# ASSIGNMENT NO.:

**Title:** Implement Strassen's matrix multiplication model and use HPC architecture (preferably BBB) for its computation.

**Aim:** Implement Strassen's matrix multiplication model and use HPC architecture (preferably BBB) for its computation. The use of MPI routines is done to incorporate parallelism.

**Objective:**

1. To understand concept of Strassen's matrix multiplication.
2. To effectively use multi-core or distributed, concurrent/Parallel environments.
3. To develop problem solving abilities using Mathematical Modeling.
4. To develop time and space efficient algorithms that could incorporate parallelism.

**Theory:**

Matrix multiplication is one of the most basic operations of scientific computing. Conventional algorithm to perform matrix multiplication is of time complexity $O(n^3)$. For multiplication of 2 2×2 matrices, the conventional algorithm involves 8 multiplications and 4 additions.

Following figure give general idea of conventional method of matrix multiplication.

$$\underset{A}{\begin{bmatrix} a & b \\ c & d \end{bmatrix}} \times \underset{B}{\begin{bmatrix} e & f \\ g & h \end{bmatrix}} = \underset{C}{\begin{bmatrix} ae + bg & af + bh \\ ce + dg & cf + dh \end{bmatrix}}$$

A, B and C are square metrices of size N x N
a, b, c and d are submatrices of A, of size N/2 x N/2
e, f, g and h are submatrices of B, of size N/2 x N/2

In 1969, Strassen introduced an algorithm to compute matrix multiplications. Strassen's algorithm uses a clever scheme that involves 7 multiplications and 18 additions. The complexity of Strassen's algorithm is $O(n^{2.807})$, which means it will run faster than the conventional algorithm for sufficiently large matrices.

$$\begin{aligned} p1 &= a(f - h) & p2 &= (a + b)h \\ p3 &= (c + d)e & p4 &= d(g - e) \\ p5 &= (a + d)(e + h) & p6 &= (b - d)(g + h) \\ p7 &= (a - c)(e + f) & & \end{aligned}$$

The A x B can be calculated using above seven multiplications.
Following are values of four sub-matrices of result C

$$\underset{A}{\begin{bmatrix} a & b \\ c & d \end{bmatrix}} \times \underset{B}{\begin{bmatrix} e & f \\ g & h \end{bmatrix}} = \underset{C}{\begin{bmatrix} p5 + p4 - p2 + p6 & p1 + p2 \\ p3 + p4 & p1 + p5 - p3 - p7 \end{bmatrix}}$$

A, B and C are square metrices of size N x N
a, b, c and d are submatrices of A, of size N/2 x N/2
e, f, g and h are submatrices of B, of size N/2 x N/2
p1, p2, p3, p4, p5, p6 and p7 are submatrices of size N/2 x N/2

*Drawbacks:*

- Strassen's algorithm cannot be applied directly unless the matrix has a dimension of $2^{k}$.

- Strassen's algorithm requires additional temporary storage and complicates cache blocking; therefore, in practice it is slower than the conventional algorithm for small size matrices.

We have used Task parallel approach, dividing the calculations of p1,p2… p7 among different cores. There is some communication overhead due to sharing of the computed values by one processor to other processors. Parallel implementation of Strassen's method gives good efficiency when size of matrices is large as communication time is much smaller compared to computational time.

**Algorithm:**

- Take two 2×2 matrices (m1 and m2) as input from user.

  $m1_{00}$ =a, $m1_{01}$ =b, $m1_{10}$ =c, $m1_{11}$=d

  $m2_{00}$ =e, $m2_{01}$ =f, $m2_{10}$ =g, $m1_{11}$=h

  Resultant matrix res=m1*m2

  $res_{00}$ =x, $res_{01}$ =y, $res_{10}$ =z, $res_{11}$=w

- Calculate p1,p2……p7.

  p1 = a(f-h)

  p2 = h(a+b)

  p3 = e(c+d)

  p4 = d(g-e)

  p5 = (a+d)(e+h)

  p6 = (b-d)(g+h)

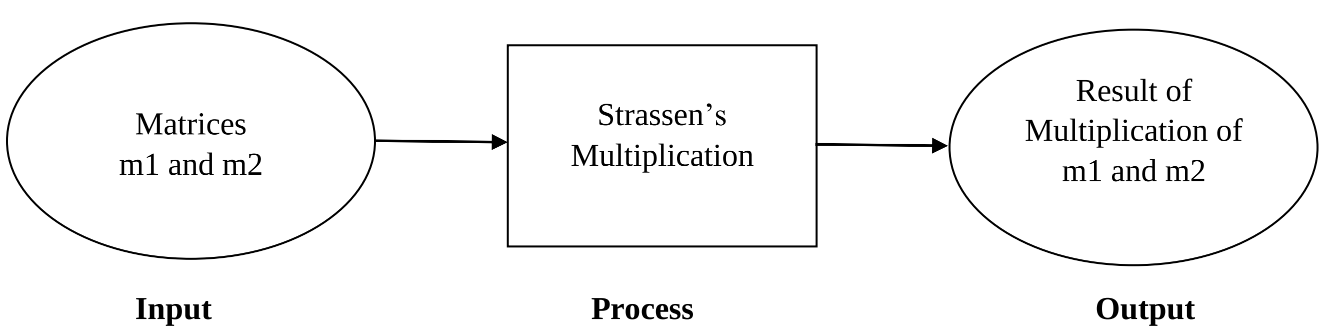
  p7 = (a-c)(e+f)

- Calculate x, y, z and w.

  x = p6+p5+p4-p2
  y = p1+p2
  z = p3+p4
  w = p1+p5-p3-p7

- Display resultant.

**Mathematical Model:**



Let S be the system such that,

S= {I, O, Fn, Sc, Fc}

I-Input Set

Matrices m1 and m2

O-Output set

Multiplication of m1 and m2.

Fn - Function Set

F1 – Calculate p1,p2...p7

F2 – Calculate Resultant

Sc – Success Set

Sc1 – Correct result obtained

Sc2 – Synchronization between all cores

Fc – Failure Cases

Fc1 – No synchronization between cores

Fc2 – multiplication is incorrect

**Input:** Two 2×2 matrices (m1, m2)

**Output:** Result of multiplication of m1 and m2.

**Conclusion:** Hence, we have implemented Strassen's matrix multiplication model using MPI routines.