CSE633 Fall 2010

MANDELBROT REAL TIME ZOOMING IN DSM ARCHITECTURE

AUTHOR: PRAKASH NAIR
PROFESSOR: DR. RUSS MILLER
DATE: NOV 30,2010

Scope of the Presentation

- Define Mandelbrot set and the zooming feature
- Algorithm for sequential and parallel approach
- Implementation
- Results
- Screenshots
  - Environment: 2 Nodes x 2 Physical processor
  - 1 Physical processor → 2 logical cores (intel hyper threading)
- Limitations and Future Scope
- References

MANDELBROT SET

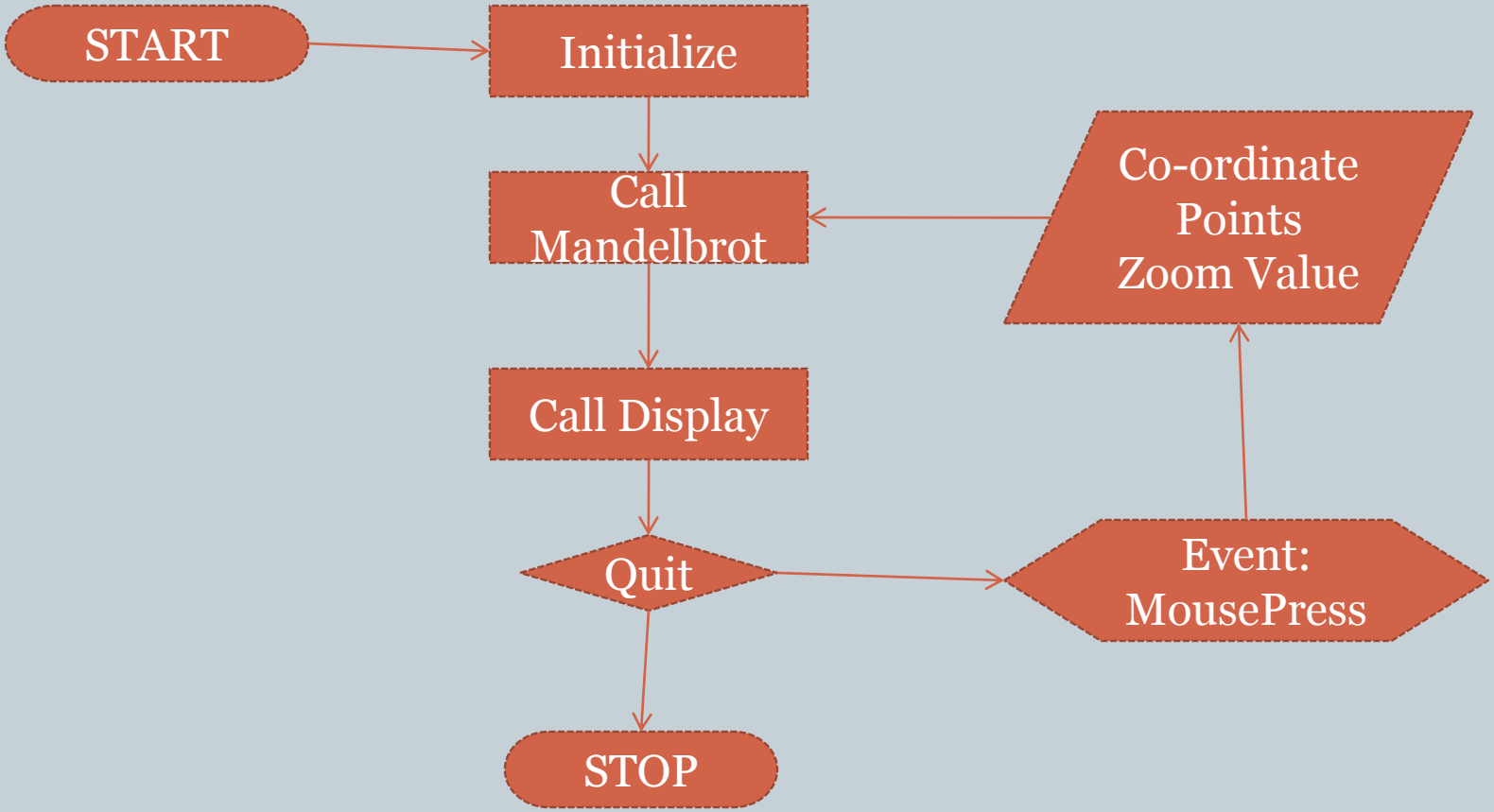
- The Mandelbrot set is a mathematical set of points in the complex plane, the boundary of which forms a fractal.
- $z_{n+1} = z_n^2 + c$
- Iterations: Multiple of 400
- Bail out: 2

- For Julia Set, keep ‘c’ constant.

