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

Venkata Sri Nishitha Nallaparaju (id: 811064952, mail\_id: vnallapa@kent.edu)

Venkata Sai Prashanth Mamunuru (id: 8110742051, mail\_id: vmamunur@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 didn’t take any help from others. But we got the data from following websites

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

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

It took 3 weeks to implement the project. Implementation of Smarter and Parallel

And smarter and Lock-Based took lot of time for understanding and implementation.

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

We haven’t implement Above & Beyond projects.

5. 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?

We have developed a testcase which covers lot of scenarios like

* Data file is empty test: In this testcase, we are checking if data file is empty weather it is throwing error or not.
* Total Population test: In this testcase it takes all population and count it and validate the results
* Population checks for coordinates: In this testcase, we are dividing grid into 20 and 25 and check the population for following rectangle (1, 1, 5, 4)
* Population for bottom Four rows test: In this testcase, we are dividing grid into 20 and 25 and check the population for following rectangle (1, 1, 20, 4)
* Zero population test:
* Middle Three Columns Population Test: In this testcase, we are dividing grid into 20 and 25 and check the population for 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 ForkJoin 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 finding. 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 the summing of census data and in 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 discussed the requirement of the project and what is the available code and how it is working. We took 3 days for the discussion.
* We divide the implementation of five methods by both of us like .one member worked on Simple and Sequential and Smarter and Sequential. Other member worked on remaining methods and tested properly. It took 1 week for implementation and individual testing
* Finally we integrate the code and tested together. Integration took 1week of time. We eliminated the duplicate code and fixed error caused by integration

4)Write the testcases for the project

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

Nishitha worked on Simple and Sequential and Smarter and Sequential.

Prashanth worked on Simple and Parallel, Smarter and Parallel and Smarter and Lock-Based.

Remaining work is shared by both of us.

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

When we are working together, it took lot of time to discuss the requirement because of different opinions on how to implement the project and lot of duplicate code developed by both of us. It is difficult to find the errors caused by integration.

It took less time for implementation of the project since it is shared by both of us and also, we get to know the different way of thinking about project