MONASH University
Information Technology

ASSESSMENT COVER SHEET

Student ID number	28098552
Given Name	Ming Shern
Family name	Siew

Unit Name and Code:	FIT3143, Parallel computing		
Campus:	Clayton		
Assignment Title:	Assignment 1: Mandelbrot set		
Name of Lecturer:	Christopher Watkins		
Name of Tutor:	Mr. Omar Al-boridi		
Tutorial Day and Time:			
Phone Number:			
Email Address:	Msie0001@student.monash.edu		
Has any part of this assignment been previously submitted as part of another unit/course? ☐ Yes ☐ No			
Due Date:	06/09/2019	Date Submitted:	06/09/2019

All work must be submitted by the due date. If an extension of work is granted this must be specified with the signature of the lecturer/tutor.

Extension granted until (date)____________________Signature of lecturer/tutor ____________________________

Please note that it is your responsibility to retain copies of your assessments.

Intentional plagiarism or collusion amounts to cheating under Part 7 of the Monash University (Council) Regulations

Plagiarism: Plagiarism means taking and using another person's ideas or manner of expressing them and passing them off as one's own. For example, by failing to give appropriate acknowledgement. The material used can be from any source (staff, students or the internet, published and unpublished works).

Collusion: Collusion means unauthorised collaboration with another person on assessable written, oral or practical work and includes paying another person to complete all or part of the work.

Where there are reasonable grounds for believing that intentional plagiarism or collusion has occurred, this will be reported to the Associate Dean (Education) or delegate, who may disallow the work concerned by prohibiting assessment or refer the matter to the Faculty Discipline Panel for a hearing.

Student Statement:

- I have read the university's Student Academic Integrity Policy and Procedures.
- I understand the consequences of engaging in plagiarism and collusion as described in Part 7 of the Monash University (Council) Regulations http://adm.monash.edu/legal/legislation/statutes
- have taken proper care to safeguard this work and made all reasonable efforts to ensure it could not be copied.
- No part of this assignment has been previously submitted as part of another unit/course.
- I acknowledge and agree that the assessor of this assignment may for the purposes of assessment, reproduce the assignment and:
  i. provide to another member of faculty and any external marker; and/or
  ii. submit it to a text matching software; and/or
  iii. submit it to a text matching software which may then retain a copy of the assignment on its database for the purpose of future plagiarism checking.
- I certify that I have not plagiarised the work of others or participated in unauthorised collaboration when preparing this assignment.

Signature .... Date…06/09/2019…………………………………

* delete (iii) if not applicable

The information on this form is collected for the primary purpose of assessing your assignment and ensuring the academic integrity requirements of the University are met. *Other purposes of collection include recording your plagiarism and collusion declaration, attending to course and administrative matters and statistical analyses. If you choose not to complete all the questions on this form it may not be possible for Monash University to assess your assignment. You have a right to access personal information that Monash University holds about you, subject to any exceptions in relevant legislation. If you wish to seek access to your personal information or inquire about the handling of your personal information, please contact the University Privacy Officer: privacyofficer@adm.monash.edu.au*

Assignment 1: Mandelbrot set

Parallel Computing

Siew Ming Shern
Students, School of IT
Monash University
Melbourne, Australia
msie0001@student.monash.edu

Abstract**—**Mandelbrot set is one of the most important mathematical problem that involve mathematical equations to form a fractal. This article illustrates the efficiency of parallel processing compared to sequential processing and discover the impact that number of processors to parallel processing when dealing with Mandelbrot set using Message Passing Interface (MPI). Row partition is chosen to perform the parallel processing in Mandelbrot set because it can achieve the simplest possible workload balance on each node for parallel processing compared to some other approach and can be supported by the theoretical speed up difference on sequential processing from parallel processing is 2.1975% and 4.4321% for N = 2 and N=4 for its respective number of processor based on Amdahl's law which indicates the suitability of the method to perform parallel processing.