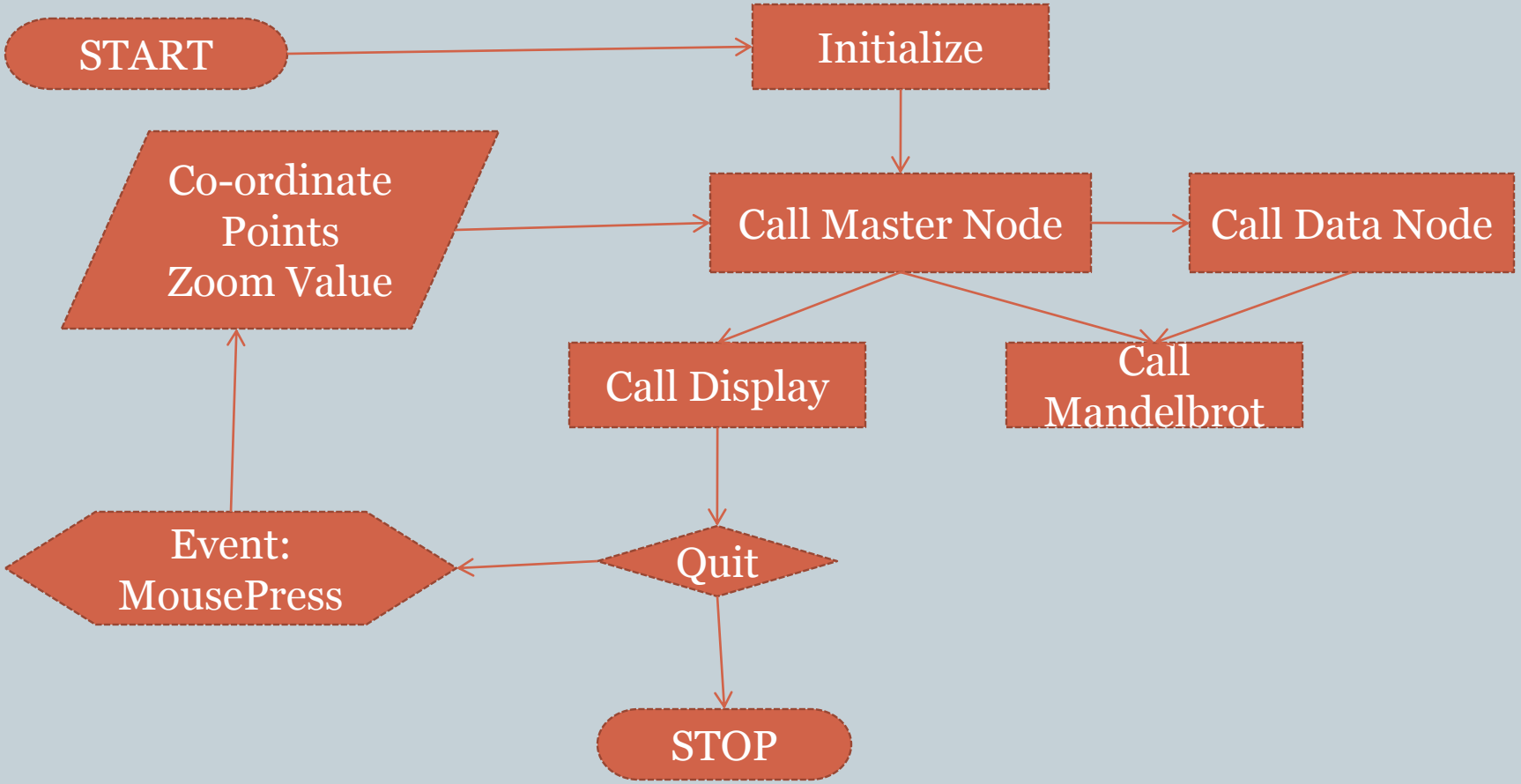
APPROACHES FOR DIFFERENT ARCHITECTURE

- SISD ARCHITECTURE
- SMP ARCHITECTURE
- MPP ARCHITECTURE
- DSM ARCHITECTURE

Sequential Algorithm



DSM Algorithm



Mandelbrot Pseudo Code

- Plot is of 400x400
- Range_of_points are the selected points, which needs to be extrapolated onto the 400x400 plot
- History Data Type and Zoom variable
- Converting (x,y) co-ordinate points to cartesian co-ordinates on complex plane.

```
