1. Who is in your group? (Only one writeup needs to be submitted per group.)

|  |  |  |
| --- | --- | --- |
| Name | Id | Email\_Id |
| Bhargav Valluru | 811109009 | [bvalluru@kent.edu](mailto:bvalluru@kent.edu) |
| Radha Krishna Murthy Chinta | 811121216 | [rchinta@kent.edu](mailto:rchinta@kent.edu) |

1. What assistance did you receive on this project? Include anyone or anything except your partner, the course staff, and the course materials/textbook.

We took a reference from the below site for a better understanding of the framework

<https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/ForkJoinPool.html>

1. How long did the project take? Which parts were most difficult? How could the project be better?

It took 3 and half weeks to implement the project, smarter and Lock-Based version was a bit challenging to us and that part took a lot of time for implementation.

1. What "Above & Beyond" projects did you implement? What was interesting or difficult about them? Describe how you implemented them?

So far, we haven’t implemented “Above & Beyond” projects. We will look forward to gaining the knowledge on them.

1. How did you test your program? What parts did you test in isolation and how? What smaller inputs did you create so that you could check your answers? What boundary cases did you consider?

Please find the test cases that will be covered in different scenarios

Test1:

***Total Population***

In this test, the whole population will be considered, and it will count it and the results are validated.

Test2:

***Zero Population***

In this test, we pass zero population and validate the results.

Test3:

***Population checks for coordinates***

In this test, the grid is divided into 20 and 25 and the population for the following rectangle (1, 1, 5, 4) is verified.

Test4:

***Population for bottom Four rows***

In this test, the grid is divided into 20 and 25 and the population for the following rectangle (1, 1, 20, 4) is verified.

Test5:

***A data file is empty***

In this test, we check if the data file is empty and whether it is throwing an error or not.

Test6:

***Middle Three Columns Population***

In this test, we are dividing the grid into 20 and 25 and check the population for the following rectangle (9, 1, 11, 25)

6. For finding the corners of the United States and for the first grid-building step, you implemented parallel algorithms using Java's Fork Join Framework. The code should have a sequential cut-off that can be varied. Perform experiments to determine the optimal value of this sequential cut-off. You need to examine cut-offs in two (really 3) places. First, you are looking at sequential vs. parallel versions of corner findings. Looking at V1 and V2 works here, vary the cutoff for V2. Second, you are looking at cut-offs in the grid-building step. Looking at V3 and V4 works here. There are two places to vary the cut-off for V4 - the summing of census data and the combining of grids. You don't have to worry about finding the optimal combination of cut-offs or anything, but we do expect data on varying cut-offs for corner finding and both phases of grid building.

Graph the results and reach appropriate conclusions. Note that if the sequential cut-off is high enough to eliminate all parallelism, then you should see performance answer the question - is this what you see?).

7. Compare the performance of version 4 to version 5 as the size of the grid changes. Intuitively, which version is better for small grids and which version for large grids? Does the experimental data validate this hypothesis? Produce and interpret an appropriate graph or graphs to reach your conclusion.

Grid is measured from (5,5) to (195,195). Data is available in the zip folder as PerformenceForV4andV5.txt. Version 4 works better for larger grids and performance increases with grid size. Vesion5 has little impact on the grid size.

8. Compare the performance of version 1 to version 3 and version 2 to version 4 as the number of queries changes. That is, how many queries are necessary before the pre-processing is worth it? Produce and interpret an appropriate graph or graphs to reach your conclusion. Note you should time the actual code answering the query, not including the time for a (very slow) human to enter the query.

9. If you worked with a partner:

a. Describe the process you used for developing and testing your code. If you divided it into pieces, describe that. If you worked on everything together, describe the actual process used. For example, discuss how long you talked about what, in what order you wrote and tested the code, and how long that required.

* We have spent almost 8 to 10 hours analyzing the requirement and verifying what was already provided. (Appx. 2 days)
* Then we spent around 12 to 14 hours analyzing the existing code provided as a source code. (Appx. 3 days)
* Since there were 5 versions that we had to develop and test. One of us worked on a Smarter and Lock-Based version and a Smarter and Sequential Version. The other member worked on the remaining methods, which we have tested together properly. It took 1 week for each of us to implement their tasks unit test them. We had regular connects to know each other’s work to make our work easier while we integrate the code at the end.
* Finally, we integrated the code and tested it together. Integration took around 5 days of time and we have utilized the rest to write the test cases for the project.

b. Describe each group member's contributions/responsibilities in the project.

Bhargav worked on Smarter and Lock-Based, Simple and Parallel, and Smarter and Sequential.

Radha Krishna worked on Smarter and Parallel, Simple and Sequential and handled most of the errors at the time of integration.

The Remaining work is shared by both of us. (Analyzing the requirement, existing code, and test cases)

c. Describe at least one good thing and one bad thing about the process of working together.

Pros:

1. It taught us how to work as a team, at the end of the day that is what we are going to see in the outside world when we meet the practical world.
2. We also learned how to delegate our tasks meaningfully which will result in minimum dependency on each other.
3. We had multiple ideas while implementing the logic, discussions helped to make it only better all the time.

Cons:

1. Since we had different course schedules, it was a bit challenging in the initial days to set up frequent meetups to discuss status. Which we overcome at a later point in time.