Keywords**—Mandelbrot set, MPI, sequential processing, parallel processing, partition, theoretical speed up difference, Amdahl's law.**

I. Introduction

Mandelbrot set was named after Benoît Mandelbrot who first discover the significance role of fractional Brownian motion to pay a tribute for his enormous contribution in variety of disciplinary study. [1] Mandelbrot set is a product of applying iterations of mathematical functions with complex quadratic polynomials and has been found involved with Julia set problem. [2]

From mathematical standpoint, Mandelbrot set have properties that can generate each pixel on complex plane with functions, f(k) = k^2 + c repetitively and use previous computation to recompute a new value.[3] Thus, for any n^{th} terms of k, it will return a summation of square root of k_{n-1} and c, where c is a constant representing complex number. Recurrence relationship for Mandelbrot set can defined as

$$k_0 = 0$$
$$k_n = k_{n-1}^2 + c \quad (1)$$

Parallel processing for Mandelbrot set will be done with Message Passing Interface (MPI) to improve the time taken to compute Mandelbrot set. Method to partition task in parallel processing are row partition which divides the sub-bricks in fractal of Mandelbrot set to various processors which perform the computation assigned to them. The performance of the parallel computing algorithms for both partition schemes are analyzed with their respective speed up factor based on the execution time of the parallel processing. Amdahl’s law analysis will be used to compare the speed up factors with theoretical speed up to ensure that the partition schemes is done to improve the computation time of sequential processing which is the crucial objective of this assignment.

II. THEORETICAL SPEED UP ANALYSIS

A. *Bernstein’s analysis*

First and foremost, Mandelbrot set must first satisfy Bernstein’s analysis in order to be

considered a Parallelizability programs.
Bernstein's conditions that must be satisfied:

$$I_2 \bigcap O_1 = \theta$$
$$I_1 \bigcap O_2 = \theta$$
$$O_1 \bigcap O_2 = \theta$$

where I_1 and O_1 represent the input and output for the first process P_0 whereas I_2 and O_2 represent the input and output for second process P1.
If Mandelbrot set does meet Bernstein's conditions, it is believed that Mandelbrot set can be parallelizable.

Equation (1) is used to determine whether Mandelbrot set is parallelizable.

$$k_0 = 0$$

$$k_n = k_{n-1}^2 + c$$

Where

$$I_1 = 0 \qquad O_1 = k_0$$
$$I_2 = k_{n-1}^2 + c \qquad O_2 = k_n$$

Using Bernstein's conditions,

$$0 \bigcap k_n = \theta$$
$$k_{n-1}^2 + c \bigcap 0 = \theta$$
$$k_0 \bigcap k_n = \theta$$

Since Bernstein's conditions shows processes P_0 and P_1 can be executed in parallel but k_n is depending on k_{n-1} if n is not 0 which indicates that the recurrence relation may have output dependency and may be inevitable to avoid communication overhead that can affect the speed up factor. Therefore, the result of Bernstein's conditions indicates that Mandelbrot set maybe parallelizable, but there will be a sign of degradation.

B. *Amdahl's law*

Sequential processing for Mandelbrot set's algorithm is analyzed to discover the theoretical speed up of the Mandelbrot set process with the multiple processors. Nonetheless, Amdahl's law is denoted with equation (2) is used to calculate the theoretical speed up factor when using multi processors for Mandelbrot set.

$$\boldsymbol{S(p) = \frac{1}{r_s + \frac{r_p}{p}}} \qquad (2)$$

Where

$\boldsymbol{r_p = parallel\ ratio\ (parallelizable\ portion)}$

$\boldsymbol{r_s = serial\ ratio\ (non-parallelizable\ portion)}$

$\boldsymbol{p = number\ of\ processors}$

Table I shows the calculated theoretical speed up factors S(p) for the number of processors p = 8 (i.e. for a single multicores computer with 8 logical processors) and p = 24.

TABLE I THEORETICAL SPEED UP FACTOR USING AMDAHL'S LAW

Number of Processors, p	2	4	6	8	10
Speed Up Factor, S(p)	1.9995	3.9979	5.9952	7.9914	9.9866

The r_s in speed up factor is represented by the portion for writing the Mandelbrot set content into file whereas the r_p is represented by generating Mandelbrot set portion.

III. DESIGN OF PARTITION SCHEMES

Both Partition scheme will have master node (rank 0) to open a new file and be responsible to gather result of all other node which also includes itself and write into the output file. The master node will then compute the amount of workload to be segregate to all nodes including itself. All computer nodes will start computing the amount of color for its respective coordinate in complex plane for Mandelbrot set to a buffer that store result of all color of partition scheme. A buffer is used to reduce the communication overhead and ensure that all nodes receive message and sends their message only once in any parallel processing. Once any nodes end its computation, it should send the result back to master nodes to write in the result from buffer to file.