c_re = MinRe + x*(MaxRe-MinRe)/(ImageWidth-1);
c_im = MaxIm - y*(MaxIm-MinIm)/(ImageHeight-1);
```

Mandelbrot Pseudo Code (cntd)

- Master Node triggers the Data Nodes (Broadcast)
- Split the plot amongst number of processor
- Split the range_of_points amongst number of processor
- For each Processor find,
- Points with minimum and maximum number of iterations
- Store the iteration count for each points in the plane

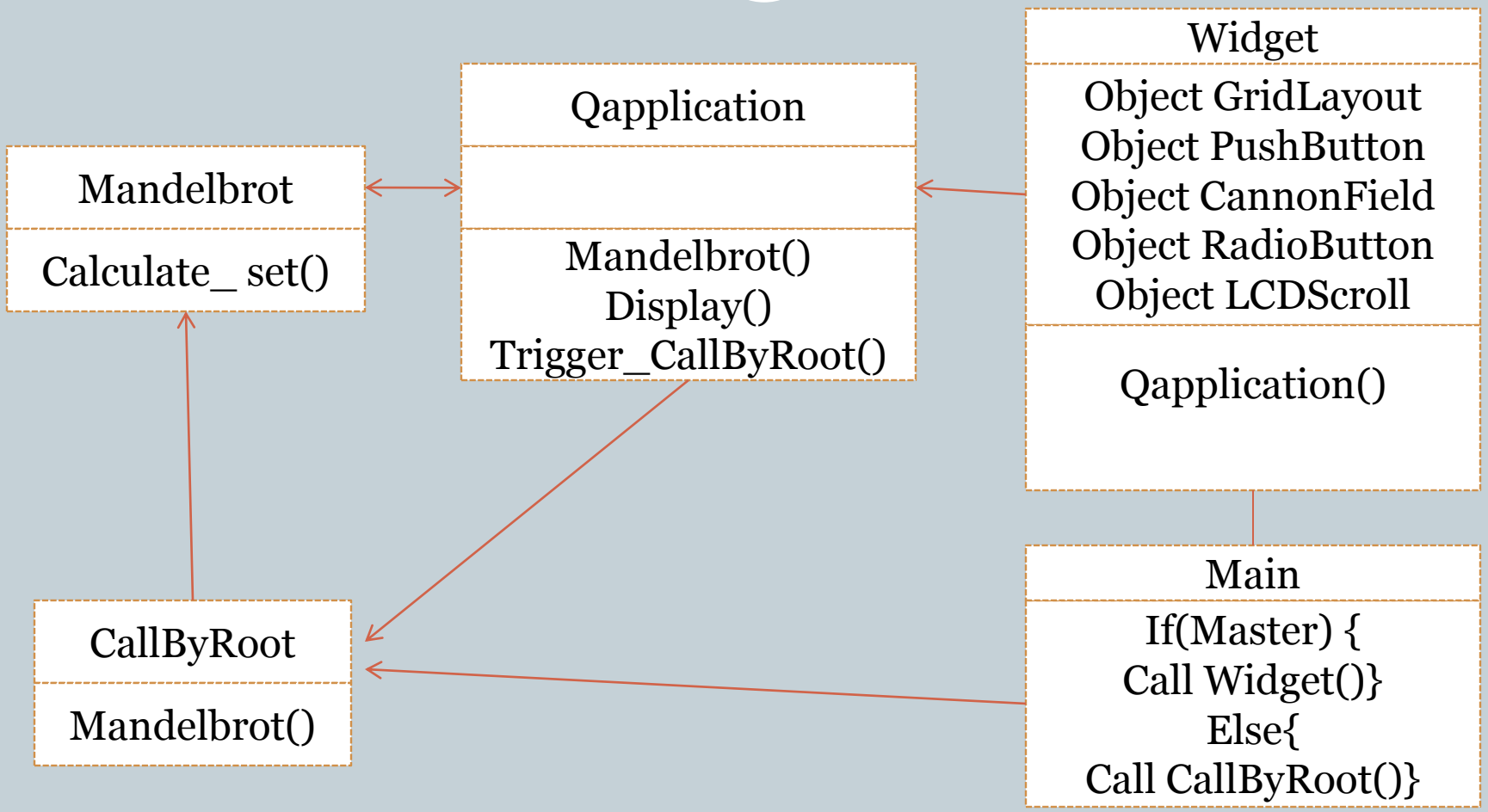
Mandelbrot Pseudo Code (cntd)

