DNS Algorithm

Given

1) Matrix A : Dimension n x n
2) Matrix B : Dimension n x n

Operation

C [][] = C[][] + A[][]*B[][]

Where C [][] is the resultant matrix

Initial configurations of DNS algorithm

Number of processors used: n x n x n

Problem complexity: Ώ (n^3/log n)

DNS algorithm complexity: Ѳ(log n)

Idea of algorithm:

For the initial configuration of the problem is to build up the two input matrices. This initial housekeeping is done using n x n x n processors in a 3-D configuration. So the 3-D DNS partitioning algorithm performs the same computation to perform multiplication on two n x n matrices as performed in the 2-D partitioning method, however the number of processors used to scale down the complexity is n time more than the number of processors used in a typical 2-D algorithm.

Description of algorithm

In this algorithm, for every process P with i rows and j columns performs matrix multiplication for every ith row of the first matrix and jth row of the second matrix. The usual message passing method technique is a one to all broadcast where every jth iterations computation is performed on the inner loop structure, i.e. on the second matrix participating in a product, and every ith iteration participates on the outer loop on the first participating matrix. The inner most loop (the actual multiplication and summation occurs in this loop) is then subjected to reduction to generate the resultant matrix. This computation of generating the resultant matrix is actually from the outer n process in the n x n x n structure which can be visualized as in like a hypercube.

Example

Let us observe the working of this algorithm for a dimension of 2

Matrix 1 population at rank = 0

0,2	1,2	2,2
0,1	1,1	2,1
0,0	1,0	2,0

Subsequent Matrix 1 configurations for ranks 0 , 1 , 2

0,0	1,0	2,0

Rank=0

0,1	1,1	2,1

Rank=1

0,2	1,2	2,2

Rank=2

Similarly the configurations for matrix 2 are populated based on column like the above process.

0,2		
0,1		
0,0		

Rank=0

	1,2	
	1,1	
	1,0	

Rank=1

		2,2
		2,1
		2,0

Rank=2

Thus the resultant matrix 3 will have the following values in the final matrix

1[0,2]X2[2,0]	1[1,2]X2[2,1]	1[2,2]X2[2,2]
1[0,1]X2[1,0]	1[1,1]X2[1,1]	1[2,1]X2[1,2]
1[0,0]X2[0,0]	1[1,0]X2[0,1]	1[2,0]X2[0,2]

The above operation is the summation over k=0, 1, 2 for every index of the matrix. This summation can be visualized as the third dimension across the z-axis in the hypercube structure.