Q1. How would you implement the complete set of tasks in HW3-PC using a single MapReduce job? (Note that your program should emit outputs for each task into a separate file.) [300-400 words]

ANS: Well, I think I implemented this assignment in one MapReduce task. The reason why I put them together is that it saves tons of time from repeatedly scanning the input file. Another reason I using this mechanism is that all the MapReduce Tasks only require just one scanning of the whole dataset and each tasks extracts different fields of information they need, so it is definitely implementable using one MapReduce job. The way how I did it is designing a new Writable class for storing my huge data sets. The Writable Class (say, MyWritable) has different variables to store different information for different tasks. MyWritable also inherits Read and Write function for MapReduce Tasks to pass intermediate file. Taking this assignment as an example, I scanned the whole data set line by line, extracting substring(index1, index2) for one variable for one task, and at the same time, extracting substring(index3, index4) for another variable for another task. Since all tasks using the same StringBuffer, there is no need to scan the dataset twice. As a result, lots of time are saved for scanning the dataset, which is also the most time-consuming step among all steps in MapReduce job. Now, let us discuss how to emit outputs for each task into separate file. First of all, I didn't do that in my assignment, so I did read some materials online and conclude a possible way to implement that (may not successful in real case). I think the key point to emit outputs in separate file is to have different reducers handling different tasks. Based on that, Mapper should write different keys for different tasks in order to separate outputs into different Reducers. Then, each Reducer can perform its tasks and write its output in different file for each task. Lets take this assignment as an example, considering we have 9 tasks, we can create 9 reducers for each task. When doing the Mapper job, generates a MyWritable object for the whole job. After that, Mapper emit different pairs of output into intermediate file, like <CO1, MyWritable>, <CO2, MyWritable>. Using a Partitioner Class to separate different key sets into different Reducers, like, **1 to Reducer1, **2 to Reducer2. Each Reducer handling specific field of MyWritable variables and write the corresponding outputs for a specific task (Say, **1 for task1, **2 for task2). At last, the output files should be in the path with different filenames like part-00000, part-00001 … There is one obvious weakness of my implementation, which is the huge data structure MyWritable is being passed into different Reducer even if they actually don't need that much information (a waste of memory), a way to handle that is to write subclass for MyWritable that specific to different tasks.

Q2. Suppose that the U.S. Census Bureau has introduced a new scheme to disseminate demographic data. In addition to releasing the entire dataset as a batch at the end of a census cycle, the Census Bureau is now providing changes to the population (number of new births or deaths for each census block for a particular time period) as a continuous stream of updates. How would you extend your solution to support this new scheme in order to provide a more up-to-date view of the U.S. population? [400-500 words]

ANS: Given this situation, one way that I come up with is using streaming that taught in class. However, that changed the implementation from MapReduce into Spark. Anyway, I am going to discuss how to use Spark Streaming to implement this task. First of all, we should start at a point using the entire dataset provided by U.S. Census Bureau to gain the whole data information that we need to provide, and this is easy and doable since it is a static dataset. Using Spark can easily get the results which is basically the same as we get from MapReduce job. Then suppose U.S. Census Bureau provides a stable stream in its port, we can build a connection socket which periodically listening to that port and receiving data stream. Once a new stream is available on that port, Spark Driver Program runs Spark jobs to process the data, instead of dealing with the whole new dataset, we can use incrementally reduction on the dataset to reduce the operation and memory each time dealing with the dataset. In my option, instead of reading the whole population, we can just focus on the new birth and deaths in the whole population and do the incrementally reduction mechanism, which is cost-effective and time-saving. Making a second job that specific for handling the periodically increments is suggested. Using Spark dealing with the incoming steam and then we can provide a more up-to-date view of the U.S. population. I think Spark is based on MapReduce and providing more functionalities and having better performance than MapReduce. Compared with MapReduce, using Spark could have faster calculating speed since Spark job is processed in cache. Besides that, Spark could also improve its performance by using parallelism in data receiving, which means Spark are more sensitive than MapReduce when dealing with huge stream of data. I think Spark is definitely the first choice to implement this request. However, if we have to implement this request based on our previous implementation using MapReduce, I think a way to do that is using the API that MapReduce provided in HadoopStreaming. However, it seems that the HadoopStreaming only supports the old MapReduce version, which greatly limits the performance of MapReduce job. On the other hand, HadoopStreaming only integrate with several few languages like python and java. Concluding from the above, I think using Spark to deal with the continuous stream is more preferable than using MapReduce.

Q3. Instead of supporting a set of fixed queries as you have done in HW3-PC, you are being asked to implement a solution that can run any arbitrary query on the census dataset using the Hadoop infrastructure. How would you design your solution to cope with this requirement? [400-500 words]

ANS: For this discussion, I think the key point to handle this is designing a program that specify the request into machine language. I would like to store the Reference Table that U.S. Census Bureau provided first, then using a program to translate the user query into machine language. Since it's a totally another story, I don't want to explain more on this. For example, if user types in "Which state has highest population in U.S." My MapReduce Program should takes in the extras parameters says "state, population, highest" for MapReduce job, which correspond to <key, data, operation>. Then, Mapper uses the Reference Table to find the data field for specific query, say population in each state may be in the field (index1, index2). After that, create a new MyWritable object to store the data and also the operation. Mapper writes the output <key, MyWritable> to reducer. Before sending to the reducer, I will design a Partitioner to specify the operation for each query. The Partitioner reads MyWritable object and extracts operation variable. Probably referring to a Operation Table to find the corresponding code for that operation (and there should be different Reducers handling different operations, oh god, its a huge amount of work). Do a switch case and direct the task into right Reducer (for this example, calculate the sum number of a key) class and send the original dataset <key, MyWritable>. Once the Reducer received the message, there is no need for itself to know what operation user want to do. The mission of itself is to run the operation designed in its body and emit the output, in this example, it will generate something like <CO, 10000>, <AL, 9800). After that, there may be one more loop of MapReduce to get the real result for the query. MapReduce Job says there needs another loop to get one state which has the highest population, then a Mapper reads key pairs and sends to a Reducer aimed at finding the highest number. After this loop, the result will be generated something like <Highest, CO>. Following this pattern, the MapReduce job could hopefully handle any query that user want to know. However, there are also some weakness of this kind of design. First, MapReduce Job will scan the whole dataset for each query, because the MapReduce Job don't know what are the other date field the next query want to use. This problem may be solved if the MapReduce use a lazy performance, which means user feeds in all his queries and then start the Job. Another con is the query is strictly limited on how many operations that have been specified in Reducer Classes, which means user actually can't get everything they want if there lack some operations in Reducers.