- Call the Reduction mechanism for collecting the points with minimum and maximum iteration
- Call the Gather mechanism, to collect the iteration count for all the points
- Master Node calls the Display method
- In Display method, divide the plot amongst the threads
- In U2, intel Xeon, we have 2 physical cores and 4 logical cores (HT). Thus, Number of threads initialized is 2 - 4

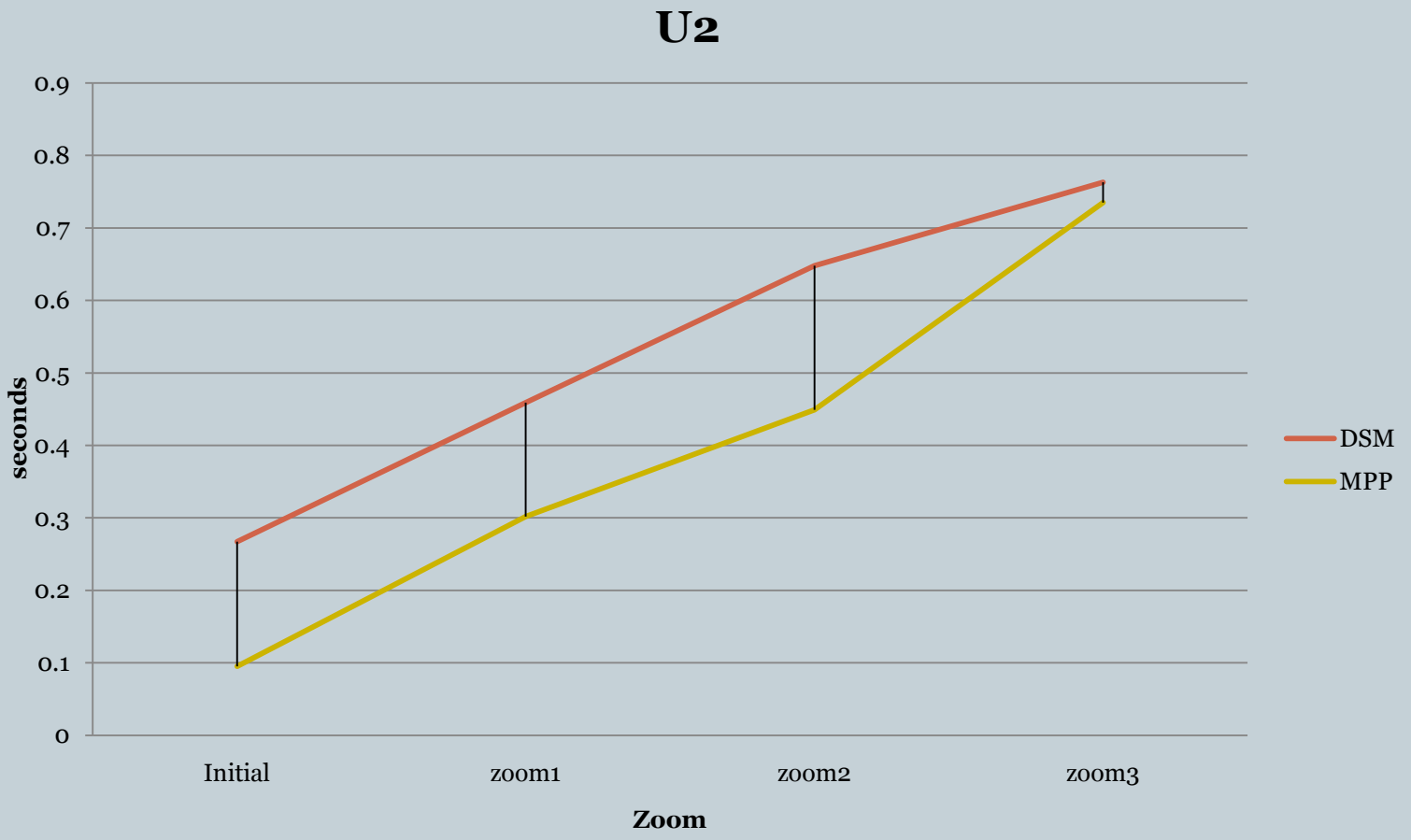
Mandelbrot Pseudo Code (cntd)

