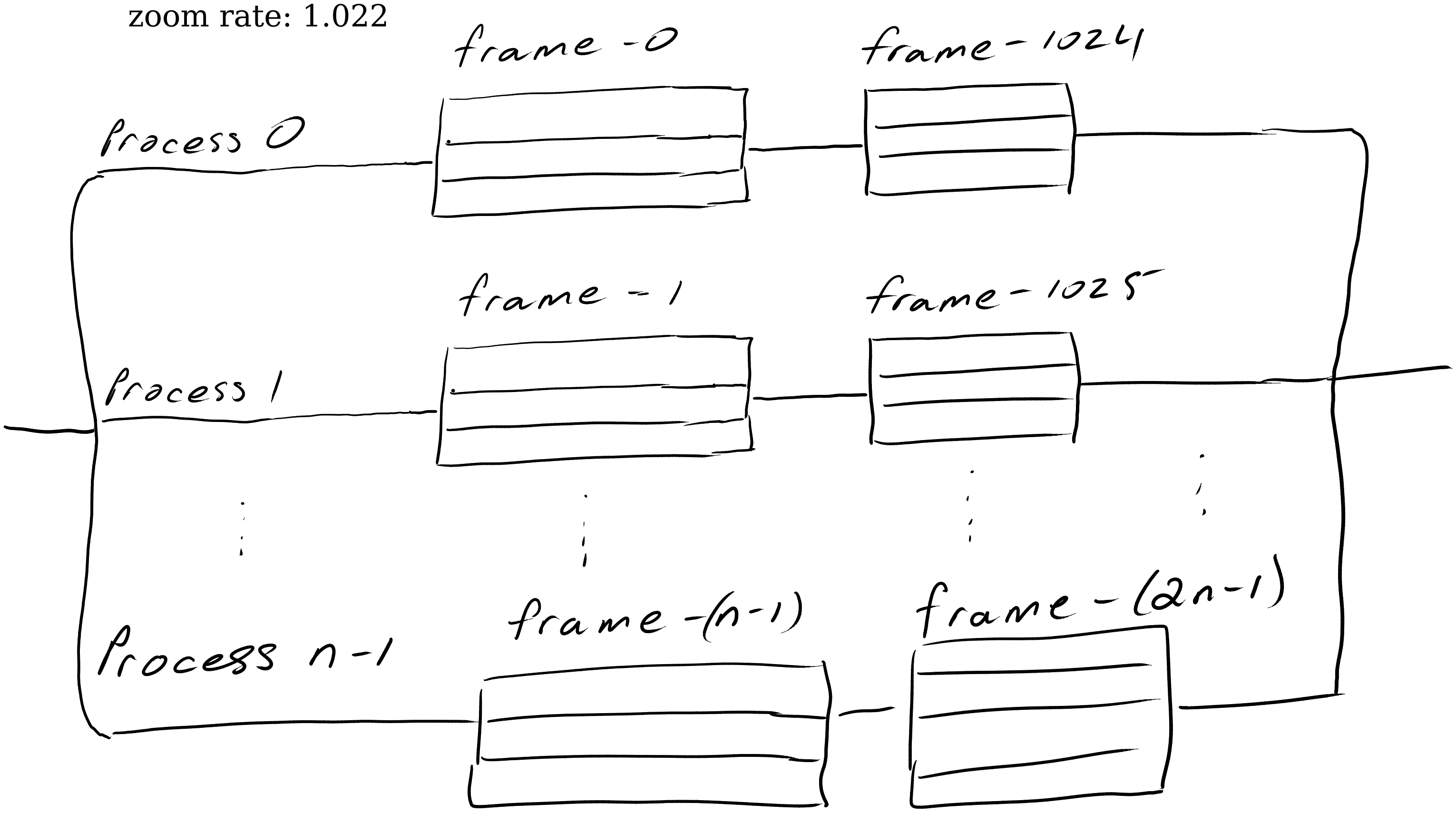
Table of Contents

Description of the algorithm and implementation

The goal of this Project was to produce frames for a deep zoom into a point of the Mandelbrot set. The algorithm from assignment 3 was reworked in the following way:

- instead of having all the processors working on the same image simultaneously, each processor works on one image, and then saves the image to a bmp. This removes the overhead of sending and receiving data, speeding up the process.
- each process is forked into 4 threads, so four threads work on one image. This introduces some overhead, but utilizes Blue Gene Q better, and overall speeds up the process.
- for this project we are only concerned with the mandelbrot set, therefore the parameters for Julia sets (c and the flag) were completely removed from the code.

