EECS201000
Introduction to Programming Laboratory

Homework 2: Mandelbrot Set

Due: July 17, 2017, 8AM

1 GOALSCHE

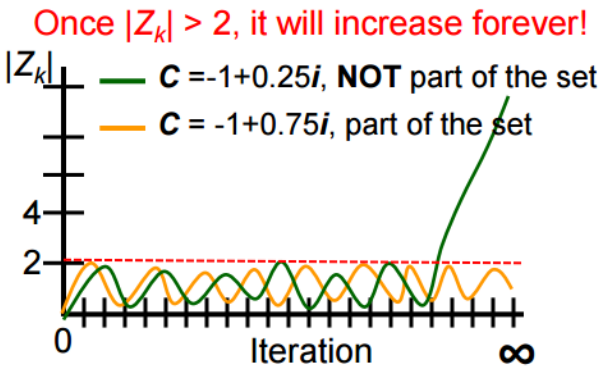
This assignment aims to get you familiar with **hybrid parallelism** (**MPI + OpenMP**), and **load balancing** techniques by implementing the Mandelbrot Set problem.

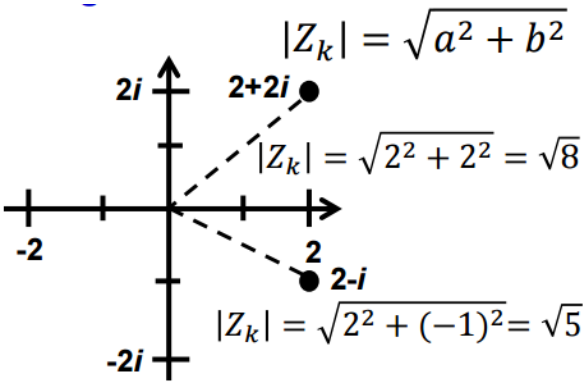
2 PROBLEM DESCRIPTION

The Mandelbrot Set is a set of complex numbers that are quasi-stable when computed by iterating the function:

$$Z_0 = C, Z_{k+1} = Z_k^2 + C$$

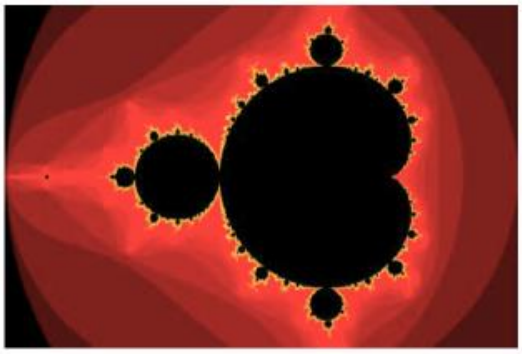
- C is some complex number: $C = a + bi$
- Z_{k+1} is the $(k+1)_{th}$ iteration of the complex number
- if $|Z_k| \leq 2$ for any k, C belongs to Mandelbrot Set





What exact is Mandelbrot Set?

- It is fractal: An object that display self-similarity at various scale; Magnifying a fractal reveals small-scale details similar to the larger-scale characteristics
- After plotting the Mandelbrot Set determined by thousands of iterations:



For more information, please refer to lecture notes.

3 INPUT / OUTPUT FORMAT

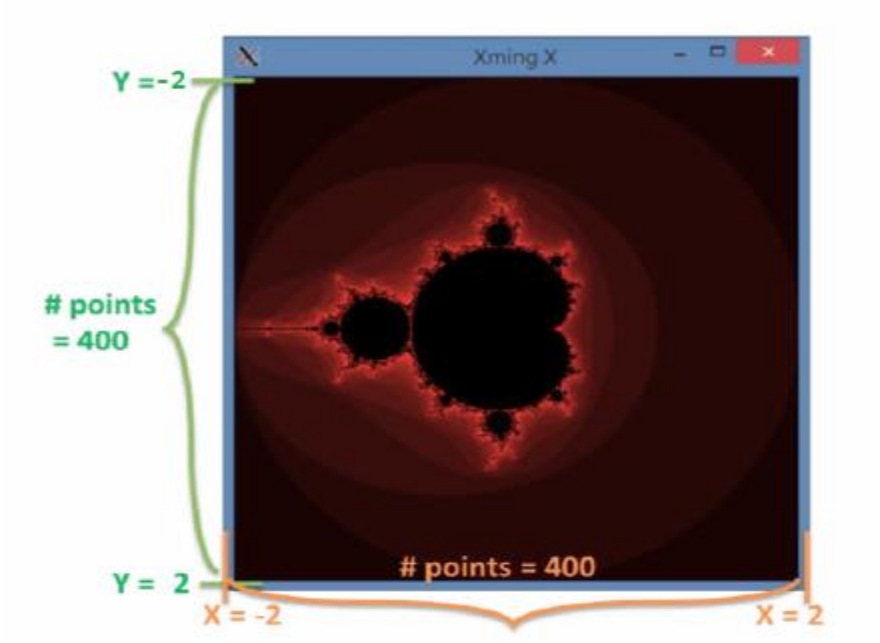
1. The value of the points is between 0-255
2. Your program accepts 8 input parameters:

