Homework 1 - Parallel Computations for Large-Scale Problems

Oussama Kaddami

February 2022

1 Describe the difference between a process and a processor!

A processor is a physical hardware device that has the capability of accessing the memory and computing new data values, while the process is a computational activity assigned to a processor. We can think of a program as an abstraction of the processor, and the process as a running program.

2 What is the performance of the system?

Using Amdahl's law, SpeedUp = $\frac{1}{s+\frac{1-s}{p}}$ where s is the sequential portion of the code.
Considering the number of instructions **N**, The execution rate = $\frac{N}{executiontime}$, thus: execution rate = $\frac{N}{serialTime}$ and the performance of the parallel execution is : $\frac{N}{parallelTime}$.
On the other hand, and $parallelTime = \frac{serialTime}{SpeedUp}$ so the performance of the system is :

$$SpeedUp \times executionRate = \frac{1}{0.1+\frac{0.9}{100}} \times 2.10^9 = 1.84.10^{10} flops$$

3 Is it possible for a system to have a system efficiency (η_P) of greater than 100%?

The maximum efficiency is 100%, which occurs when the computation can be divided into equal-duration parallel problems (Speedup = p) and in which case all processors are used all time throughout the execution.

4 Describe a modification to handle this situation

A simple solution is to split the array into 10 sub array and assign the computation of each to a process in order to have a balanced workload.

5 You are given some time on a new parallel computer. You run a program which parallelizes perfectly during certain phases, but which must run serially during others.

a. What fraction of the serial running time is spent in the serial section?

We have $t_p = s + \frac{1-s}{p} \times t_s$, where t_p is the parallel execution time and t_s the sequential execution time, which leads to :

$$s = \frac{\frac{p.t_p}{t_s} - 1}{p - 1} = 0.25$$

b. Determine the parallel speedup.

The amdahl's law gives: SpeedUp $= \frac{1}{s+\frac{1-s}{p}} = 2.9$. We note that the Gustafson's law provides an other expression but the amdahl's law is the relevant one in this case, since we measure the speedup for a program and for a fixed number of processors.

c. What is the speedup on 15 processors?

We consider that we have three synchronization points and that the three part occupy respectively, 20%, 20% and 60% of the execution time. We are not provided the respective speedups, so we consider that the respective speedUp equals the minimal number of processors for the optimal execution time.

The speedUp is : $0.2 \times 5 + 0.2 \times 10 + 0.6 \times 15 = 13.5$

6 Parallel implementation of the Mandelbrot algorithm

a. Implement the Mandelbrot program using MPI

Please find the source code attached in the archive.

b. Reproduce the figure from the lecture notes

The figure is plotted using the library **Pillow** of python. The corresponding code is also attached in the archive.

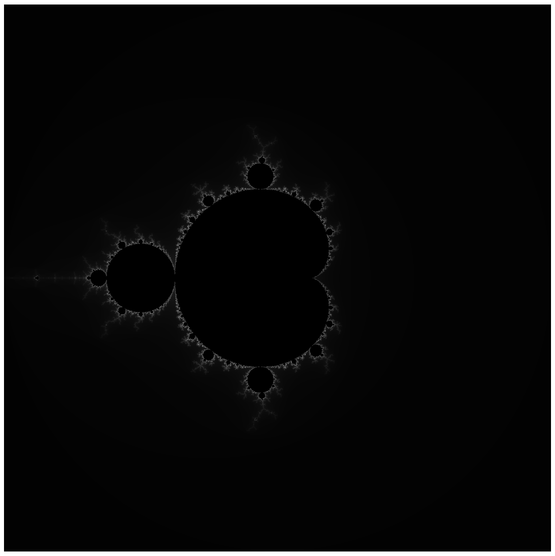

Figure 1: Mandelbrot set

c. Magnify some interesting parts of the figure. Do this by computing only parts of the complete picture using higher resolutions

We can try to modify different parameters in order to investigate different aspects: we can try a higher resolution, or modify the boundary. Modifying the boundary **b** is equivalent to a zoom in (decreasing b) and out(increasing b). Increasing the resolution permits to have more visible details in the graphic.

