Gamma correction

of a PGM file

*using **openMP** and **openMPI***

Professors: Gianpaolo Cugola, Sam Guinea
Authors: Riccardo Polvara, Giovanni Zingaro, Francesco Zoffoli

What is a PGM file

«Portable Greyscale Map»

HEADER

```
P2
# feep.pgm
24 7
15
```

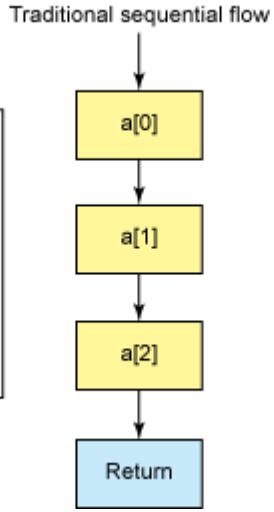
Format ID
Comment
Width and Height
Max grey value

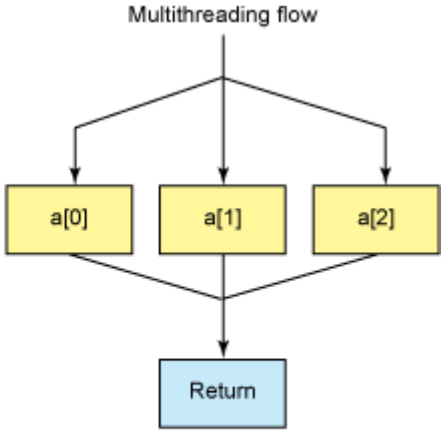
```
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  3  3  3  3  0  0  7  7  7  7  0  0 11 11 11 11  0  0 15 15 15 15  0
 0  3  0  0  0  0  0  7  0  0  0  0  0 11  0  0  0  0  0 15  0  0 15  0
 0  3  3  3  0  0  0  7  7  7  0  0  0 11 11 11  0  0  0 15 15 15 15  0
 0  3  0  0  0  0  0  7  0  0  0  0  0 11  0  0  0  0  0 15  0  0  0  0
 0  3  0  0  0  0  0  7  7  7  7  0  0 11 11 11 11  0  0 15  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
```

OpenMP

Multithreading C edample with OpenMP

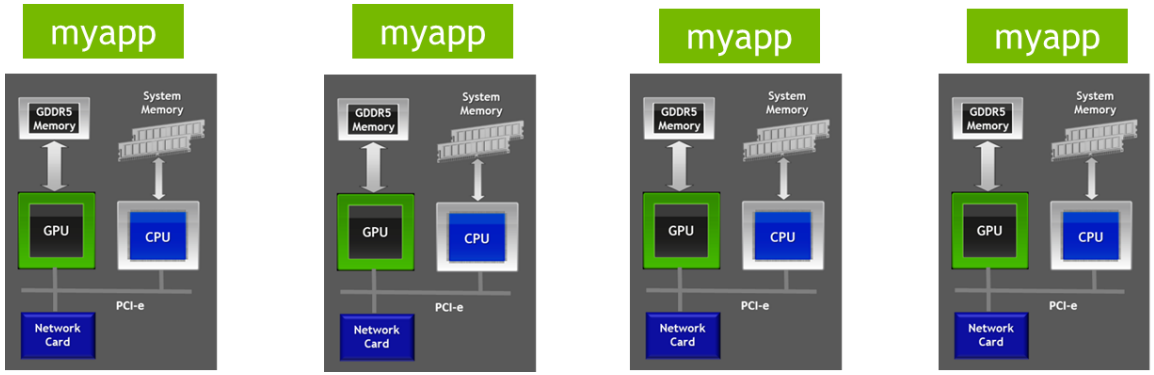
```
int main( int argc, char **argv)
(
        int i, a[3]
        #pragma omp parallel for
        for (i - o ; i < 3 ; i++) {
                a[i] = i*i;
        )
        return 0;
}
```





