Design and Analysis of Algorithms CSE2012

Dr. Ramesh Ragala

February 11, 2022

- **Problem:**
  - Assume Two Matrice A and B with Order nxn. Compute the Product of A and B Matrices i.e C = A.B
  - Input: $A = [\ a_{i,j}]$ and $B = [\ b_{i,j}\ ]$
  - Output: $C = [\ c_{i,j}\ ] = A\ .\ B$. *Where i,j = 1, 2, .. n*

$$\begin{bmatrix} c_{11} & c_{12} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ c_{n1} & c_{n2} & \dots & c_{nn} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} X \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ b_{n1} & b_{n2} & \dots & b_{nn} \end{bmatrix}$$

Multiplication Problem

- **Conventional way of Matrix Multiplication:**

$$c_{ij} = \sum_{k=1}^{n} a_{ij}.b_{ij}$$

Algorithm MatrixMul(a,b,c,n)

```
{
for i ← 1 to n do
    for j ← 1 to n do
        c_ij ← 0;
        for k ← 1 to n do
            c_ij ← c_ij + a_ik . b_kj
        end for
    end for
end for
}
```

- **Running time = $\Theta(n^3)$**

- **DAC Approach:**
  - **Divide Step:** Partition the Matrices of Order n x n into $\frac{1}{2}$n X $\frac{1}{2}$n matrices
  - **Conquer Step:** Call the Divide Step, until the matrix order becomes 2 by 2.
  - **Combine Step:** Use 8 scalar multiplications and 4 additions to get resultant matrix. The formulae will be discuss later

$$\begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} X \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}$$

- $\mathrm{C}_{11} = (\mathrm{A}_{11}\mathrm{xB}_{11}) + (\mathrm{A}_{12}\mathrm{xB}_{21})$
- $\mathrm{C}_{12} = (\mathrm{A}_{11}\mathrm{xB}_{12}) + (\mathrm{A}_{12}\mathrm{xB}_{22})$
- $\mathrm{C}_{21} = (\mathrm{A}_{21}\mathrm{xB}_{11}) + (\mathrm{A}_{22}\mathrm{xB}_{21})$
- $\mathrm{C}_{22} = (\mathrm{A}_{21}\mathrm{xB}_{12}) + (\mathrm{A}_{22}\mathrm{xB}_{22})$

- Running Time: $\mathrm{T(n)} = 8\mathrm{T(n/2)} + \theta(n^2)$
- Apply the Master Theorem (case -1) $\mathrm{T(n)} = \theta(n^3)$

Matrix Multiplication

- Algorithm for Matrix Multiplication using DAC Approach:

Algorithm DACMM(A,B,n)

{
Assume **C** be the new Matrix of Order n by n
if $(n = 1)$ **then**
    $C \leftarrow A.B$
else
    $C_{11} \leftarrow DACMM(A_{11}, B_{11}) + DACMM(A_{12}, B_{21})$
    $C_{12} \leftarrow DACMM(A_{11}, B_{12}) + DACMM(A_{12}, B_{22})$
    $C_{21} \leftarrow DACMM(A_{21}, B_{11}) + DACMM(A_{22}, B_{21})$
    $C_{22} \leftarrow DACMM(A_{21}, B_{12}) + DACMM(A_{22}, B_{22})$
end if
}

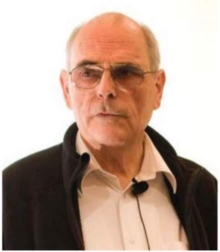
Matrix Multiplication

- Matrix Multiplication using DAC Approach:

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

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

- **Strassen's Matrix Multiplications:**
  - Proposed new approach for Matrix Multiplications (1968)
  - It runs asymptotically faster then $\theta(n^3)$.
  - It also uses DAC approach to solve the Matrix Multiplication
  - In this method, we use 7 recursive multiplications and 18 additions
  - The cost of the scalar multiplication is more than the cost of scalar Addition. So this method reduce the one scalar multiplication.



- **Divide Step:** Partition the Matrices of Order n x n into $\frac{1}{2}$n X $\frac{1}{2}$n matrices
- **Conquer Step:** Call the Divide Step, until the matrix order becomes 2 by 2.
- **Combine Step:** Uses 7 Multiplications and 18 additions / subtraction (11 additions and 7 subtractions) to get the result matrix. The equations or formulae discuss later.

$$\begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} X \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}$$

- The equations to get resultant matrix in strassen's method as follows

$p_1 \leftarrow (A_{11} + A_{22})$ x $(B_{11} + B_{22})$
$p_2 \leftarrow (B_{11})$ x $(A_{21} + A_{22})$
$p_3 \leftarrow (A_{11})$ x $(B_{12} - B_{22})$
$p_4 \leftarrow (A_{22})$ x $(B_{21} - B_{11})$
$p_5 \leftarrow (B_{22})$ x $(A_{11} + A_{12})$
$p_6 \leftarrow (A_{21} - A_{11})$x $(B_{11} + B_{12})$
$p_7 \leftarrow (A_{12} - A_{22})$x $(B_{21} + B_{22})$

$C_{11} \leftarrow p_1 + p_4 + p_7 - p_5$
$C_{12} \leftarrow p_3 + p_5$
$C_{21} \leftarrow p_2 + p_4$
$C_{22} \leftarrow p_1 + p_3 + p_6 - p_2$

- Strassen's Matrix Multiplication approach

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

- Algorithm for Strassen's Matrix Multiplication:

Algorithm STMM(A,B)

```
{
Assume C be the new Matrix of Order n by n
if (n = 1) then
    C ← A.B
else
    p1 ← STMM((A11 + A22), (B11 + B22))
    p2 ← STMM((B11), (A21 + A22))
    p3 ← STMM((A11), (B12 − B22))
    p4 ← STMM((A22), (B21 − B11))
    p5 ← STMM((B22), (A11 + A12))
    p6 ← STMM((A21 − A11), (B11 + B12))
    p7 ← STMM((A12 − A22), (B21 + B22))
end if
}
```

- Example in Strassen's Method

Matrix Multiplication

- Complexity Analysis