An Investigation of Chaos Theory Using Supercomputer Techniques

Bryan Ward

Background

- The theory of non-linear functions, such that small differences in the input of the function can result in large and unpredictable differences in the output.
- Seen all over the world:
  - Weather
  - Stock Market
  - Physics

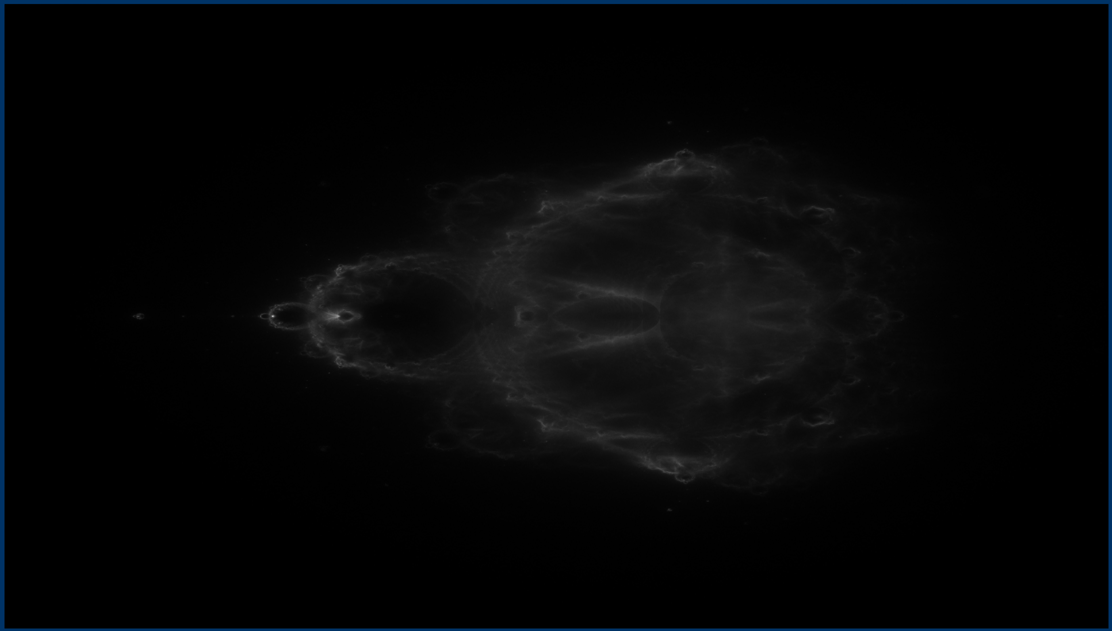
Purpose

1) To investigate and learn about chaos and fractals