A. *Row Segmentation-based Partition Scheme*

The root nodes will compute the workloads for each processor by dividing the height of y-axis on complex plane with number of processors. Once root nodes had done computation, it will broadcast the workload to all other nodes to start the Mandelbrot set computation. The following formula can be described as below:

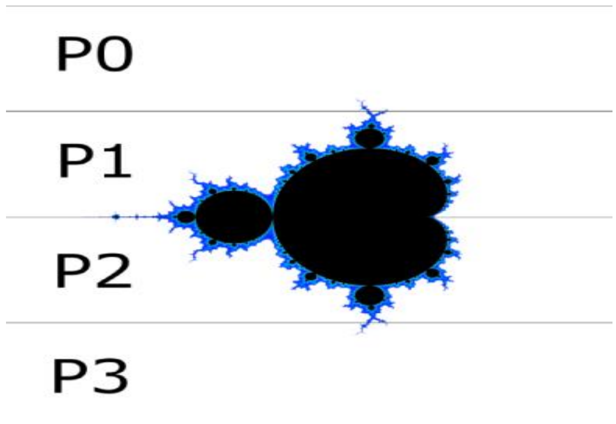
$$workloads\ per\ processor = \frac{iYmax}{number\ of\ processor} \quad (3)$$

Where iYmax is the height of y-axis on complex plane.

Once all other computer nodes has received its message, they will responsible to compute their respective start of the fractal functions loops and end of the fractal functions loops using their rank value and workloads per processor value to determine the partitions region on Mandelbrot set that the nodes should compute .However, these computation requires a bit of critical thinking to determine the first iteration value and last iteration value. Therefore, mathematical formula can be formed to determine both first iteration value and last iteration value for y-axis loops using following:

$$first\ iteration\ value = workload * (rank + 1)$$

$$last\ iteration\ value = workload * (rank)$$



(a) Figure 2: Row Partition Schemes

Figure above shows the rows partition scheme for number of processors = 4.

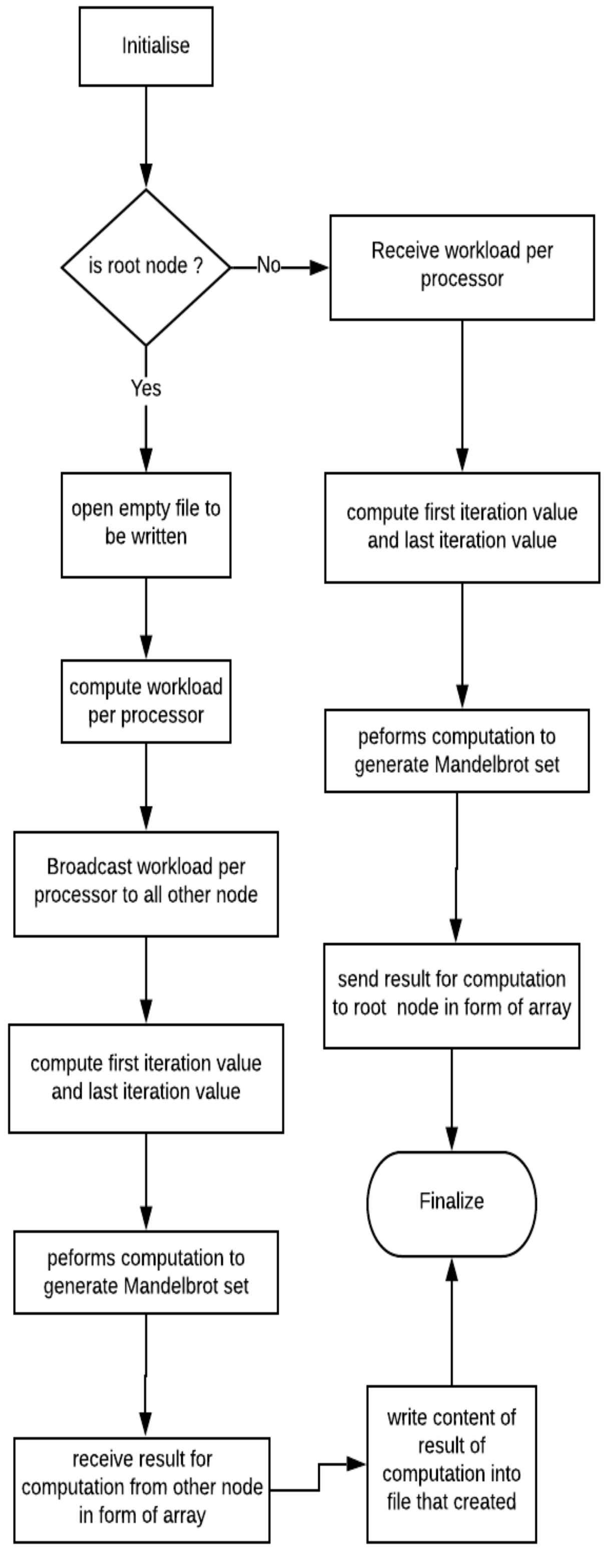
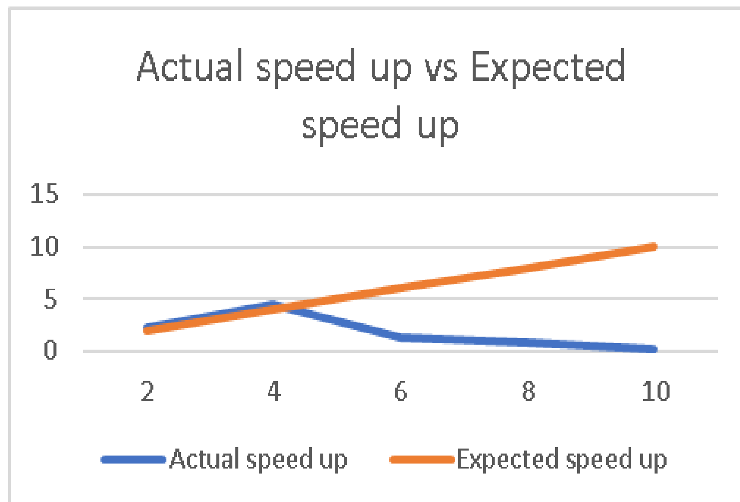


