**PARALLEL AND DISTRIBUTED COMPUTING LAB**

**REPORT**

**NAME:** S Shyam Sundaram

**REG NO:** 19BCE1560

**PROGRAMMING ENVIRONMENT:** MPI

**PROBLEM:** MPI

**DATE:** 17th November, 2021

**HARDWARE CONFIGURATION:**

|  |  |  |  |
| --- | --- | --- | --- |
| CPU NAME | | : | Intel core i5 – 1035G1 @ 1.00 Ghz |
| Number of Sockets: | | : | 1 |
| Cores per Socket | | : | 4 |
| Threads per core | | : | 1 |
| L1 | Cache size | : | 320KB |
| L2 | Cache size | : | 2MB |
| L3 | Cache size (Shared): | | 6MB |
| RAM | | : | 8 GB |

**QUESTION**

Write an MPI program that reads the RGB matrices of an image and find the average of these three, thereby converting the image to a grayscale version.

**CODE**

**avg.c**

#include <mpi.h>

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

const int n = 479; //height of image

const int m = 500; //width of image

void readMatrix(int matrix[n][m],int color)

{

    FILE \*fp;

    if(color==0) //0: red, 1: green, anything else: blue

    fp = fopen("r.txt", "r");

    else if(color==1)

    fp = fopen("g.txt", "r");

    else

    fp = fopen("b.txt", "r");

    int r=0,c=0;

    int x;

    while((fscanf(fp,"%d,",&x)!=EOF))

    {

        //printf("%d-",x);

        if(c==m)

        {

            r++;

            c=0;

        }

        matrix[r][c++]=x;

    }

    fclose(fp);

}

void writeMatrix(int matrix[n][m])

{

    FILE \*fp;

    fp=fopen("final.txt","a");

    for(int i=0;i<n;++i)

    {

        for(int j=0;j<m;++j)

        {

            fprintf(fp,"%d, ",matrix[i][j]);

        }

        fprintf(fp,"\n");

    }

    fclose(fp);

}

int main(int argc, char\* argv[])

{

    int r[n][m],g[n][m],b[n][m];

    int id = 0;

    int comm\_size = 0;

    int final[n][m];

    int start,siz;

    readMatrix(r,0);

    readMatrix(g,1);

    readMatrix(b,2);

    double t1, t2;

    t1 = MPI\_Wtime();

    MPI\_Init(&argc, &argv);

    MPI\_Comm\_rank(MPI\_COMM\_WORLD, &id);

    MPI\_Comm\_size(MPI\_COMM\_WORLD, &comm\_size);

    start=n%comm\_size;

    siz=n/comm\_size;

    if(id==0)

    {

        readMatrix(r,0);

        readMatrix(g,1);

        readMatrix(b,2);

        printf("Name: S Shyam Sundaram\nReg num: 19BCE1560\n\n");

        printf("%d and %d\n",start,siz);

        if(start!=0)

        {

            for(int i=0;i<start;++i)

            for(int j=0;j<m;++j)

            final[i][j]=(r[i][j]+g[i][j]+b[i][j])/3;

        }

    }

    int rrow[m\*siz],grow[m\*siz],brow[m\*siz], avg[m\*siz];

    MPI\_Scatter(r[start], siz\*m, MPI\_INT, rrow, siz\*m, MPI\_INT, 0, MPI\_COMM\_WORLD);

    MPI\_Scatter(g[start], siz\*m, MPI\_INT, grow, siz\*m, MPI\_INT, 0, MPI\_COMM\_WORLD);

    MPI\_Scatter(b[start], siz\*m, MPI\_INT, brow, siz\*m, MPI\_INT, 0, MPI\_COMM\_WORLD);

    for(int k=0;k<m\*siz;++k)

        avg[k]=(rrow[k]+grow[k]+brow[k])/3;

    if(start<m)

    MPI\_Gather(avg,siz\*m,MPI\_INT,final[start],siz\*m,MPI\_INT,0,MPI\_COMM\_WORLD);

    if(id==0)

    {

        printf("Final is matrix written to file \n");

        writeMatrix(final);

    }

    MPI\_Finalize();

    t2 = MPI\_Wtime();

    printf( "Elapsed time is %f\n", t2 - t1 );

    return 0;