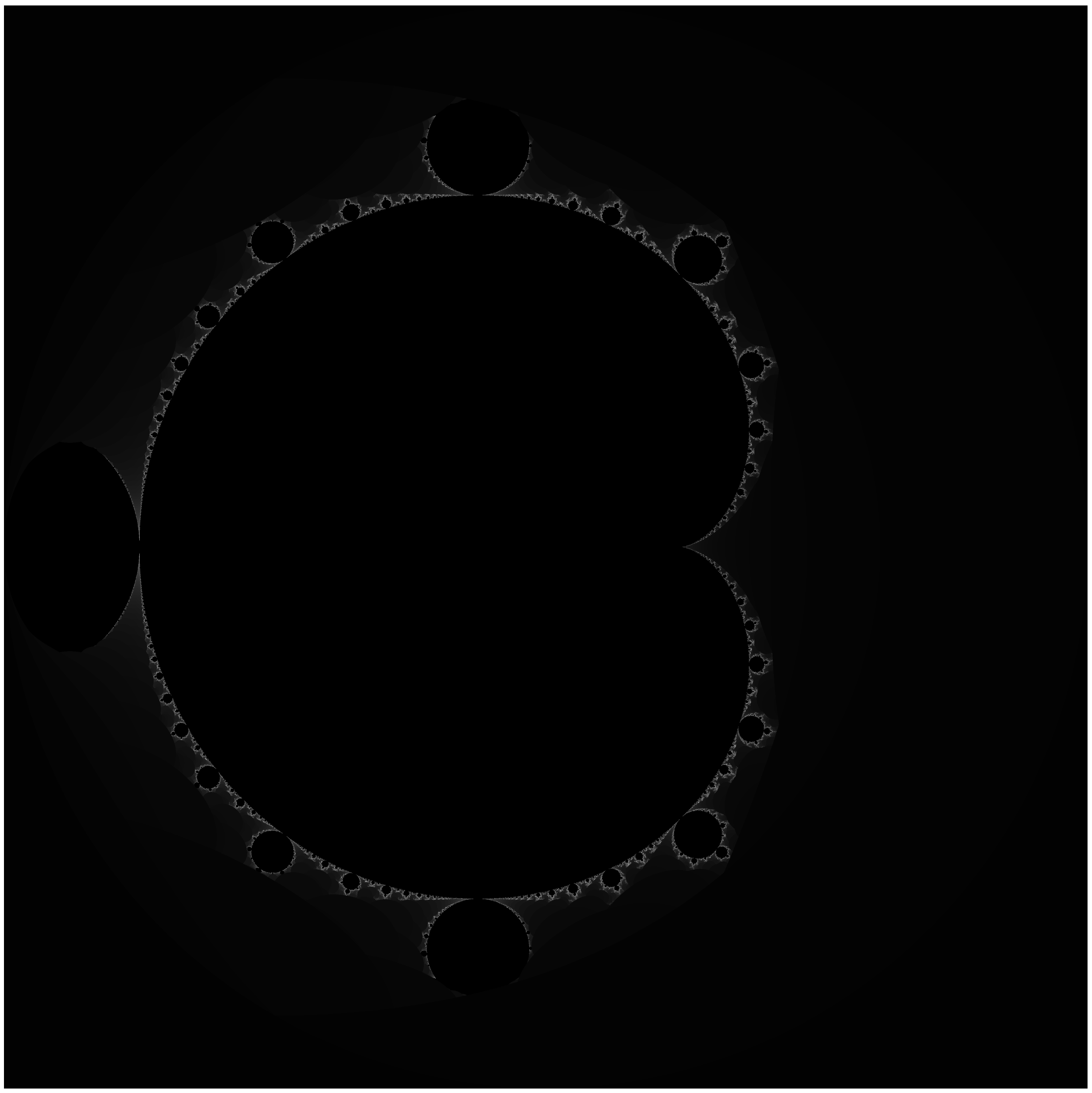

Figure 2: Mandelbrot set with b = 1 and with 4096*4096 resolution

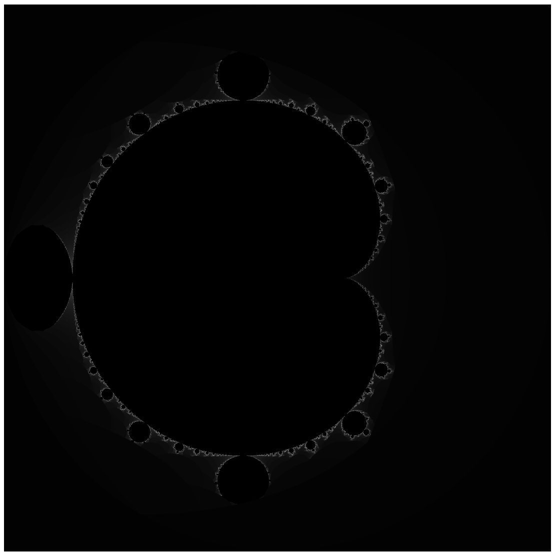

Figure 3: Mandelbrot set with b = 1 and with 2048*2048 resolution