Q4. How would you extend your current solution to identify locations that are most similar to each other and also those that are most dissimilar to each other? Your similarity measures may include distributions relating to: age, gender, etc. [400-500 words]

ANS: To solve this problem, I think the key point is to find an expression system to compare two fields of data. Based on my solution, for example, if I calculate the distribution of age and also the distribution of age by gender, I want to calculate a value (lets say idea from Distance Value and P-Value, but there are also other algorithms like Hash Value, Smith-Waterman algorithm) by defining an expression:

H(value) = sqrt((x1-y1)^2 + (x2-y2)^2 + … + (xi - yi)^2) / sqrt((size1 - size2)^2);

In this example, I would calculate:

H(value) = sqrt ((Age1 - Gender1)^2 + (Age2 - Gender2)^2 + …) / sqrt((size1 - size2)^2);

The expression basically calculate the distance value of two locations (because they are locations), and divide the value by the average size (also calculate in distance form) of the location length (idea from P-value). When comparing two locations, I want to calculate the value of each slot in one location, and then calculate the H value of the two locations. In this example, Age and Age by Gender have same number of partitions, and their percentage value (comparing numbers makes less sense in this case) are almost the same, after calculating the H value for them, we can tell whether they are similar or not (predefine distribution of values table). If the two locations are not in the same size, we can still use the above equation to calculate the value. And actually it turns out with much more higher value which means they are less likely similar. According to the expression above, locations with the same size are more likely to have a small score and locations with large difference in size are more likely to have a higher score. When we compare the score of pairs of locations, smaller score corresponds to higher similarity and higher score corresponds to smaller similarity. In this assignment, there are high similarity between House Value(owner-occupied) and Contract Rent than between House Value(owner-occupied) and Houses (Urban or Rural). And Age has more similarity with Age by Gender than with Sex. However, there are some shortage of this algorithm. First of all, I put equal emphasis on each slot of location. Second, I could not strictly define a range that suitable for all size of different locations. Lastly, this algorithm could only compare numbers in locations whereas they could not tell some potential similarity between two locations, say age by gender distribution has connection with gender distribution because the sum of age by gender equals each gender population.

Q5. Consider the case where two additional datasets has been made available to you. The first dataset includes information about *migration patterns* (alongside demographics such as age, gender, and educational levels) into and out of a state. The second dataset includes *economic data* such as number of new jobs that were created, the sectors that these jobs were created in, the average pay, educational levels requirements for jobs, etc. Describe how you will design a framework that allows you to make reasonably accurate projections about the expected population levels in a particular state. Assume that you have census, migration patterns, and economic data for 1990, 2000, and 2010. In the case of migration patterns and economic data assume that you also have yearly data for the past 10 years. [~500 words]

ANS: I think I would solve this problem by following steps. First of all, find the relating influential factors that attribute to the changes of population. For the census dataset, some majority influential factors could be contract rent, urban house percentage (there are definitely other factors, but I just list some major ones from the list in assignment). For the migration patterns, I think age, education levels are more important. Lastly, for economic data, I would pick number of new jobs, sectors of the jobs, average pay, education levels requirements. As to second step, I would determine how they affect the projection of the expected population. According to general acknowledgement, people are more likely to move into a new place if there has more job opportunities with higher salary, besides people also prefer places with high urban rates and have large sectors of jobs. On the contrast, higher rent and jobs requiring higher education may prevent people from coming here. Concluding from above, contract rent, age, education level may have inverse affects on the population, and the urban house percentage, numbers of new jobs, sectors of the jobs, average pay have the positive influence on the change of population. After that, the third step is to find how much influence they could have on the population changes (if I define affection rate into 3, high, medium, low). Then, for the number of new jobs and average pay must have the high influential constant, and the contract rent, education level have medium influential rate, and urban house percentage, age, sectors of jobs have the low influential constant. Based on my analysis, I can conclude a general expression of change of population:

P (population) = (C1 * #Jobs + C2 * $Avg-Paid - C3 * $Contract Rent - C4 * #Education level + C5 * %(ubran house percentage) - C6 * #age + C7 * #Sectors of Jobs) * P(population)

Given the above equation, I would do the MapReduce Job on those three input files, and extract related dataset as input to calculate my constant numbers (c1 - c7). This is like an machine learning step. I would divide my dataset into 2 parts, 2/3 for learning (building the equation) and 1/3 for verifying (check the equation). Since I have the imaginary equation with some unknown constant, then I uses existed dataset to find the best constant for my equation. After previous steps, I could get a theoretical reasonable equation to predict the population change based on

seven factors (listed above with underlines, NOTE: there is more than seven factors could affect, I just list a subset of them as an example). After all of those steps done, a specific equation is generated for predicting the population changes. We can predict population based on this equation in the next several years. However, there are still other tools to implement this algorithm, like linear regression, population prediction algorithm.