Karthikeyan Karur Balu

Parallel computing

Question 5 (HW 3)

Main Idea

The DNS algorithm is a different modularized approach compared to the traditional approach of matrix multiplication. If the matrix of the order n * n (n rows, n columns), then the DNS algorithms uses n * n * n processors for computing the result. It performs matrix multiplication in time of the order O(logn) because each processor perform the multiplication so we have n * n *n processors. The total multiplication that is done over here is n * n * n and each done by one processor. In such case, all the multiplication is done in O(1). Finally, the addition is done using O(log n). So the total run time is O(log n).

Description

Assume that n * n * n processes are available for multiplying two n * n matrices. These processes are aranged in a three-dimentional n x n x n logical array.
Since the matrix multiplication algorithm performs n3 scalar multiplications, each of the n3 processes is assigned a single scalar multiplication.
The processes are labeled according to their location in the array, and the multiplication A[i,k] B[k,j] is assigned to process P[i,j,k] (0i,j,k<n).
After each process performs a single multiplication, the contents of P[i,j,0],P[i,j,1],…,P[i,j,n-1] are added to obtain C[i,j].
The addition for all C[i,j] can be carried out simultaneously in log n steps each.
Thus, it takes one step to multiply and log n steps to add.
It takes time (log n) to multiply the n x n matrix.

Example

Consider the following Matrix Multiplication A and B

Matrix A

1	2	3
4	5	6
7	8	9

Matrix B

10	11	12
13	14	15
16	17	18

The # in the above matrices are distributed among the processors as described in the above description

C[0,0] = (A[0,0] * B[0,0]) from k = 0 + (A[0,0] * B[0,0]) from k = 1 + (A[0,0] * B[0,0]) from k = 0

Likewise you need to do for all the rows and columns of the matrix.

Matrix A

Matrix B

K = 0 (After moving A[I,j] to P[I,j,0] to P[I,j,j]

1	1	1
4	4	4
7	7	7

10	11	12
10	11	12
10	11	12

K = 1

2	2	2
5	5	5
8	8	8

13	14	15
13	14	15
13	14	15

K = 2

3	3	3
6	6	6
9	9	9

16	17	17
16	17	18
16	17	18

Calculations

C[0,0] = (A[0,0] * B[0,0]) from k = 0 + (A[0,0] * B[0,0]) from k = 1 + (A[0,0] * B[0,0]) from k = 0

C[0,0] = 1 * 10 + 2 * 13 + 3 * 16 = 84

C[0,1] = 1 * 11 + 2 * 14 + 3 * 17 =

C[0,2] = 1 * 12 + 2 * 15 + 3 * 17

Continuing this calculation we get the result as Matrix C

84	90	96
201	216	231
318	342	366