2) To learn about High performance computing algorithms

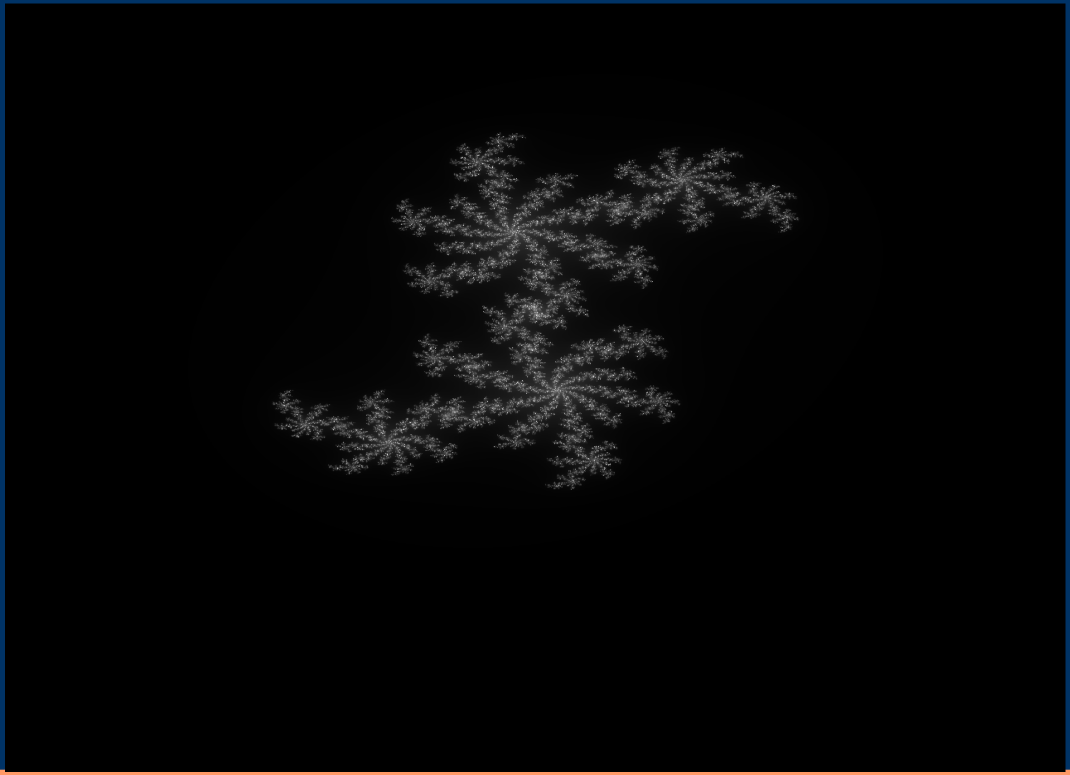
Fractals

- A mathematically generated pattern that is reproducible at any magnification or reduction.
- An example mathematical chaotic system



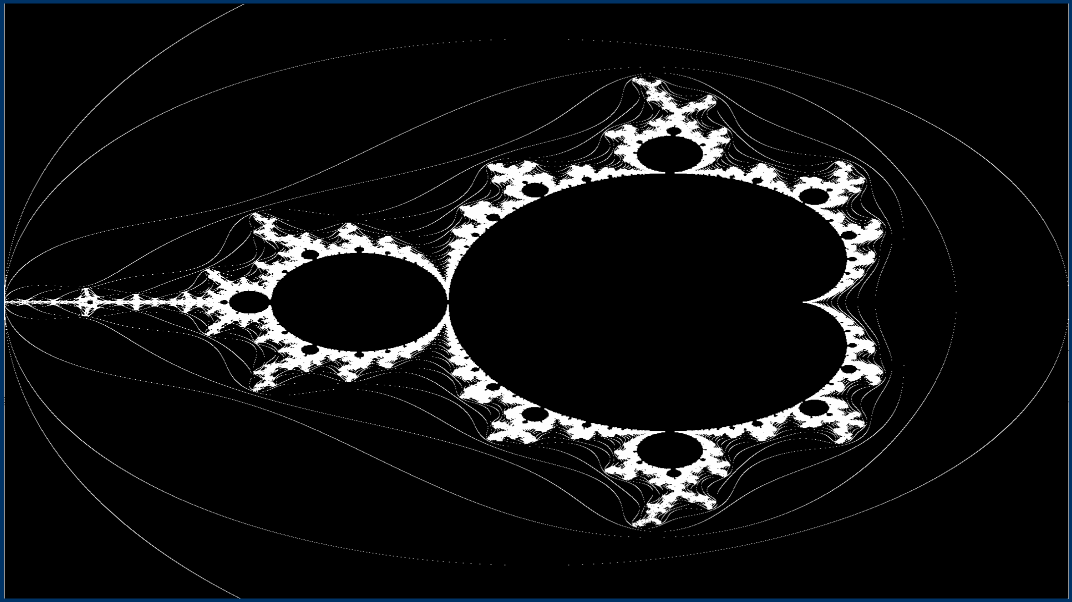
Julia Set

- Complex recursive equation
- $z(n+1) = z(n)^2 + C$
- C constant, z(0) based on point



