  55 100
 70 100 190
C_loc on node 1
 50  60
 70  85
```

```
130 160
C_loc on node 2
 50  70 130
 60  85 160
C_loc on node 3
 90 110
110 135
Global matrix CC on node 0
 30  40   0   0  70
 40  55   0   0 100
  0   0   0   0   0
  0   0   0   0   0
 70 100   0   0 190
Global matrix CC on node 1
  0   0  50  60   0
  0   0  70  85   0
  0   0   0   0   0
  0   0   0   0   0
  0   0 130 160   0
Global matrix CC on node 2
  0   0   0   0   0
  0   0   0   0   0
 50  70   0   0 130
 60  85   0   0 160
  0   0   0   0   0
Global matrix CC on node 3
  0   0   0   0   0
  0   0   0   0   0
  0   0  90 110   0
  0   0 110 135   0
  0   0   0   0   0
```

```
[MI_gr_TPS1@hpc]$./mpiCC_ScL -o Example2.9.2.exe Example2.9.2.cpp
[MI_gr_TPS1@hpc]$ /opt/openmpi/bin/mpirun -n 4 -host compute-0-0,compute-0-4
Example2.9.2.exe
Global matrix AA:
  0   1   2   3   4
 10  11  12  13  14
 20  21  22  23  24
 30  31  32  33  34
 40  41  42  43  44

Global matrix BB:
  5   6   7   8   9
 15  16  17  18  19
 25  26  27  28  29
 35  36  37  38  39
 45  46  47  48  49

Local A(3*3) on node 0 (0,0)
  0   1   4
 10  11  14
 40  41  44
Local B(3*3) on node 0 (0,0)
```

```
  5   6   9
 15  16  19
 45  46  49

Local A(2*3) on node 1 (0,1)
 20  21  24
 30  31  34
Local B(2*3) on node 1 (0,1)
 25  26  29
 35  36  39

Local A(3*2) on node 2 (1,0)
  2   3
 12  13
 42  43
Local B(3*2) on node 2 (1,0)
  7   8
 17  18
 47  48

Local A(2*2) on node 3 (1,1)
 22  23
 32  33
Local B(2*2) on node 3 (1,1)
 27  28
 37  38

Local C(3*3) on node 0 (0,0)
800 1800 4800
835 1885 5035
940 2140 5740
Local C(3*2) on node 1 (0,1)
2800 3800
2935 3985
3340 4540
Local C(2*3) on node 2 (1,0)
870 1970 5270
905 2055 5505
Local C(2*2) on node 3 (1,1)
3070 4170
3205 4355
Global matrix CC =AA*BB:
  800  1800  2800  3800  4800
  835  1885  2935  3985  5035
  870  1970  3070  4170  5270
  905  2055  3205  4355  5505
  940  2140  3340  4540  5740
```