Figure 3 Flowchart for Row Partition Scheme

Number of Processors, p	2	4	6	8	10
Speed Up Factor, s(p)	2.1975	4.4321	1.2674	0.7674	0.2674



IV. RESULTS AND DISCUSSIONS

The result of speed up factor with Amdahl's law shows that Mandelbrot set can be optimized with parallel processing but there's a sign of limitation on how much is too much. This can be seen number of processors improves the computational time but also degrades at some number of processors. This limitation could be contributed from communication overhead that the parallel programs could not avoid, and it can be supported with Mandelbrot set breaching one of the Bernstein's condition. The best possible explanation to this is that output dependency in Mandelbrot set requires large amount of communication regardless how well a parallel algorithm is implemented, it can't deny that all processor must at some points make communication which causes delay.

Therefore, the result of the experiment shows that there won’t be a perfect algorithm that could have a speed up factor running linearly but drastic steps to minimize the communication overhead can be done by using a buffer or any other method in order to improve the computational time.

VI. CONCLUSIONS

Mandelbrot set has output dependency as its biggest flaws that affect performance of any parallel algorithm. The experiment results show that the Mandelbrot set should be carefully implemented to ensure the communication overhead on parallel algorithm does not affect negatively on computational time.

The theoretical speed up factors may not consider the effect of delay caused by communication between processor. Thus, it will be developer responsibility to ensure effect that is not consider in theoretical speed up to be dealt drastically to allow program to run more efficiently. Moreover, it is found that no matter how well the workload is distributed if developer does not deal with potential communication overhead, performance degradation is almost certainly happened.

VII. REFERENCES

[1] Taqqu, M. S. (2013). Benoît Mandelbrot and Fractional. Statistical Science, 131–134.

[2] Schober, J. H. (1992). The area of the Mandelbrot set . Numerische Mathematik, 59-72.

[3] Everest,G , Poorten, A.V.D, Shparlinski, I & Ward, T . (n.d.). Recurrence Sequences. 2000 mathematics subject classification, 1-11.

APPENDIX

Computer specifications	***a) Intel Core i7-9700k*** ***b) 8*** ***c) 8gb*** ***d) 70 Gb/s***
Value of `iXmax`	**8,000 (default)**
Value of `iYmax`	**8,000 (default)**
Value of `IterationMax`	**2,000 (default)**

	Serial program	**Parallel Program**			
		MPI			
		(e.g., 2 logical processors)	**(e.g., 4 logical processors)**	**(e.g., 6 logical processors)**	**(e.g., 8 logical processors)**
Run #1	103.546013	98.1232	101.2322	153.4432	174.5445
Run #2	102.431013	97.4334	99.3432	151.1431	172.1415
Run #3	102.421125	95.8788	103.2233	152.4112	170.5411
Run #4	103.112733	97.6653	102.3421	152.1132	173.3345
Run #5	102.781893	98.6754	98.1098	153.4131	175.2211
Average time	102.858555	97.55522	100.1232	152.6712	**173.1565**