./executable ${1} ${2} ${3} ${4} ${5} ${6} ${7} ${8}

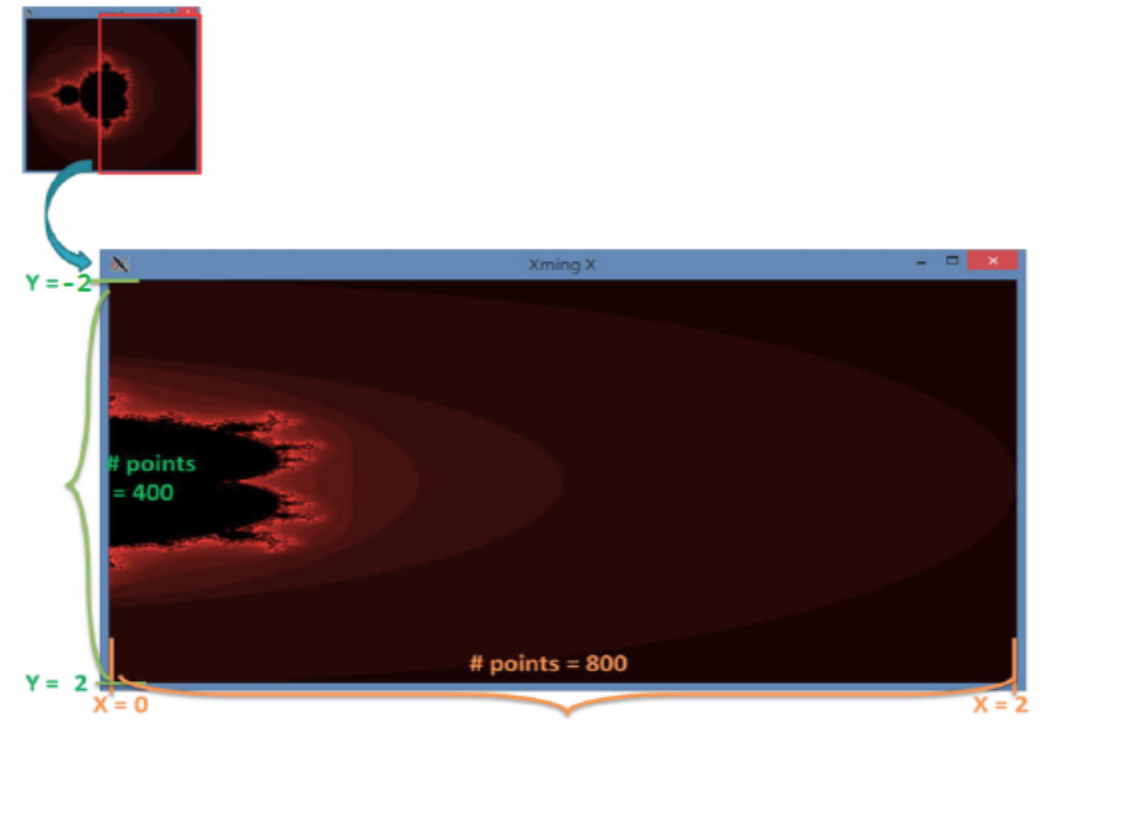
- ${1}: number of threads per process [int, between 1 ~ 12]
- ${2}: left range of real-axis [double, between -10 ~ 10]
- ${3}: right range of real-axis [double, between -10 ~ 10]
- ${4}: lower range of imag-axis [double, between -10 ~ 10]
- ${5}: upper range of imag-axis [double, between -10 ~ 10]
- ${6}: number of points in x-axis [int, between 200 ~ 4000]
- ${7}: number of points in y-axis [int, between 200 ~ 4000]
- ${8}: output filename

3. Output:
   - The output file lists the value of each point in the plot from right to left and top to bottom.
   - You may visualize the output file through X-window by running the script located at **/home/ipl2017/shared/hw2** with your output filename as the only input argument.

Example 1: **mpirun -n 2 ./HomeWork2 4 -2 2 -2 2 400 400 output_file**



Example 2: **mpirun -n 2 ./HomeWork2 4 0 2 -2 2 800 400 output_file**



4 OPTIMIZATION HINTS

- **Use dynamic load balancing algorithm**
- Minimize communication overhead

5 GRADING

1. **Correctness** (60%)
   - During the demo time, TA will use a lot of different combinations of parameters to test your program, and check whether the data values in your output file are correct.
2. **Performance** (30%)
   - Performance is measured by the program execution time when X-window is disabled.
   - Points are giving according to the performance ranking of your program among all the students.
3. **Demo** (10%)
   - Each student is given 5 minutes to explain your implementation followed by some questions from TA.
   - Not debugging or code modification is allowed during the demo.
   - Points are given according to your understanding and explanation of your code, and your answers of the TA questions.
4. **Late Policy**
   - 80% within 3 days, 60% afterwards.

6 SUBMISSION

- Please upload the following files to **homework/HW2** directory on **apollo31** under your home directory before **7/17(Mon) 8:00AM**: **(The folder will be locked after deadline)**

  i、 **HW2_{student-ID}.c**

  ii、 **Makefile**

  Make sure your compile script can execute correctly and your code has no compile error in the **uploaded folder**

7 REMINDER

1. **We provide a sequential version of Mandelbrot Set under /home/ipl2017/shared/hw2 for your reference.**
2. Compilation:

   mpicc HW2_{student-ID}.c -o HW2_{student-ID} -fopenmp

   We provide a sample makefile under /home/ipl2017/shared/hw2 for your reference, please change ID of MS_s105062553.c in makefile to your ID.
3. Since we have limited resources, please **start your work ASAP**. Do not leave it until the last day!
4. **Do NOT try to abuse the computing nodes by ssh to them directly**. If we ever find you doing that, you will get 0 point for the homework!
5. **0 will be given to cheater** (even copying code from the Internet), but discussion on code is encouraged.
6. Asking questions through iLMS is welcomed!