- MPI_Reduce()
- MPI_Gather()
- MPI_Bcast() {As a triggering event}
- Grep –c “processor” /proc/cpuinfo
- OMP for,critical,section
- GMP

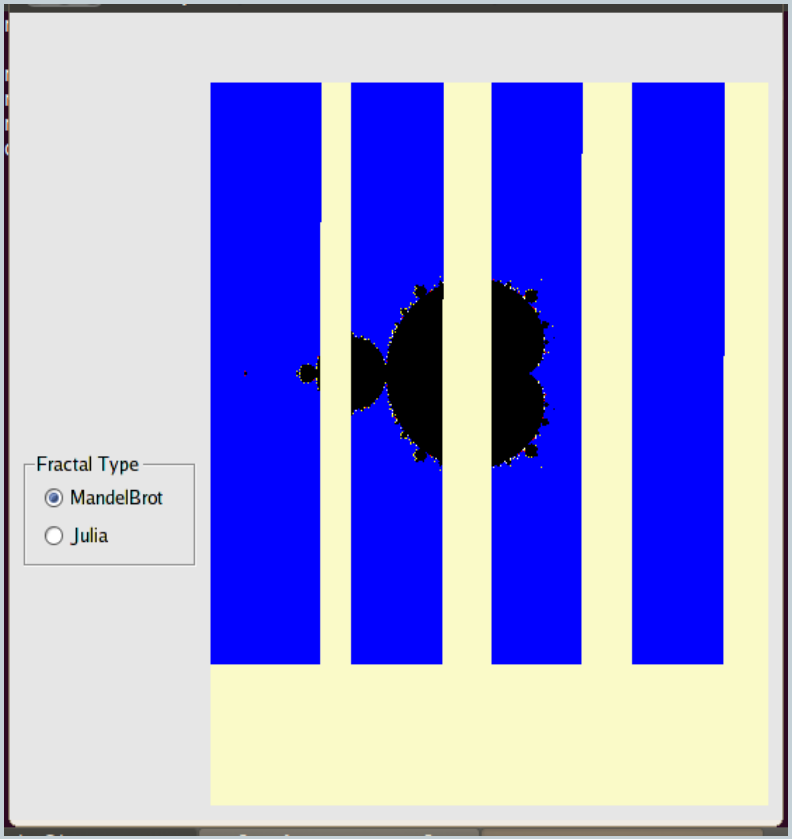
Implementation