Mandelbrot Set

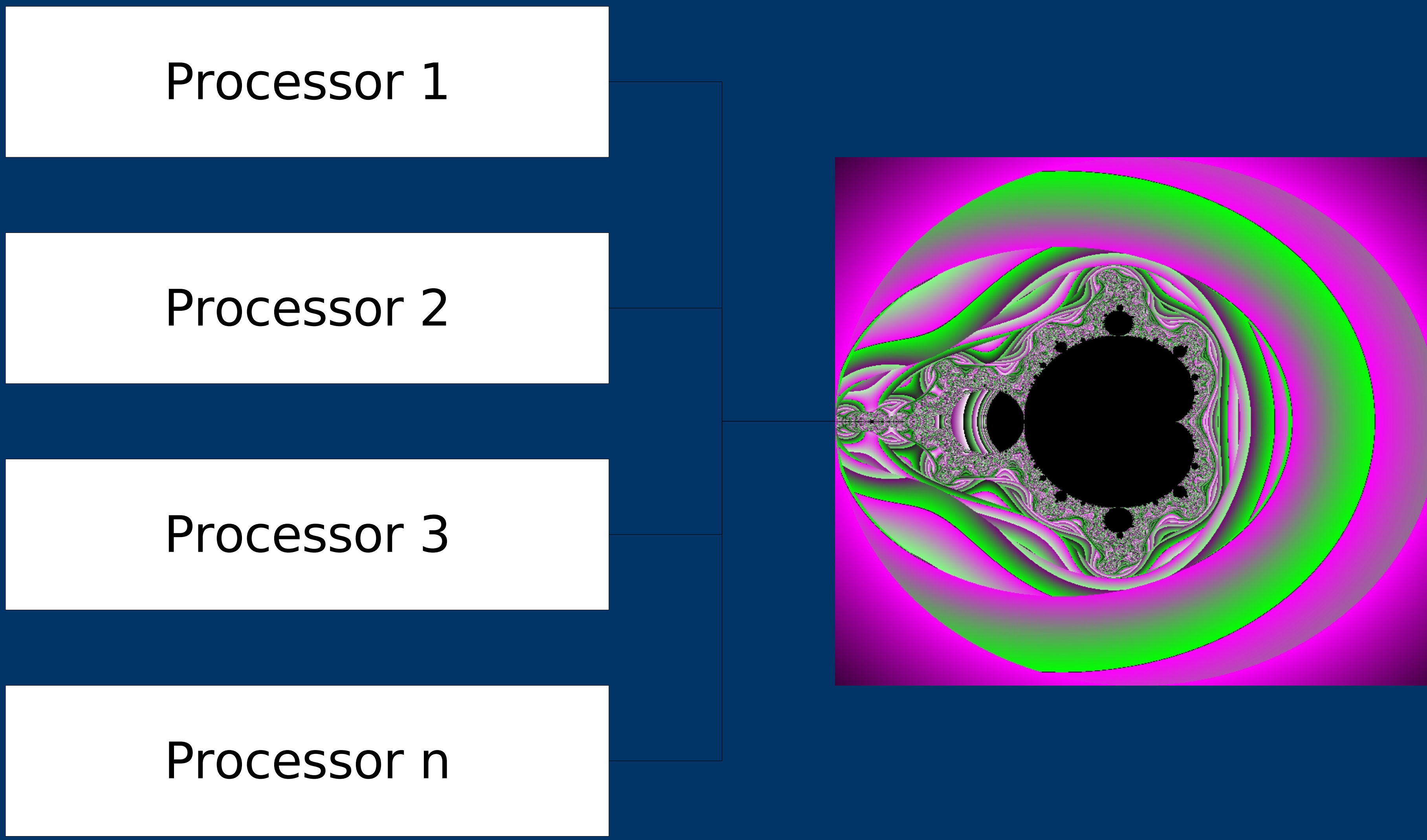
- Complex recursive equation
- z(n+1) = z(n)^2 + C
- z(0) = 0, C based on point
- Set of all Julia set fractals



Supercomputing

- Fractal images are “Embarrassingly parallel” and thus lend themselves to supercomputing and the Message Passing Interface (MPI)
- In the case of the Julia set video, processors can share the load by generating different frames
- Each pixel can be calculated independently, processors split the image and calculate portions.