}

**getrgb.py**

import numpy as np

from PIL import Image

img=Image.open('img1.jpg')

arr=np.array(img)

print(arr.shape)

print(arr)

r=open("r.txt","a")

g=open("g.txt","a")

b=open("b.txt","a")

for i in range(arr.shape[0]):

    for j in range(arr.shape[1]):

        r.write(str(arr[i][j][0]).rstrip('\n')+", ")

        g.write(str(arr[i][j][1]).rstrip('\n')+", ")

        b.write(str(arr[i][j][2]).rstrip('\n')+", ")

    r.write("\n")

    g.write("\n")

    b.write("\n")

r.close()

g.close()

b.close()

**writeimg.py**

import numpy as np

import matplotlib.pyplot as plt

f=open("final.txt","r")

l=f.readlines()

pix=[]

for i in range(len(l)):

    li=[int(x) for x in l[i].split(", ")[:-1]]

    pix.append(li)

pix=np.array(pix)

print(pix.shape)

print(pix)

plt.imshow(pix, cmap="gray")

plt.show()

**COMMANDS**

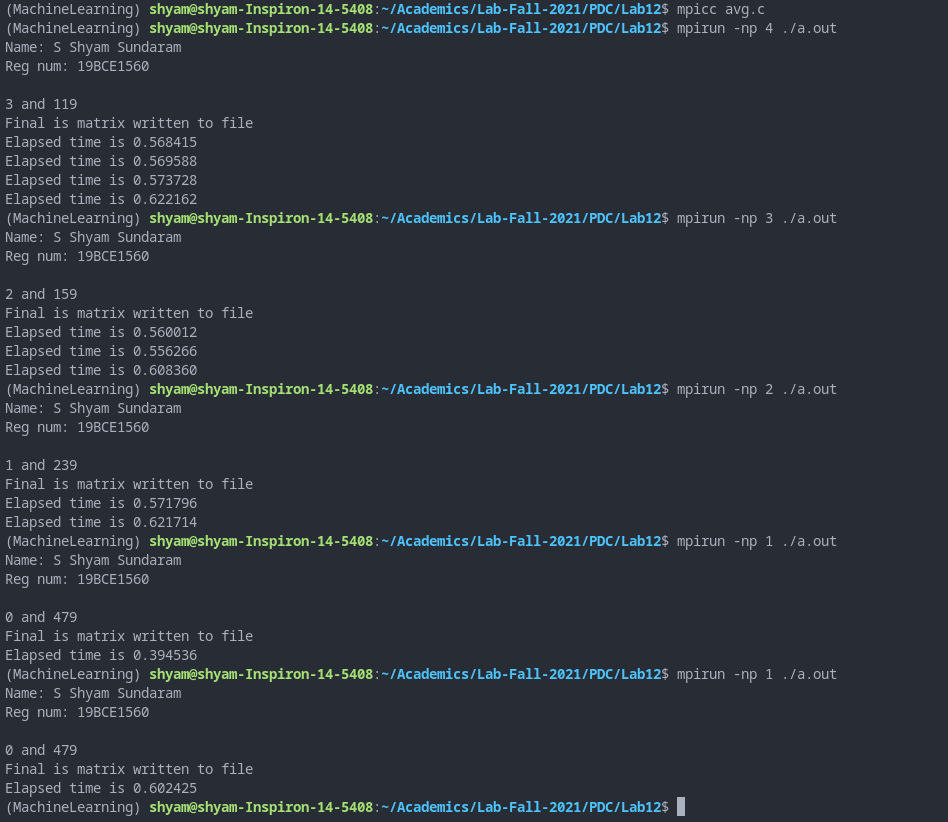
python getrgb.py

mpicc avg.c