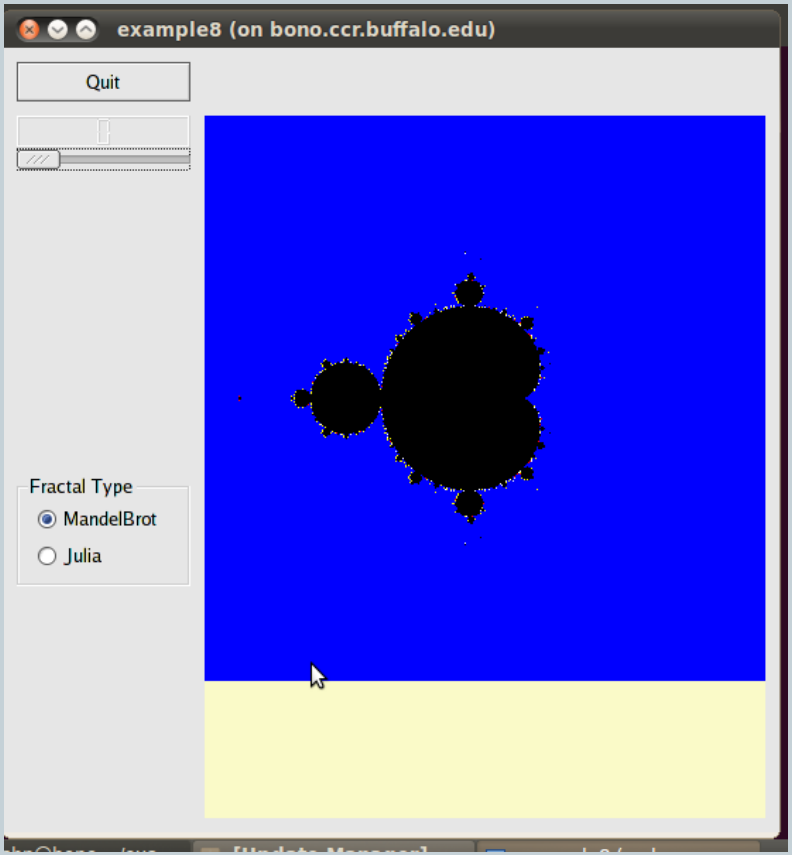


Results

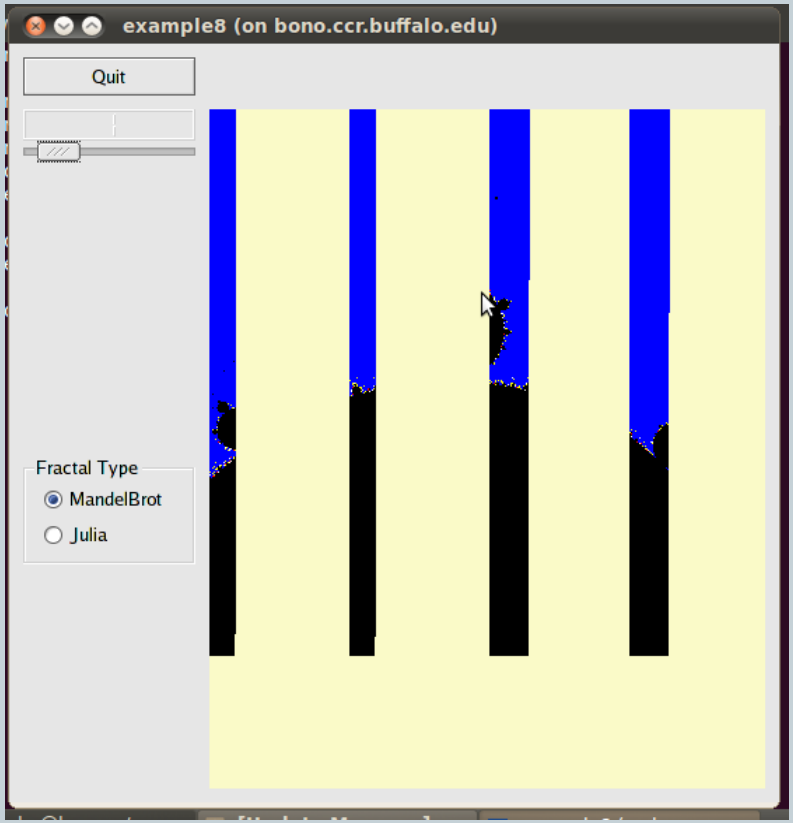

U2
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
seconds
Initial
zoom1
zoom2
zoom3
Zoom
DSM
MPP

