CS 5500 Homework 6

Sally Devitry (A01980316)

Approach

In this assignment, we were tasked with writing a parallel program that creates a 512x512-pixel mandelbrot image in ppm format and running it on multiple nodes in the cluster.

My program uses a leader/worker system to divide the work up. I have process 0 send different rows of pixels to the worker processes. The worker processes then calculate the number of iterations for each pixel on the row it was given. It then sends the row calculations back to process 0 and once process 0 has received all of the rows back, it writes all of the data to the file.

Implementation

For process 0 to assign the rows evenly to the worker processes, the following code is used:

```
dest = (j % (size-1))+1;
```

When process 0 has received all of the row calculations back, it sends a poison message to all of the worker processes to stop execution. It also writes to the ppm file to create the mandelbrot image.

A while (true) loop is used to have the worker processes continually accept new rows to calculate. The loop breaks when a worker process receives a poison message from the leader process.

My parallel mandelbrot with 4 processes generally took 800-900ms. With 8 processes it generally took 750-850ms. The serial version that Professor Watson wrote generally took about 900-1000ms. My parallel version didn't offer very much speed up, but I believe this is due to the fact that a large chunk of the execution time is going to be overhead in either case.

Run commands:

```
mpirun -np 4 --oversubscribe a.out
mpirun -np 8 --oversubscribe a.out
```

Some example output:

For 4 processes - (`mpirun -np 4 --oversubscribe a.out`)

```
working....
Execution took 867 ms.
```

For 8 processes - (`mpirun -np 8 --oversubscribe a.out`)

```
working...
Execution took 812 ms.
```

Concluding Remarks

Overall, this was an interesting and fun assignment. I enjoyed messing with the colors of the mandelbrot and trying different things to get more speed-up. The tough part for me was dealing with arrays in C++. I got several seg faults that took me a while to fix.