openMPI

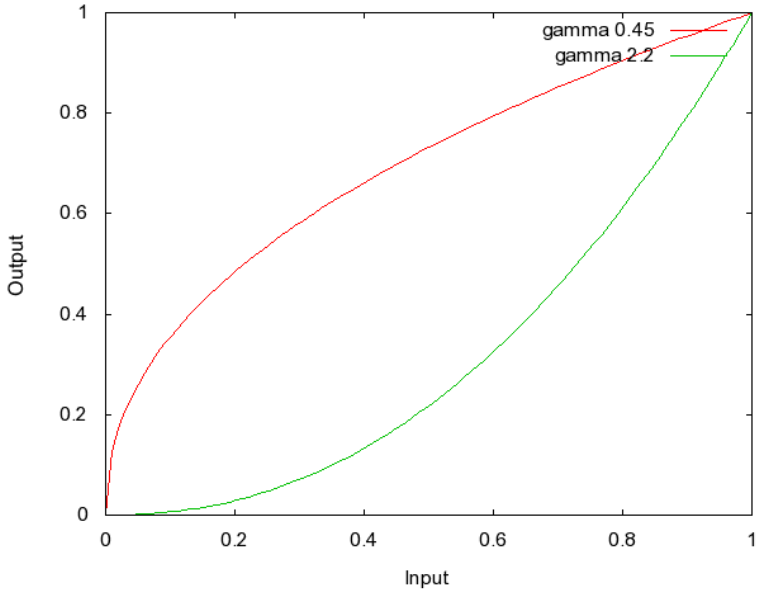
```
mpirun -np 4 ./myapp <args>
```

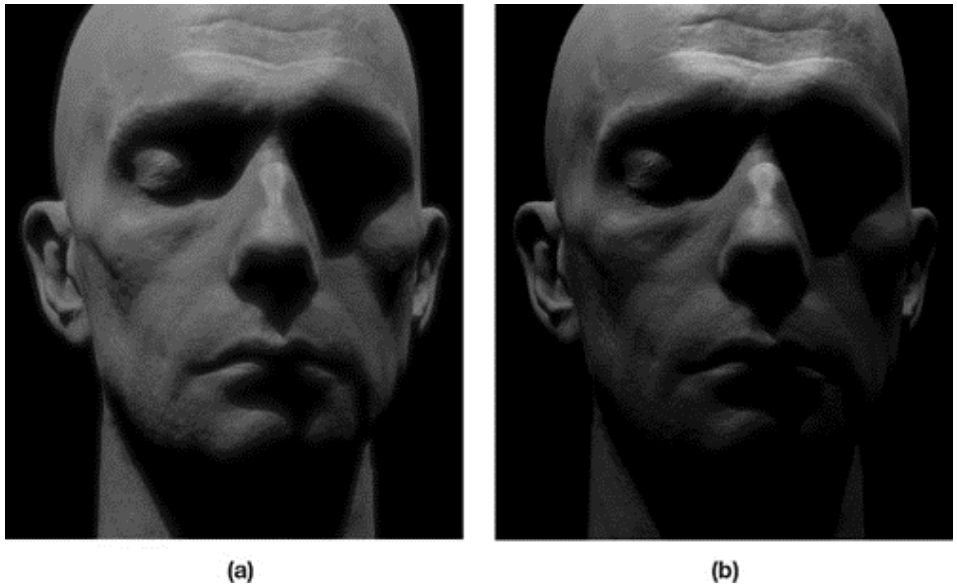


Gamma correction

$$P' = max\_value*(P/max\_value)^{g}$$

P= pixel value
G= gamma value selected
Max_value = max grey value accepetd

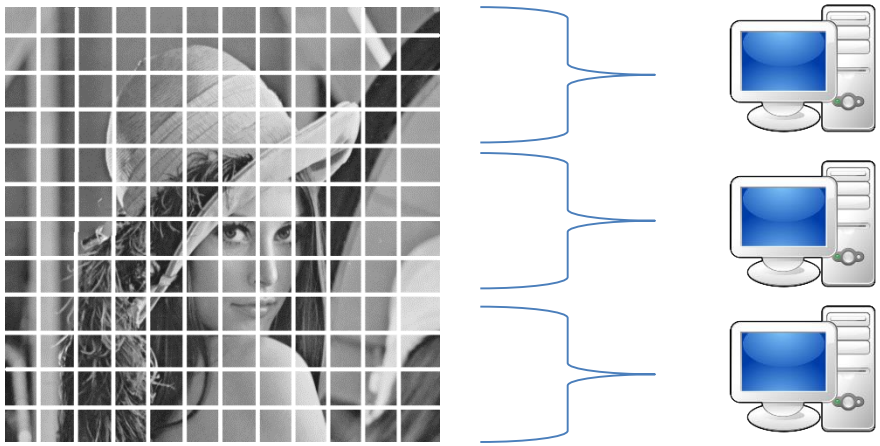




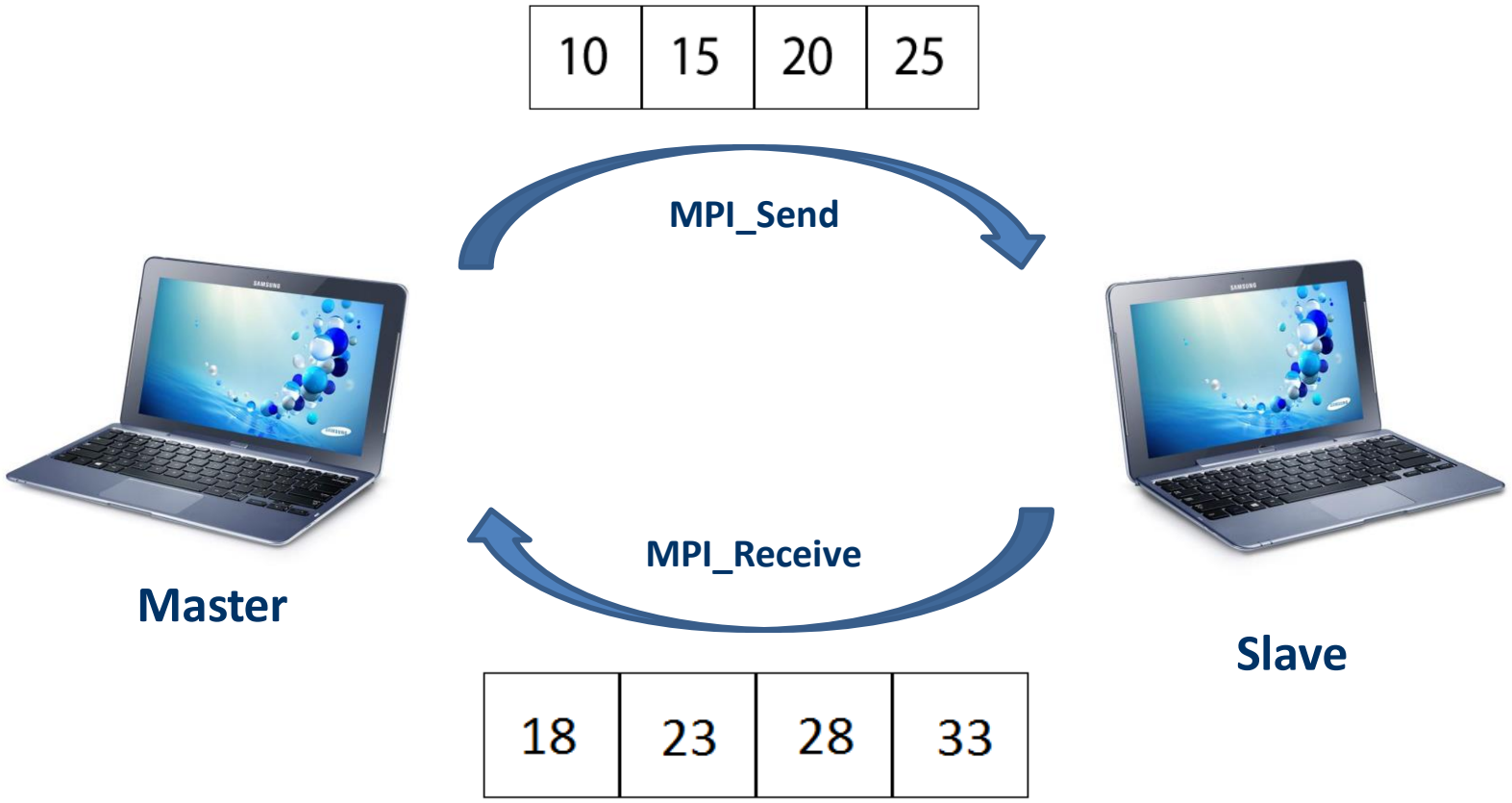
(a) (b)

Key Ideas

- The image is a vector of integer
- Divide the img data into several chunks. Then send the chunks to all the "slaves" for them to compute. The master takes care of the last chunk, which will be the biggest, since it takes care to compute to the end of the vector in order to cover for approximations in the chunk_size computation



How it works



DEMO