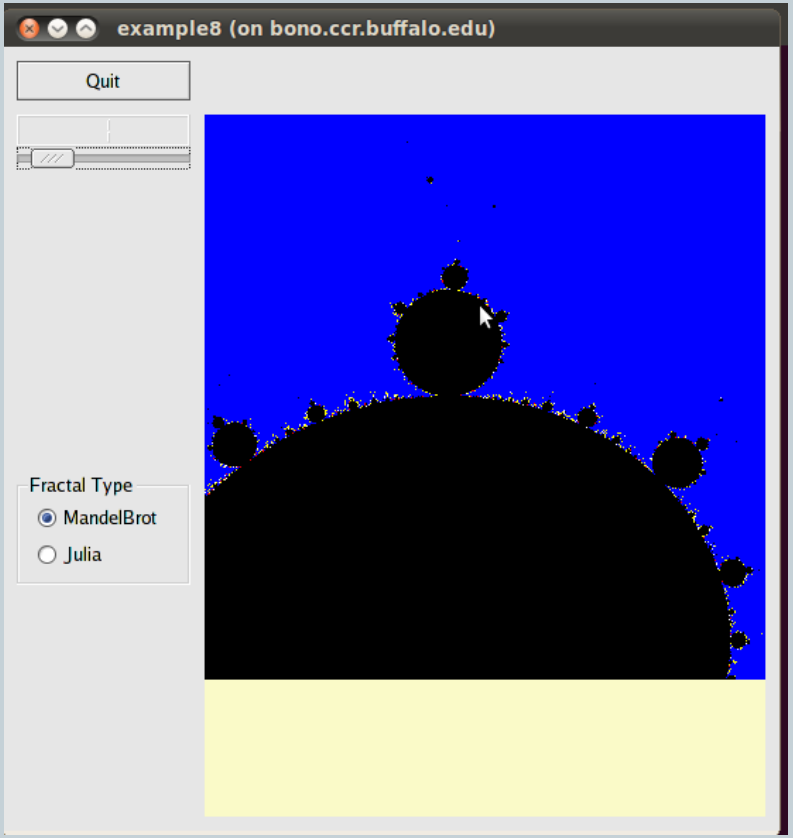
Screenshots (Initiate)



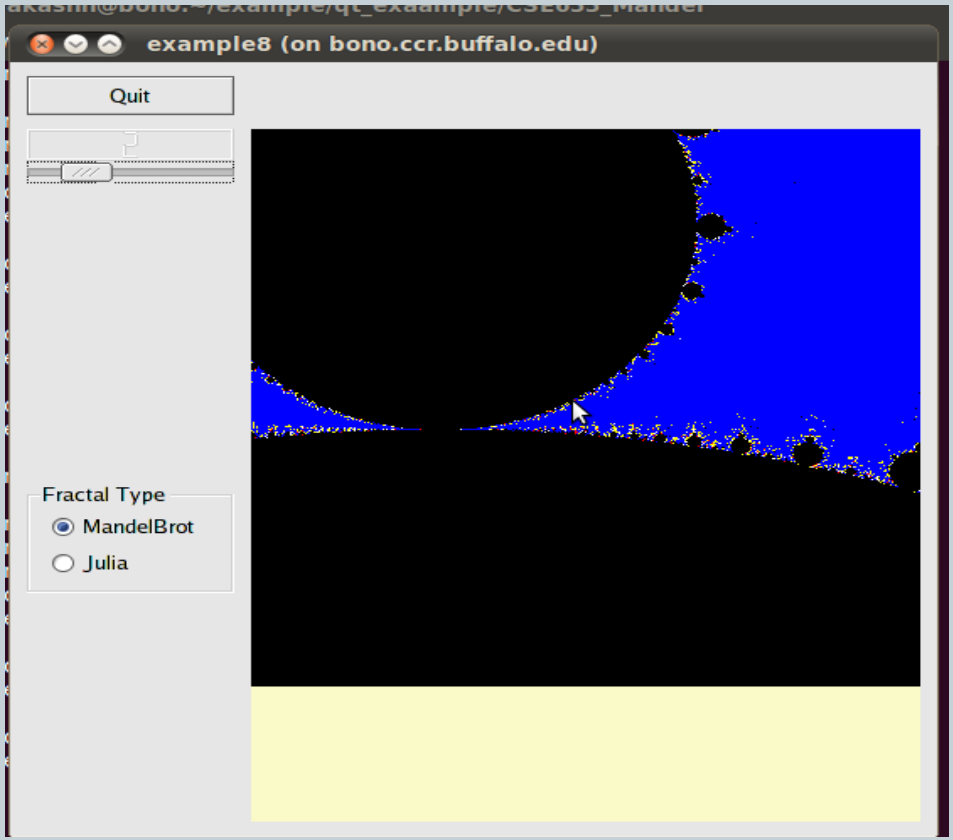


Screenshots (Zoom 1)

