**Objectives**

* Compare and Contrast MapReduce performance when using larger HDFS block size (256 MB)
* Analyze and explain the processes MapReduce uses when dealing with larger datasets (160 GB)
* Understand, analyze, and explain the uses of the MapReduce features we have covered so far and their impact on the ability to process a MapReduce Job

**Outcomes**

  At the conclusion of this lab and lecture you will have gained experience using large datasets, large blocked datasets, and increased your ability to analyze and decompose a MapReduce Job request.

Create a folder in your private Github Repo called week-08, and a sub-folder per each deliverable item, named item-one, item-two, and so forth.  As always cite page numbers from the book when making conclusions.

**Item-one:** Re-Run the Analysis portion of last weeks initial assignment, this time using the large block storage located in 90and92-256 and 90-93-256 HDFS directories respectively.  Create a ReadMe.md file with your compartive analysis.  Run the following jobs:   MaxTemperature Job without compression on both datasets, with data input compressed (.gz and .bz2), without Intermediate Compression and then with it.  Run both sets of data with a Combiner and without a Combiner. Then run those programs again with multiple reducers, discovering the optimal number of reducers - explain in the ReadMe.md how you reached your conclusion.

**Item-Two:**  Using the 80-99 dataset and the MaxTemperature from Chapter 2 and the sample program from Ch 06 (COUNTERS) Create a Map Reduce job to do the following  (you will need to provide your analysis in your ReadMe.md file as well as all class files that produced your results:  You can make seperate classes or combine them into a single class.

* Write a job to find the Max and Min temperature for each year
* ~~Write a job to find Max and Min temp per year~~
* Write a job to find the **Average of the per year**Max and Min temperatures
* Write a job to find Median Temperature per year
* Write a job to find Std Deviation

**Item-Three**: Extending item-two, if possible write a combiner class for the above requirements.  If a writing a Combiner is not possible/efficient, explain why.

**Item-Four:** Using MaxTemperature and the 80-99.txt dataset, find station ID with largest percentage of MalFormed Records.  Provide the class files that will generate this address as well as the Answer in a ReadMe.md file.  Note Speed is not an issue here, the results are. You may not need a reducer for this.

**Deliverable**: Submit your Github URL

Item –one Analysis:

1. In Item-one we carry out all the steps which we did in week-07 assignment, but here the only difference was the dataset on which the jobs were performed.
2. I carried out the jobs on the datasets 90and92-256 and 90-93-256 using MaxTemperature, MaxTemperatureWithCombiner, With and without Compression, and lastly Compression and Combiner.
3. The major difference here as compared to week-07 was the time taken by each job, which was way more than as observed for the small datasets.
4. By using multiple reducers, I concluded that chaining helps to run the job simultaneously which saves time and the results are faster and quicker.
5. Below are some of the screen shots of the output and the graph.

Item –two Analysis:

1. **In this section, we first find out the Minimum and Maximum temperature per year on the 80-99 datasets.**

**To run the Code, follow the below steps:**

1. Copy MinMaxTemperature.java, MinMaxTemperatureReducer.java,from the src folder of Item-two iniside hadoopbook/ch-02/src/main/java/
2. Compile the code using command : hadoop com.sun.tools.javac.Main \*.java
3. Create a jar using command : jar cf minmax.jar \*.class
4. run the commnad using minmax.jar and MinMaxTemperature class.
5. The results will be displayed inside the output directory.
6. **To find out the Average temperature per year on the 80-99 datasets.**

**To run the Code, follow the below steps:**

1. Copy AvgMinMaxTemperature.java, AvgMinMaxTemperaturReducer.java,from the src folder of Item-two iniside hadoopbook/ch-02/src/main/java/
2. Compile the code using command : hadoop com.sun.tools.javac.Main \*.java
3. Create a jar using command : jar cf avgminmax.jar \*.class
4. run the commnad using avgminmax.jar and AvgMinMaxTemperature class.
5. The results will be displayed inside the output directory.
6. **To find out the Median temperature per year on the 80-99 datasets.**

**To run the Code, follow the below steps:**

1. Copy MedianTemperature.java, MedianTemperatureReducer.java,from the src folder of Item-two iniside hadoopbook/ch-02/src/main/java/
2. Compile the code using command : hadoop com.sun.tools.javac.Main \*.java
3. Create a jar using command : jar cf median.jar \*.class
4. run the commnad using avgminmax.jar and MedianTemperature class.
5. The results will be displayed inside the output directory.
6. be displayed inside the output directory.
7. **To find out the standard deviation per year on the 80-99 datasets**

**To run the Code, follow the below steps:**

1. Copy SDTemperature.java, SDTemperatureReducer.java,from the src folder of Item-two iniside hadoopbook/ch-02/src/main/java/
2. Compile the code using command : hadoop com.sun.tools.javac.Main \*.java
3. Create a jar using command : jar cf sd.jar \*.class
4. run the command using sd.jar and SDTemperature class.
5. The results will be displayed inside the output directory.
6. be displayed inside the output directory.