A8 Report

Abhishek Ravi Chandran, Chinmayee Vaidya, Chintan Pathak, Mania Abdi

March 26, 2016

High Level Design:

Using Spark for to run A4, which was previously run using Hadoop.

A4 High level Design:

We run 2 jobs one after the other to perform the analysis. The first job groups all the flights together by year and carrier and get the values scheduled elapsed time and price. This is then passed to the reducer, where we apply simple regression by calculating slope and intercept for the data. The we get the linear regression for N by calculating intercept+slope*N.

Then we have a java method that reads the output files from s3 to compare all the flights every year. The flight that has the lowest price the most number of times for all the years is returned and passed to the second job.. In the second job we only filter out the records for the carrier given by the above method. Then we calculate the median for every week.

A8 High level Design:

We read all the files from the input folder, on which we apply a filter to ignore all bad records. This is then transformed to a JavaPairRDD with the key as 'carrier year'. We then perform a persist operation with storage level as MEMORY_AND_DISK_SER, which serializes to store the objects in memory and spills to disk when memory limit is reached. We chose this as it is more space efficient, we use scala Serializable interface for our objects. The persist operation ensures that further operations on data happen in memory, no more file reads are necessary. We perform further map operations on the RDD that we have, to reduce the data to get values needed for calculating linear regression such as, number of records, total duration, total price, sum of duration^2 and sum of duration*price. This new RDD is converted into a HashMap, which is then used to perform linear regression and then find the cheapest carrier for both n=1 and n=200. Once we have the carrier for both n=1 and n=200, we then filter the first RDD we have which contains all the records to get only the carrier we need and perform more transform operations to get all the weekly median prices. The resultant RDDs are written to files.

Design Difference:

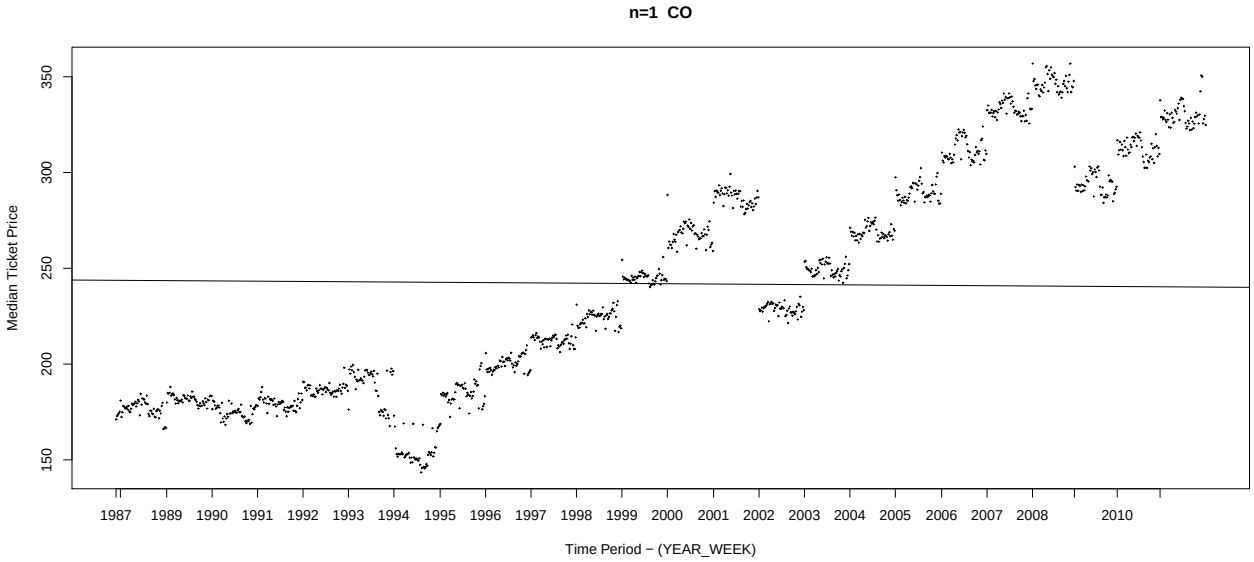
- Instead of performing 2 mapreduce jobs, we could complete everything in one job.
- In spark we just have to read the files once, when everything is in memory all other o perations are much faster.
- Reduced lines of code, since we can apply various transformations easily.
- Spark was much more easy to understand and implement.

Performance:

Hadoop: 2 jobs ran for ~1 hour with 3 m3.xlarge machines

Spark: ~20 mins with 3 m3.xlarge machines

Graph n=1



Graph n=200