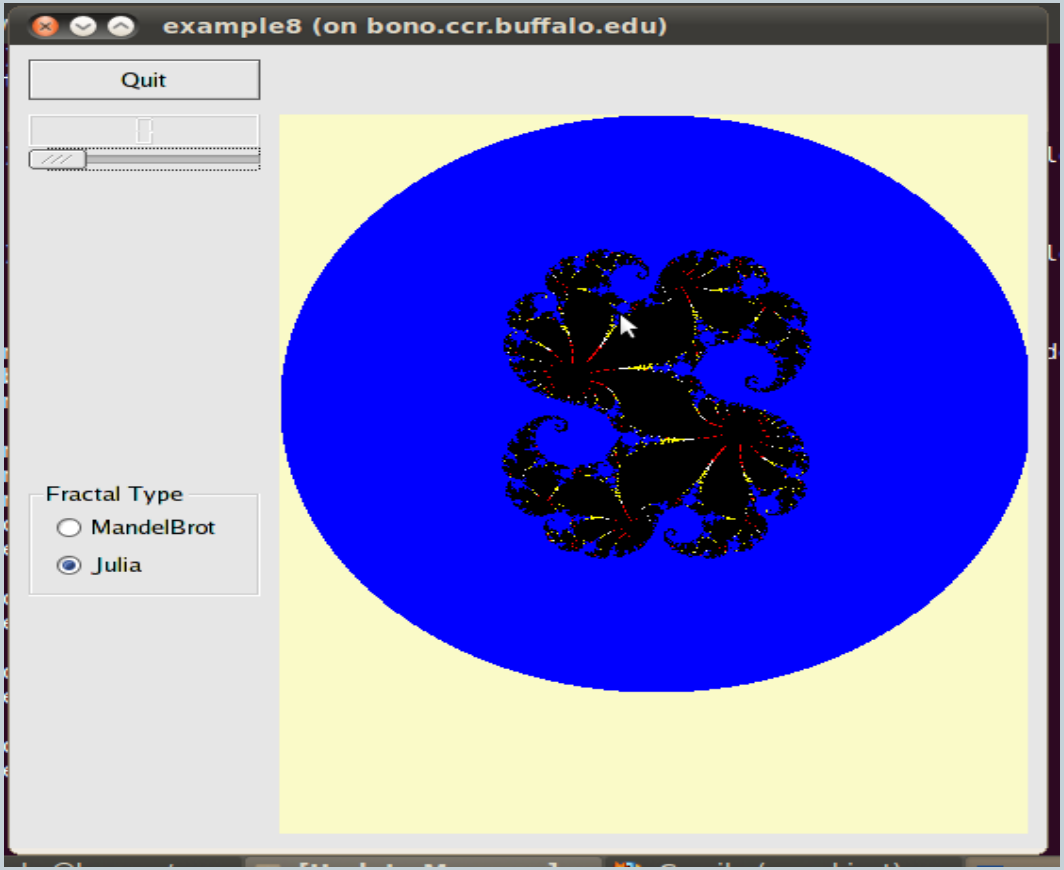




Screenshots (Zoom2)



Screenshot (Julia set Initiate)



Limitations and Future Scope

- The zooming beyond 7x does not show significant insight. More analysis required on this.
- Though the sequential algorithm is optimal, the re-engineering for the MPI version could have better
- More arenas for using OpenMP

References

- http://doc.qt.nokia.com/3.3
- Wikipedia.org
- Notes by Prof M. Jones
- Multi-core Faculty Training_v21.01_Student Workbook (Intel OpenMP 3.0)
- MPI:The Complete Reference –Vol 1.0
- GMP:GNU multiple precision library