Daniel Eisenberg
CSc 422
Distributed project report

For this project, I created a distributed fractal generation C application. It primarily computes the Mandelbrot set on user specified rectangles in the complex plane.

The application uses a multiple instruction, multiple data model and employs the master/slave paradigm. One copy of one program, fractal.c, creates an X11 server and requests values for each of the pixels of an associated window. Arbitrarily many copies of another program, fractalcalc.c, calculate set inclusion and return these values. The first program then draws the points as appropriate. Communication between the master and each slave is facilitated via OpenMPI.

After each program establishes a connection to the MPI comm world, the master program sends the dimensions of its X11 window to each slave. It then performs a non-blocking send of dummy values to each slave, followed by a non-blocking receive for each slave. Following this, for each pixel in the window, the master program waits for the first pending receive request to complete, handles that receive, and performs a non-blocking send and receive for the pixel. In this way each pixel is populated. Once it has finished, the master sends a sentinal value to each slave and disconnects from the MPI comm world.

Finally, the master writes the generated image to a pixmap file and hangs until the user prompts an exit by pressing a key in the terminal.

Each slave receives an identical set of arguments from which it determines the rectangle of the complex plane to translate points to and how many iterations should define set inclusion of a point. Each slave connects to the MPI comm world and blocks until it receives the screen width and height values. Once these values are received, a slave enters the main loop.

In the main loop, the slave waits to receive a pixel value (if it receives the sentinal value, it will break the loop, disconnect from the MPI comm world, and exit), translates this to the appropriate point in the complex plane, and performs a sequence estimate for the point until the point is determined not in the set or the iteration count is reached. It then returns the pertinent information to the master program via a blocking send.

The sequence calculation proceeds as follows. Given a point c in the complex plane, define a function G_c by $G_c(z) = z^2 + c$. We then examine the sequence $(0, G_c(0), G_c(G_c(0)), \ldots)$. We define c to be a point in the Mandelbrot set M if and only if this sequence converges to a finite limit. It can be shown that $c \in M$ if and only if the sequence converges to a point within the circle of radius 2 centered at the origin. Thus to calculate inclusion of c, we calculate each term of the sequence for c until its magnitude is at least 2 or we reach our maximum iteration parameter.

Initially, I intended to use an escape-time algorithm to color points exterior to the set. However, my attempts at implementation of this feature caused excessive sluggishness in the X11 server, so I decided that feature may not be appropriate for a program intended to forward X11 via ssh.

The application supports user-defined rectangles, so arbitrary zooms on the Mandelbrot set may be calculated. The program also supports rendering of the tricorn fractal set, also known as the Mandelbar set.