Frame size: 1280 x 720
maxiter: 10,000
zoom rate: 1.022



This approach worked well, until zoom levels of about 6.1e58, where the calculations started taking over 60 minutes. The last frame to be calculated in under 60 minutes on Blue Gene / Q was:

frame-06220.bmp: radius 1.642208e-59, zoom level 6.089363e+58

The requirement for the Project was a 3 minute video at 30 fps:
3 minutes = 180 seconds
180 seconds * 30 fps = 5400 frames.

This mark was reached with frame-5400.bmp:
frame-05400.bmp: radius 9.229171e-52, zoom level 1.083521e+51

Frames 0-3700 (radius 1 - 1.07e-35) were calculated using 128 bits of precision. Frames 3701 - 5400 (radius 1.052530e-35 - 9.229171e-52) were calculated using 256 bits of precision.

I did write a double precision version of the code, but did not use it.

Timing Data:

Note: maxiter of 10000 was used

128 bits of precision:
==> frame.00000-03700.out <== **computed on Blue Gene / Q**
Total compute time for 3701 images: 1346718.20 seconds (22445.30 minutes)
Total parallel time for 3701 images: 1880.37 seconds (31.34 minutes)

256 bits of precision:
==> frame.03700-03800.out <== **computed on VELOCITY (8 logical cores)**
Total compute time for 101 images: 61986.88 seconds (1033.11 minutes)
Total parallel time for 101 images: 7947.16 seconds (132.45 minutes)

==> frame.03801-04824.out <== **computed on Blue Gene / Q**
Total compute time for 1024 images: 1857175.62 seconds (30952.93 minutes)
Total parallel time for 1024 images: 2087.31 seconds (34.79 minutes)

==> frame.04825-05848.out <== **computed on Blue Gene / Q**
Total compute time for 1024 images: 2467067.35 seconds (41117.79 minutes)
Total parallel time for 1024 images: 3290.09 seconds (54.83 minutes)

The total time taken to compute the first 5848 frames:

- 5748 computed on Blue Gene / Q in 31.34+ 34.79 + 54.83 = 120.96 minutes.
- 100 frames computed on a 4 core machine with hyperthreading (8 logical cores) in 132.45 minutes.

In any event, the entire first 5400 could be done on Blue Gene in under 2 hours, especially if the double precision code was used for the first 1024 frames. Since one processor is working on one frame at a time, whether Blue Gene was processing one frame or 1024 frames of images with a radius from e-46 to e-52, it would take not much less than 60 minutes (see frame_data directory for detailed breakdown if interested). The actual time taken for each image was not recorded, only the parallel time and the total time for all the images.

From here on, I decided I wanted to put my video to music. The audio file I chose was 4 minutes and 25 seconds long. This translated into 7950 total frames needed. So I continued making frames.

As mentioned previously, I was only able to make up to frame-06220.bmp using this alogorithm on Blue Gene / Q, as beyond this point each frame takes over 60 minutes. Frames 6221 - 7921 were created on mpihost, and my 4 core hyperthreaded (8 logical) machine.

In order to be able to run the code on Blue Gene / Q again, I decided to revert back to the algorithm in assignment 3, where all the processors work on the same image at the same time, and process 0 saves the image.

By the time all was said and done, I ended up with 12,735 images, enough for a 48 fps video of 4 minutes and 25 seconds.
(see Project/video/tiffany_antopolski_final_project.webm).