Supercomputing



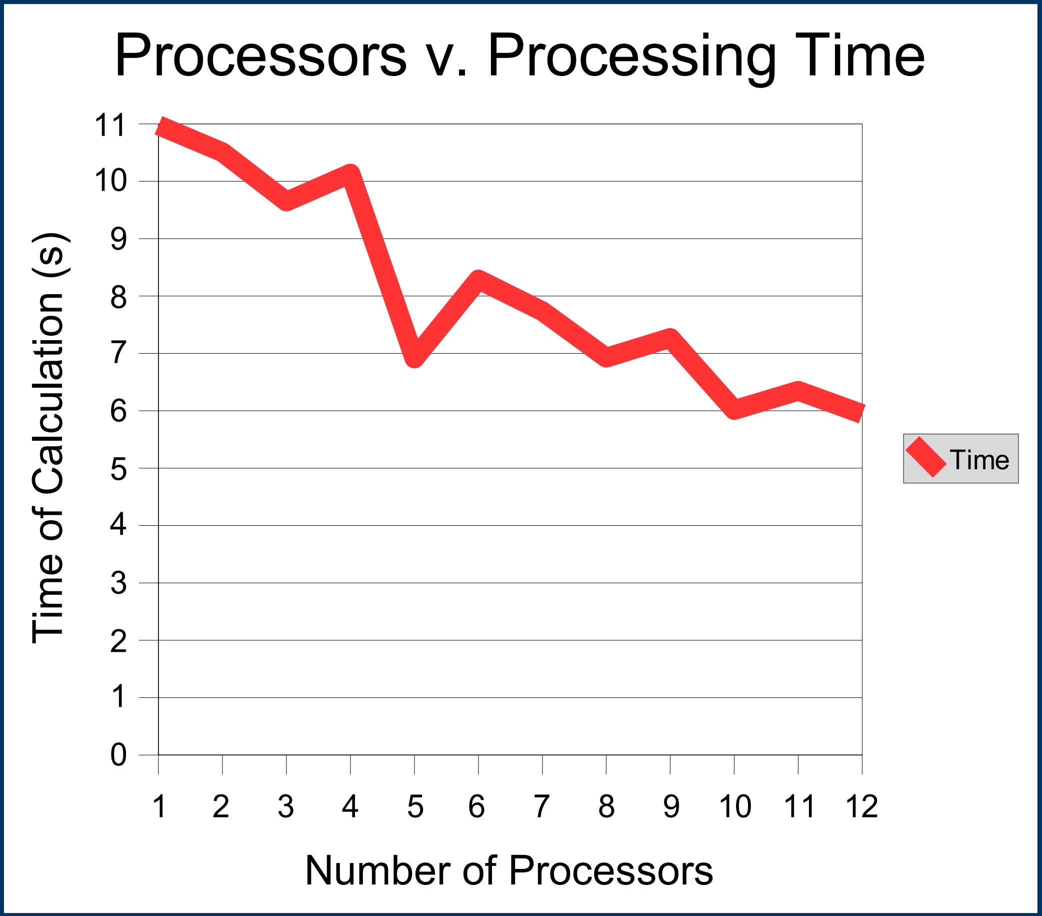
Other Work

- Binary *.pgm output – smaller files, less i/o time
- “Buddhabrot” – different rendering algorithm
- Julia set video – stringing together consecutive Julia set fractals to create a video

Results

- Performance is increased with more processors.
- Speed is not the original time divided by number of processors
- This is due to time for messages to pass between different processors.
- More message passing time for more processors.

Results



$$t_p = \frac{iterations * n}{p} + (p-1)(t_{data}) + k$$