The Mandelbrot set
Hybrid MPI/OpenMP implementation

Alessandro Della Siega

May 2024

1 Introduction

The goal of this assignment is to compute and visulize the Mandelbrot set. The Mandelbrot set properties make this problem embarrassingly parallel, and it is a perfect candidate for a hybrid MPI/OpenMP implementation.

The computation will be performed on the ORFEO computational resource. In particular, we will take advantage of 2 nodes of the EPYC partition which consists of 8 nodes equipped with two AMD EPYC 7H12 CPUs.

The analysis of the computation is done in the context of a hybrid MPI/OpenMP implementation. We will evaluate the strong scaling and the weak scaling as follows:

- MPI scaling: we will consider a single OpenMP thread per MPI process and we will increase the number of MPI processes;
- OpenMP scaling: we will consider a single MPI process and we will increase the number of OMP threads.

2 Implementation

The C implementation consists of three main files:

- `main.c`, it contains the main function in which MPI is initialized, the amount of work is distributed among the MPI processes, and the computation is performed. In the end, the computation is gathered and the image is saved. The main function also takes care of the timing of the computation;

- `mandelbrot.c`, in this file we define the function that computes the Mandelbrot set. In this function, we iterate over the pixels of the image and we compute the value of the Mandelbrot set for each pixel. The iteration is within a OpenMP parallel region.

- `image.c`, in this file we define the function that saves the image in the PGM format;

- `timing_utils.c`, in this file we define the functions that are used to measure the time of the computation.