The timing data for the remaining frames follows. Computed on Blue Gene / Q unless otherwise specified.

```
==> frame.05849-06220.out <== //6219 had improper image header according
to "convert"
Total compute time for 372 images:  1282424.71 seconds (21373.75 minutes)
Total parallel time for 372 images: 3590.84 seconds (59.85 minutes)

==> frame.06219.out <== computed on VELOCITY (8 logical cores)
Total compute time for 1 images:  965.60 seconds (16.09 minutes)
Total parallel time for 1 images: 965.60 seconds (16.09 minutes)

==> frame.06221-06872.out <== computed on VELOCITY (8 logical cores)
Total compute time for 652 images:  1100833.44 seconds (18347.22 minutes)
Total parallel time for 652 images: 140260.15 seconds (2337.67 minutes)

==> frame.06783.out <==
Total compute time for 1 images:  6632.89 seconds (110.55 minutes)
Total parallel time for 1 images: 9.30 seconds (0.15 minutes)

==> frame.06874-07896.out <== computed on MPIHOST
Total compute time for 1023 images:  1449045.78 seconds (24150.76 minutes)
Total parallel time for 1023 images: 14362.67 seconds (239.38 minutes)

==> frame.07897.out <== computed on VELOCITY (8 logical cores)
Total compute time for 1 images:  2060.36 seconds (34.34 minutes)
Total parallel time for 1 images: 2060.36 seconds (34.34 minutes)

==> frame.07898-07920.out <== computed on VELOCITY (8 logical cores)
Total compute time for 23 images:  50855.58 seconds (847.59 minutes)
Total parallel time for 23 images: 6686.95 seconds (111.45 minutes)

512 bit of precision:
==> frame.07921-07928.out <==  computed on MPIHOST
Total compute time for 8 images:  98194.73 seconds (1636.58 minutes)
Total parallel time for 8 images: 12345.96 seconds (205.77 minutes)
```

```
==> frame.07929-07935.out <==
Total compute time for 6 images:  55661.80 seconds (927.70 minutes)
Total parallel time for 6 images: 77.37 seconds (1.29 minutes)

==> frame.07936-08000.out <==
Total compute time for 15 images:  140931.97 seconds (2348.87 minutes)
Total parallel time for 15 images: 195.80 seconds (3.26 minutes)

==> frame.08001-08200.out <==
Total compute time for 200 images:  1900341.33 seconds (31672.36 minutes)
Total parallel time for 200 images: 2639.89 seconds (44.00 minutes)

==> frame.08201-08350.out <==
Total compute time for 150 images:  1432028.49 seconds (23867.14 minutes)
Total parallel time for 150 images: 1989.34 seconds (33.16 minutes)

==> frame.08351-08500.out <==
Total compute time for 150 images:  1446105.02 seconds (24101.75 minutes)
Total parallel time for 150 images: 2008.90 seconds (33.48 minutes)

==> frame.08501-08700.out <==
Total compute time for 200 images:  1964376.08 seconds (32739.60 minutes)
Total parallel time for 200 images: 2728.83 seconds (45.48 minutes)

==> frame.08701-08900.out <==
Total compute time for 200 images:  2023806.86 seconds (33730.11 minutes)
Total parallel time for 200 images: 2811.37 seconds (46.86 minutes)

==> frame.08901-09050.out <==
Total compute time for 150 images:  1611510.66 seconds (26858.51 minutes)
Total parallel time for 150 images: 2238.61 seconds (37.31 minutes)

==> frame.09051-09250.out <==
Total compute time for 200 images:  2287838.63 seconds (38130.64 minutes)
Total parallel time for 200 images: 3178.10 seconds (52.97 minutes)

==> frame.09251-09440.out <==
Total compute time for 190 images:  2187189.54 seconds (36453.16 minutes)
Total parallel time for 190 images: 3038.30 seconds (50.64 minutes)

==> frame.09441-09635.out <==
Total compute time for 195 images:  2259391.10 seconds (37656.52 minutes)
Total parallel time for 195 images: 3138.59 seconds (52.31 minutes)
```

```
==> frame.09636-09830.out <==
Total compute time for 195 images:  2305497.75 seconds (38424.96 minutes)
Total parallel time for 195 images: 3202.64 seconds (53.38 minutes)

==> frame.09831-10025.out <==
Total compute time for 195 images:  2351667.48 seconds (39194.46 minutes)
Total parallel time for 195 images: 3266.77 seconds (54.45 minutes)

==> frame.10026-10220.out <==
Total compute time for 195 images:  2517640.40 seconds (41960.67 minutes)
Total parallel time for 195 images: 3497.27 seconds (58.29 minutes)

==> frame.10221-10411.out <==
//initially set beyond frame 10411, but time out at 10411 at 60 minutes.

==> frame.10412-10565.out <==
Total compute time for 154 images:  2117327.53 seconds (35288.79 minutes)
Total parallel time for 154 images: 2941.17 seconds (49.02 minutes)

==> frame.10566-10750.out <==
Total compute time for 185 images:  2581912.47 seconds (43031.87 minutes)
Total parallel time for 185 images: 3586.49 seconds (59.77 minutes)

==> frame.10751-10924.out <==
Total compute time for 174 images:  2491729.95 seconds (41528.83 minutes)
Total parallel time for 174 images: 3461.21 seconds (57.69 minutes)

==> frame.10925-11101.out <==
//initially set beyond frame 11101, but time out at 11101 at 60 minutes.

==> frame.11102-11250.out <==
Total compute time for 149 images:  2184132.68 seconds (36402.21 minutes)
Total parallel time for 149 images: 3033.94 seconds (50.57 minutes)

==> frame.111250-11400.out <==
Total compute time for 150 images:  2244946.68 seconds (37415.78 minutes)
Total parallel time for 150 images: 3118.41 seconds (51.97 minutes)

==> frame.11401-11550.out <==
Total compute time for 150 images:  2369793.21 seconds (39496.55 minutes)
Total parallel time for 150 images: 3291.83 seconds (54.86 minutes)

==> frame.11551-11700.out <==
Total compute time for 150 images:  2483813.02 seconds (41396.88 minutes)
Total parallel time for 150 images: 3450.20 seconds (57.50 minutes)

==> frame.11701-11712.out <==
timed out at frame 11712 at 5 minutes

==> frame.11713-11850.out <==
Total compute time for 138 images:  2276327.91 seconds (37938.80 minutes)
Total parallel time for 138 images: 3161.97 seconds (52.70 minutes)

==> frame.11851-11895.out <==
Total compute time for 45 images:  742617.98 seconds (12376.97 minutes)
Total parallel time for 45 images: 1031.55 seconds (17.19 minutes)

==> frame.11896-12024.out <==
Total compute time for 129 images:  2134619.54 seconds (35576.99 minutes)
Total parallel time for 129 images: 2965.12 seconds (49.42 minutes)

==> frame.12024-12124.out <==
Total compute time for 101 images:  1676407.72 seconds (27940.13 minutes)
Total parallel time for 101 images: 2328.65 seconds (38.81 minutes)

==> frame.12125-12225.out <==
Total compute time for 101 images:  1674726.89 seconds (27912.11 minutes)
Total parallel time for 101 images: 2326.31 seconds (38.77 minutes)

==> frame.12226-12335.out <==
The .out file stopped at 12237.  I don't know what happened.  So
I am missing data for frames 12238 - 12335... and time taken for
12226 - 12335, but can estimate.

==> frame.12336-12435.out <==
Total compute time for 100 images:  1670706.38 seconds (27845.11 minutes)
Total parallel time for 100 images: 2320.73 seconds (38.68 minutes)

==> frame.12436-12535.out <==
Total compute time for 100 images:  1668071.86 seconds (27801.20 minutes)
Total parallel time for 100 images: 2317.10 seconds (38.62 minutes)

==> frame.12536-12635.out <==
Total compute time for 100 images:  1669916.31 seconds (27831.94 minutes)
Total parallel time for 100 images: 2319.64 seconds (38.66 minutes)
```

```
==> frame.12636-12735.out <==
Total compute time for 100 images:  1674144.04 seconds (27902.40 minutes)
Total parallel time for 100 images: 2325.50 seconds (38.76 minutes)
```

