

CS471-Parallel Processing-Spring 2020‬‏  
PID23867580

**Topic 2: Matrix Multiplication using Coarse-Grained Parallelism**

|  |  |  |
| --- | --- | --- |
| **Name** | **ID** | **Email** |
| Atef Magdy Mitwally | 20170136 | atefmagdy12@gmail.com |
| Abdulrahman Bahig Mahmoud | 20170143 | abdelrahmanbahig04@gmail.com |
| Ibrahim Ramadan Abdou | 20170002 | ibrahemramadan130@gmail.com |
| Ahmed Sayed Mansour | 20170022 | ahmed.mans20719@gmail.com |
| Hatem Sayed Ali | 20170084 | hatemgad98@gmail.com |

Under the supervision of:

**Dr Sabah Sayed**

**s.sayed@fci-cu.edu.eg**

1.Serial program Algorithm for matrix multiplication

Mat1[m][r]

Mat2[r][n]

Mat3[m][n]

For (i=0 🡪m)  
 For(j=0🡪n)

Mat3[i][j]=0

For(k=0🡪r)

Mat3[i][j] += Mat1[i][k]\*Mat2[k][j]

endFor

endFor

endFor

2. Applying Foster’s Methodology

1. Partitioning:

Step1: set result matrix by zero

Mat3[n-1][m-1]=0

Mat3[0][1]=0

Mat3[0][0]=0

……

Step2: calculate each element

Mat3[0][0]+=

mat1[0][r-1]\*mat2[r-1][0]

Mat3[0][0]+=mat1[0][1]\*mat2[1][0]

Mat3[0][0]+=mat1[0][0]\*mat2[0][0]

…

Mat3[0][1]+=mat1[0][1]\*mat2[1][1]

Mat3[0][1]+=

mat1[0][r-1]\*mat2[r-1][1]

Mat3[0][1]+=mat1[0][0]\*mat2[0][1]

…

…

Mat3[0][r-1]+=

mat1[0][0]\*mat2[0][0]

Mat3[0][r-1]+=

mat1[0][1]\*mat2[1][0]

Mat3[0][r-1]+=

mat1[0][r-1]\*mat2[r-1][0]

…

….

…

Mat3[m-1][n-1]+=

mat1[m-1][1]\*mat2[1][n-1]

Mat3[m-1][n-1]+=

mat1[m-1][0]\*mat2[0][n-1]

‘

Mat3[m-1][n-1]+=

mat1[m-1][r-1]\*mat2[r-1][n-1]

2. Communicate:

To calculate each element it must be 0 first then multiply mat1.row by mat2.column element by element.

….

Mat3[0][1]=0

Mat3[0][0]=0

…

Mat3[0][0]+=

mat1[0][r-1]\*mat2[r-1][0]

Mat3[0][0]+=mat1[0][1]\*mat2[1][0]

Mat3[0][0]+=mat1[0][0]\*mat2[0][0]

…

Mat3[0][1]+=

mat1[0][r-1]\*mat2[r-1][1]

Mat3[0][1]+=mat1[0][1]\*mat2[1][1]

Mat3[0][1]+=mat1[0][0]\*mat2[0][1]

..

3. Agglomeration:

We can aggregate set element by 0 then calculate its value as one task instead of 2 tasks and then send it to other process.

Mat3[0][0]=0

Mat3[0][0]+=mat1[0][1]\*mat2[1][0]

Mat3[0][0]+=

mat1[0][r-1]\*mat2[r-1][0]

Mat3[0][0]+=mat1[0][0]\*mat2[0][0]

4. Mapping:

Each thread is responsible for calculate a row or number of rows and send it to master to gathering them.   
It would be Coarse-Grained Parallelism because each thread takes a few large tasks.

3. Parallel program Algorithm for matrix multiplication

Mat1[m][r]

Mat2[r][n]

Mat3[m][n]

P //Num of processes

start[p]  
end[p]

For (f=0🡪P)

For (i=start[f] 🡪end[f])

For(j=0🡪n)

Mat3[i][j]=0

For(k=0🡪r)

Mat3[i][j] += Mat1[i][k]\*Mat2[k][j]

endFor

endFor

endFor  
endFor