

PARALLEL AND DISTRIBUTED COMPUTING LAB

REPORT

NAME: S Shyam Sundaram

REG NO: 19BCE1560

PROGRAMMING ENVIRONMENT: MPI

PROBLEM: MPI

DATE: 17th November, 2021

HARDWARE CONFIGURATION:

		Intel core i5 – 1035G1 @ 1.00
CPU NAME	:	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

```

(MachineLearning) shyam@shyam-Inspiron-14-5408:~/Academics/Lab-Fall-2021/PDC/Lab12$ mpicc avg.c
(MachineLearning) shyam@shyam-Inspiron-14-5408:~/Academics/Lab-Fall-2021/PDC/Lab12$ mpirun -np 4 ./a.out
Name: S Shyam Sundaram
Reg num: 19BCE1560

3 and 119
Final is matrix written to file
Elapsed time is 0.568415
Elapsed time is 0.569588
Elapsed time is 0.573728
Elapsed time is 0.622162

```

avg.c with 4 processes

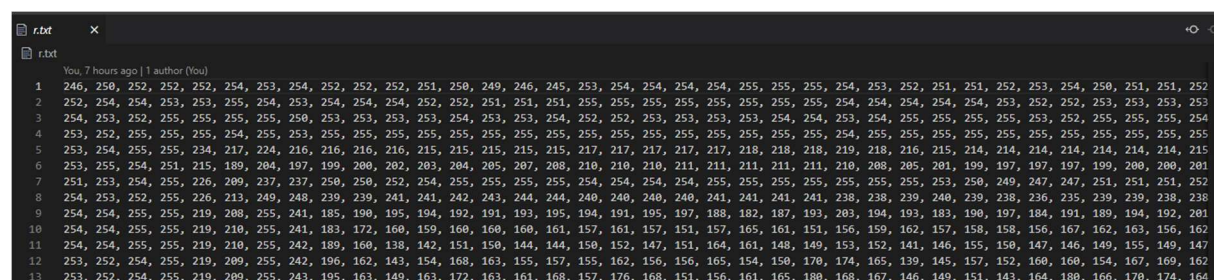
```

(MachineLearning) shyam@shyam-Inspiron-14-5408:~/Academics/Lab-Fall-2021/PDC/Lab12$ mpirun -np 3 ./a.out
Name: S Shyam Sundaram
Reg num: 19BCE1560

2 and 159
Final is matrix written to file
Elapsed time is 0.560012
Elapsed time is 0.556266
Elapsed time is 0.608360

```

avg.c with 3 processes



```

r.txt
x
You 7 hours ago | 1 author (You)
1 246, 250, 252, 252, 252, 254, 253, 254, 252, 252, 252, 251, 250, 249, 246, 245, 253, 254, 254, 254, 254, 255, 255, 255, 254, 253, 252, 251, 251, 252, 253, 254, 250, 251, 251, 252
2 252, 254, 254, 253, 253, 255, 254, 253, 254, 254, 254, 252, 252, 251, 251, 251, 255, 255, 255, 255, 255, 255, 255, 254, 254, 254, 254, 253, 252, 252, 253, 253, 253, 253
3 254, 253, 252, 255, 255, 255, 255, 250, 253, 253, 253, 253, 254, 253, 253, 254, 252, 252, 253, 253, 253, 253, 254, 254, 253, 254, 255, 255, 255, 255, 253, 252, 255, 255, 255, 254
4 253, 252, 255, 255, 255, 254, 255, 253, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 254, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255
5 253, 254, 255, 255, 234, 217, 224, 216, 216, 216, 215, 215, 215, 215, 215, 215, 217, 217, 217, 217, 218, 218, 218, 219, 218, 218, 216, 215, 214, 214, 214, 214, 214, 214, 215
6 253, 255, 254, 251, 215, 189, 204, 197, 199, 200, 202, 203, 204, 205, 207, 208, 210, 210, 210, 211, 211, 211, 211, 211, 210, 208, 205, 201, 199, 197, 197, 197, 199, 200, 200, 201
7 251, 253, 254, 255, 226, 209, 237, 237, 250, 250, 252, 254, 255, 255, 255, 255, 254, 254, 254, 254, 255, 255, 255, 255, 255, 253, 250, 249, 247, 247, 251, 251, 251, 252
8 254, 253, 252, 255, 226, 213, 249, 248, 239, 239, 241, 241, 242, 243, 244, 244, 240, 240, 240, 240, 241, 241, 241, 241, 238, 238, 239, 240, 239, 238, 236, 235, 239, 239, 238, 238
9 254, 254, 255, 255, 219, 208, 255, 241, 185, 190, 195, 194, 192, 191, 193, 195, 194, 191, 195, 197, 188, 182, 187, 193, 203, 194, 193, 183, 190, 197, 184, 191, 189, 194, 192, 201
10 254, 254, 255, 255, 219, 210, 255, 241, 183, 172, 160, 159, 160, 160, 160, 161, 157, 161, 157, 151, 157, 165, 161, 151, 156, 159, 162, 157, 158, 158, 156, 167, 162, 163, 156, 162
11 254, 254, 255, 255, 219, 210, 255, 242, 189, 160, 138, 142, 151, 150, 144, 144, 150, 152, 147, 151, 164, 161, 148, 149, 153, 152, 141, 146, 155, 150, 147, 146, 149, 155, 149, 147
12 253, 252, 254, 255, 219, 209, 255, 242, 196, 162, 143, 154, 168, 163, 155, 157, 155, 162, 156, 156, 165, 154, 150, 170, 174, 165, 139, 145, 157, 152, 160, 160, 154, 167, 169, 162
13 253, 252, 254, 255, 219, 209, 255, 243, 195, 163, 149, 163, 172, 163, 161, 168, 157, 176, 168, 151, 156, 161, 165, 180, 168, 167, 146, 149, 151, 143, 164, 180, 166, 170, 174, 164