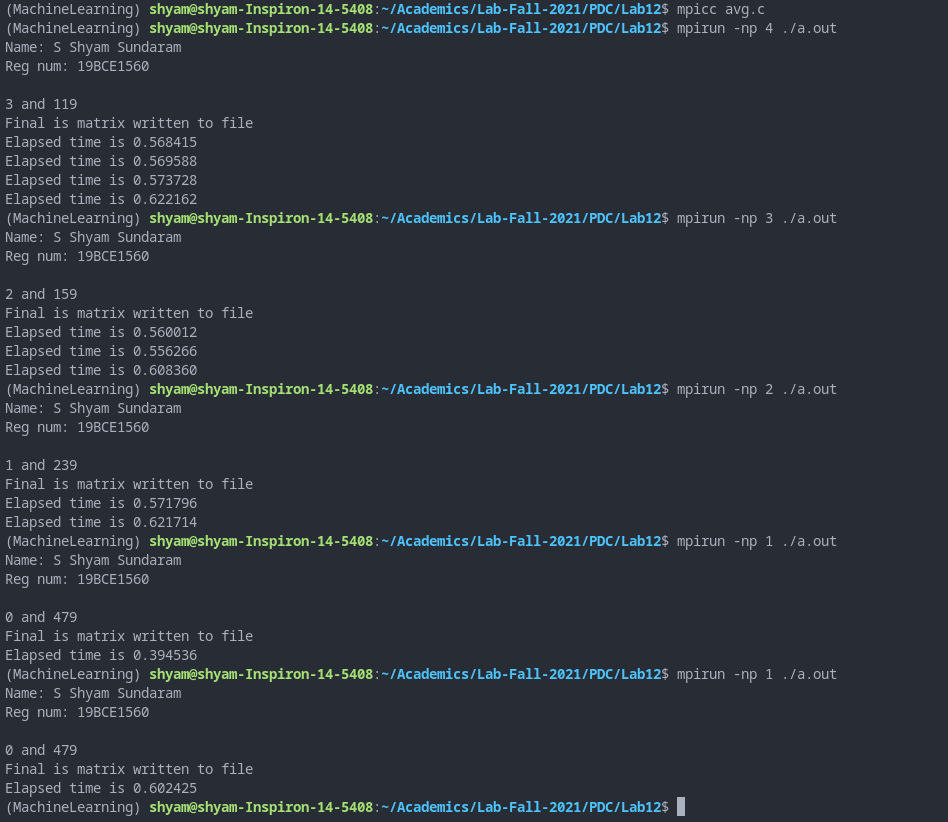
mpirun --oversubscribe -np 4 ./a.out

python writeimg.py

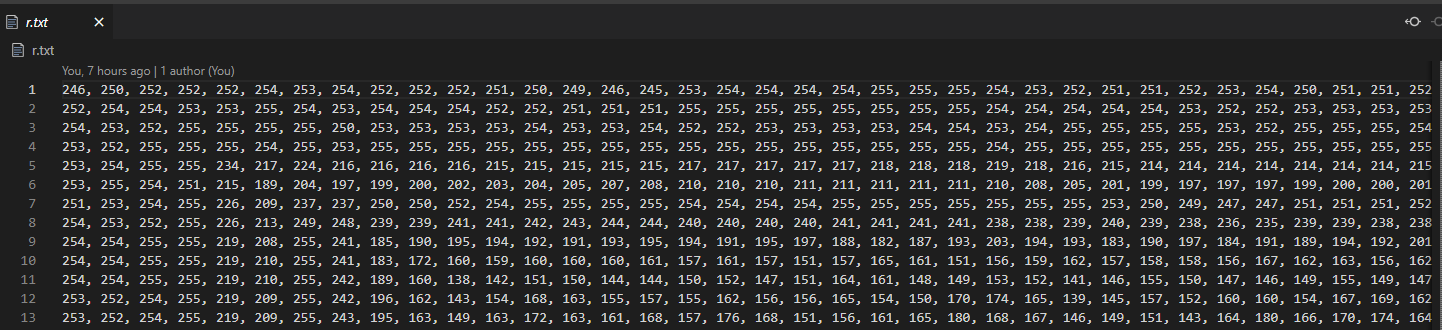
**OUTPUT**

****

avg.c with 4 processes

****

avg.c with 3 processes



Text file holding R values of all pixels: r.txt. G and B values stored in g.txt and b.txt respectively.



Image used: img1.jpg

**OBSERVATION**

Each process gets a set of rows from the R, G and B matrices. Their average is calculated and written back into a text file to be read by another python program that forms the resultant image which is grayscale.

|  |  |  |
| --- | --- | --- |
| **Height x Width** | **NUMBER OF PROCESSES** | **TIME** |
| 479 x 500 | 1 | 0.602425 |
| 2 | 0.621714 |
| 3 | 0.608360 |
| 4 | 0.568415 |
| 100x1000 | 1 | 10.3 |
| 2 | 6.32 |
| 3 |  |
| 4 | 1.34 |

**CONCLUSION**

We have computed the average of R, G and B matrices and formed a grayscale image.