Final Region

frame-12735.bmp: radius 4.393817e-121, zoom level 2.275925e+120

Final point: (Credit: youtube user/ToobMug)
-1.999992058192687313022543729145102991847847126539783513462412398802843930231508477489212755944776005625280804144656355319

-1.450204033707124987456880881808950903622191144103813894555735824841993065946725590960484042851951351629540869648130526467e-12

Summary of Timing Stats

10919 frames computed on Blue Gene/Q of SciNet using 1024 cores in 27.6 hours.

1023 frames computed on mpihost using 128 cores in 4 hours, and an additional 8 frames using 8 cores in 3.4 hours.

778 frames computed on a quad core machine with hyperthreading, using 8 logical cores in 43.9 hours.

Total frames: 12736 total frames
Video: 48 fps

Post Processing

The frames were stitched together using Gstreamer and Cairo, with overlays created by Gimp.

Appendix A : Input File

The required input file is in the form:

real_part_of_point
imag_part_of_point
x_start_radius
y_start_radius
first_frame_number to compute
last_frame_number to compute
width
height
maxiters
zoom_rate
filename (no extension)

File used:

```
-1.999992058192687313022543729145102991847847126539783513462412398
80284393023150847748921275594477600562528080414465635531948058891
68776310285490614450736429953754788660243332023729585214640464426
39161141787629439733589289724815890177763835096121061675026913413
309691231725011972076
-0.000000000001450204033707124987456880881808950903622191144103813
89455573582484199306594672559096048404285195135162954086964813052
64673019670684917184921565878386733871135726145545458134158551726
50467322561748969106686059449116724947320308488642086833924759392
799191924373703620251
1
0.5625
0
7950
1280
720
10000
1.022
frame
```

Appendix B : Source Code

There are 3 different versions of the code:

1. alg1_dbl: This is a double precision version. The algorithm is as outlined on page 2 of this report.

2. alg1_gmp: The gmp version of the algorithm outline on page 2.

3. alg2_gmp: Similar to assingment 3. The code for Julia sets was removed (flag and parameter c). Additionally, the code in main is modified such that many images are produced.