```

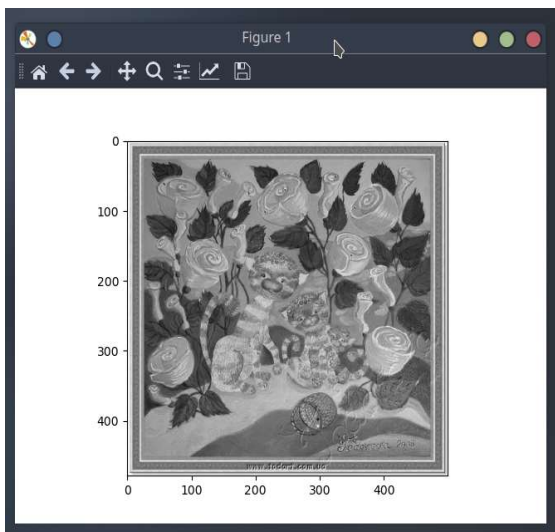
Text file holding R values of all pixels: r.txt. G and B values stored in g.txt and b.txt respectively.



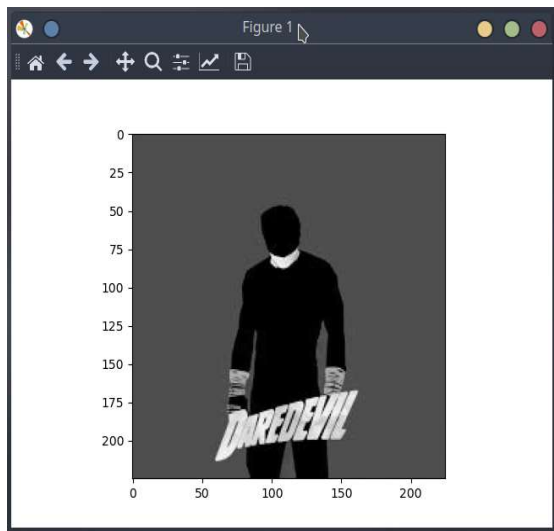
Image used: img1.jpg



Image used: images.jpg



Final grayscale of img1.jpg

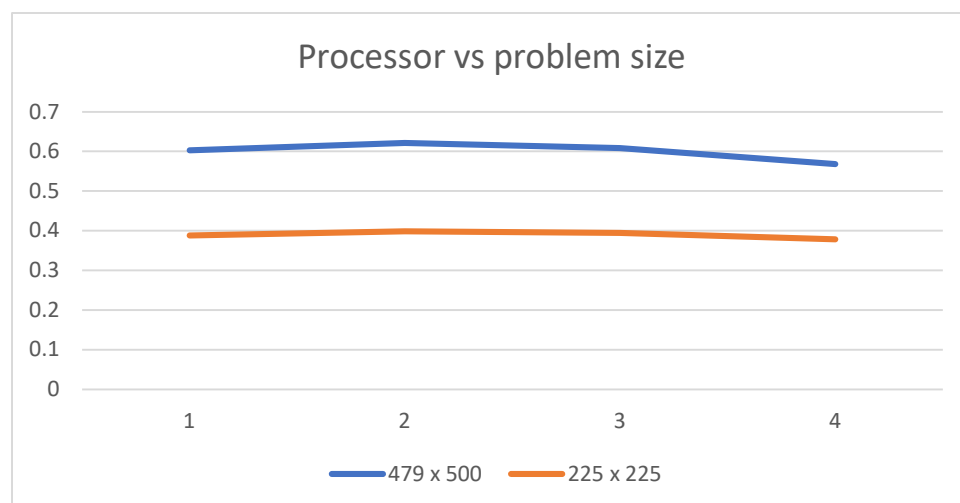


Final grayscale of images.jpg

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
225x225	1	0.387871
	2	0.398878
	3	0.394761
	4	0.378416



CONCLUSION

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