Chapter 2

Matrix Multiplication using Mesh Networks

2.1 Objective:

- At the end of the lecture, the learner would be able to:
  - Define a two dimension mesh networks in different forms
  - Evaluate a two dimensional mesh network without wraparound connections
  - Understand a method to multiply two matrices using two dimensional mesh.

2.2 Mesh Network

First let us discuss the definition of a mesh network.
Mesh Network: A set of nodes arranged in the form of a p dimensional lattice is called a mesh network. In a mesh network only neighbouring nodes can communicate with each other. Therefore interior nodes can communicate with 2p other nodes.

Example of a Mesh Network

Figure 2.1: A 2 Dimensional Mesh with and without wraparound connections

2.2.1 Evaluation of a Mesh Network

Let us assume that the mesh has no wraparound connection. A mesh network can be evaluated using two factors (i) Diameter and (ii)Bisection width.

(i) Diameter:A diameter of a mesh network is the longest distance between two nodes. The diameter of a p dimensional mesh with k^p nodes is p(k-1).
(ii)Bisection width: It is defined as the minimum number of edges to be removed to divide the network into two halves. The bisection width of a p dimensional mesh with k^p nodes is k^{p-1}.

2.2.2 Matrix Multiplication using 2D Mesh

Given a 2D mesh SIMD model with wraparound connections it is easy to devise an algorithm that uses n^2 processors to multiply two nxn arrays in theta(n) time.

Initial location of matrix elements to processing elements

Let each Processing Element $PE_{i,j}$ represents two elements $a_{i,j}$, $b_{i,j}$

In the original state there are only n processing elements containing a pair of scalars suitable for multiplication.

Stagger matrices A and B so that every processor has a pair of scalars that need to be multiplied.

The elements of A will move in leftward rotation and the elements of B move in upward rotation. These movements present each PE with a new pair of values to be multiplied.

Now let us look at the actions of a single processing element. After matrices A and B have been staggered, the PE P(1,2) performs the multiplications and additions form the dot product $C_{1,2}$.

Let us look at the following animation

Figure 2.2: Staggering of Matrix elements

The pseudo code for the matrix multiplication is given below. The first phase of the parallel algorithm staggers two matrices. The second phase computes all products a_{ik} X b_{kj} and accumulate sums when the phase II is complete.

Algorithm

```
Procedure MATRIXMULT
begin
  for k = 1 to n-1 step 1 do
  begin
    for all $P_{i,j}$ where i and j ranges from 1 to n do
      if i is greater than k then
        rotate a in the east direction
      end if
      if j is greater than k then
```

```
rotate b in the south direction
end if
end
for all $P_{i,j}$ where i and j lies between 1 and n do
compute the product of a and b and store it in c
for k= 1 to n-1 step 1 do for all $P_{i,j}$ where i and j ranges from 1 to n do
rotate a in the east
rotate b in the south
c=c+aXb
end
```

Explanation with an example

Let us consider the following two matrices.

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

$$B = \begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix}$$

The matrix elements are stored in a two dimensional mesh as follows. The elements of the matrix A are in red and the elements of the matrix B are in green.

$$Mesh = \begin{pmatrix} a11|b11 & a12|b12 \\ a21|b21 & a22|b22 \end{pmatrix}$$

That is,

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

Let the number of processors be n^2=4. Let the processors be P(1,1), P(1,2), P(2,1) and P(2,2). Let k ranges from 1 to 2-1=1.
According to the algorithm, P(1,2), P(2,1) and P(2,2) will carry out the movements. That is there is no change in a11 and b11. b12 and b22 are moved up while a21 and a22 are moved left. Now the resultant Mesh after column movement is given below.

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

Now the element 6 wraparound to the bottom.

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

Now P(2,1) and P(2,2) will move $a_{2,1}$ and $a_{2,2}$ towards its left respectively. Therefore the resultant Mesh will be

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

The processors $P_{i,j}$ i,j ranges from 1 to 2 will compute the product of $a_{i,j}$ and $b_{i,j}$ respectively
The resultant matrix after computing the product will be

$$C = \begin{pmatrix} 5 & 16 \\ 28 & 18 \end{pmatrix}$$

Next all $P_{i,j}$ will move its $a_{i,j}$ and $b_{i,j}$ left and up respectively.
The resultant Mesh will be

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

Now all the processors $P_{i,j}$ compute the product of $a_{i,j}$ and $b_{i,j}$ respectively and add with the old value of $C_{i,j}$.
Hence the resultant product matrix will be

$$C = \begin{pmatrix} 19 & 22 \\ 43 & 50 \end{pmatrix}$$

2.3 Analysis

Communication

Let us assume that the given matrices A and B are divided into sub matrices each of order m x m, where m is the number of processes or processors. Each process will have q x q sub matrices where q=n/m. The initial alignment requires a maximum of q-1 shift operations for both A and B.

Next there will be q-1 shift operations for A and B each. Each shift operation involves m^2 elements.
$T_{Communication}$ = 4(q-1)(startuptime + m^2datatime)=O(qm^2)=O(mn) since q=n/m

Computation

Each sub matrix multiplication requires m^3 multiplication and m^3 addition. Therefore with q-1 shifts, $T_{Computation}$ = $2qm^3$=$2m^2$n. Hence $T_{Computation}$=O(m^2n).