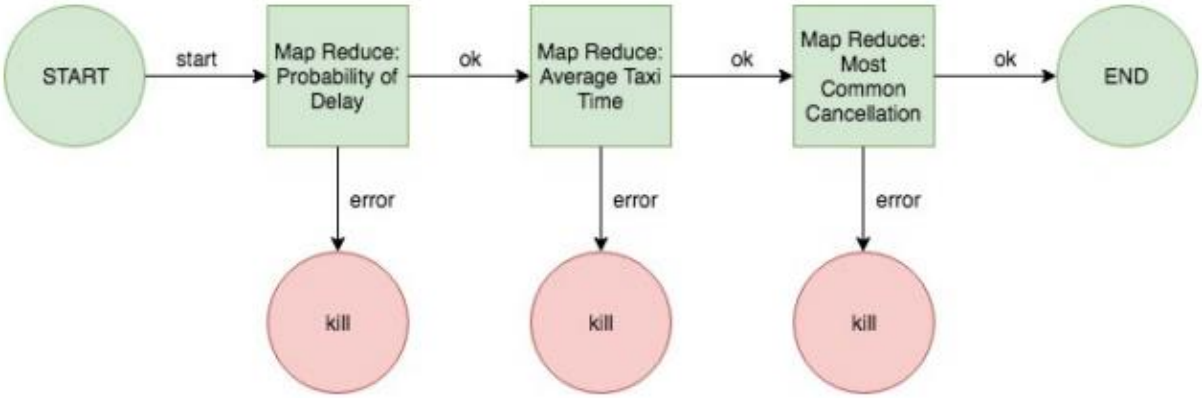
Flight Data Analysis

Yaohua Zhao

CS 644 Spring 2020

Part I. Oozie Workflow



Part II. Description of Algorithms

On-Schedule with MapReduce

1) If the "year" is valid and the delay time is less than 20 minutes, pass on-time: carrier code and '1'; not on-time: carrier code and '0' to the reducer.
2) Reducer calculates the addition of all the '1' appear for each unique carrier. Then the sum of on-time is divided by the sum of unique on-times and non-on-times. That will get the on-time rate of a specific carrier.
3) Reducer then sorts the on-time rate for all unique carriers and return the top 3 and bottom 3 of the results.
4) If the no output, print "No output".

Average Taxi Time with MapReduce

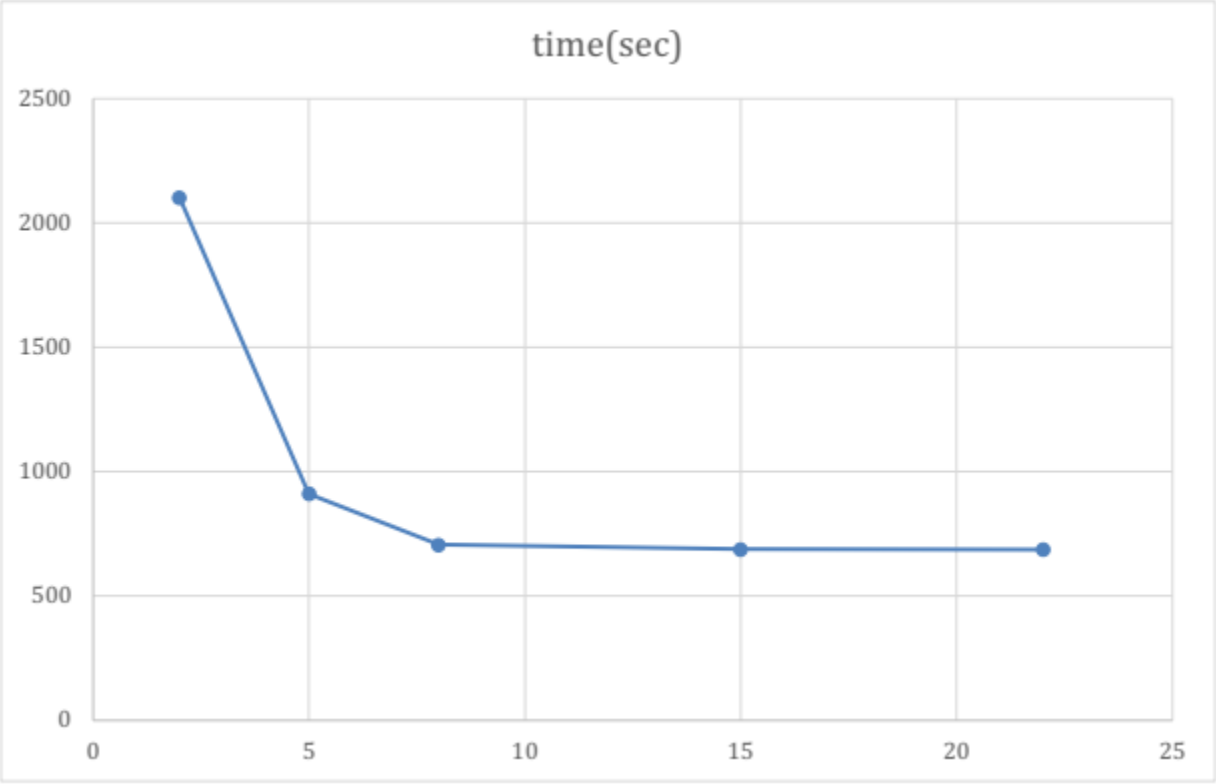
1) Mapper scans the dataset and get all the taxi-in or taxi-out which is not NULL. Then it will combine all the data as pair [original airport, taxi out] and [dest-airport, taxi in].
2) Reducer receives the pairs, then calculate the average of a airport, and get the a pair of [airport, taxi time] for the each valid airport.
3) Reducer sorts all the valid airport taxi time pairs, and then return the top 3 and bottom 3 of the airport with taxi time.

4) if the no output, print “No output”.

Most Common Cancellation with MapReduce

1) Mapper scans the dataset, gets all the “cancelled” with ‘1’ and the “cancel code’ without ‘NA’. Then combine them as pair.
2) Reducer collects the pairs and calculates the sum of times appear for each cancel code, and then reduces the pairs from mapper to [cancel code, happen times].
3) Reducer sorts all the pairs of [cancel code, happen times] and get the top 1 reason of the cancelation.
4) If the no output, print “No output”.

Part III. Execution time with the number of nodes increase

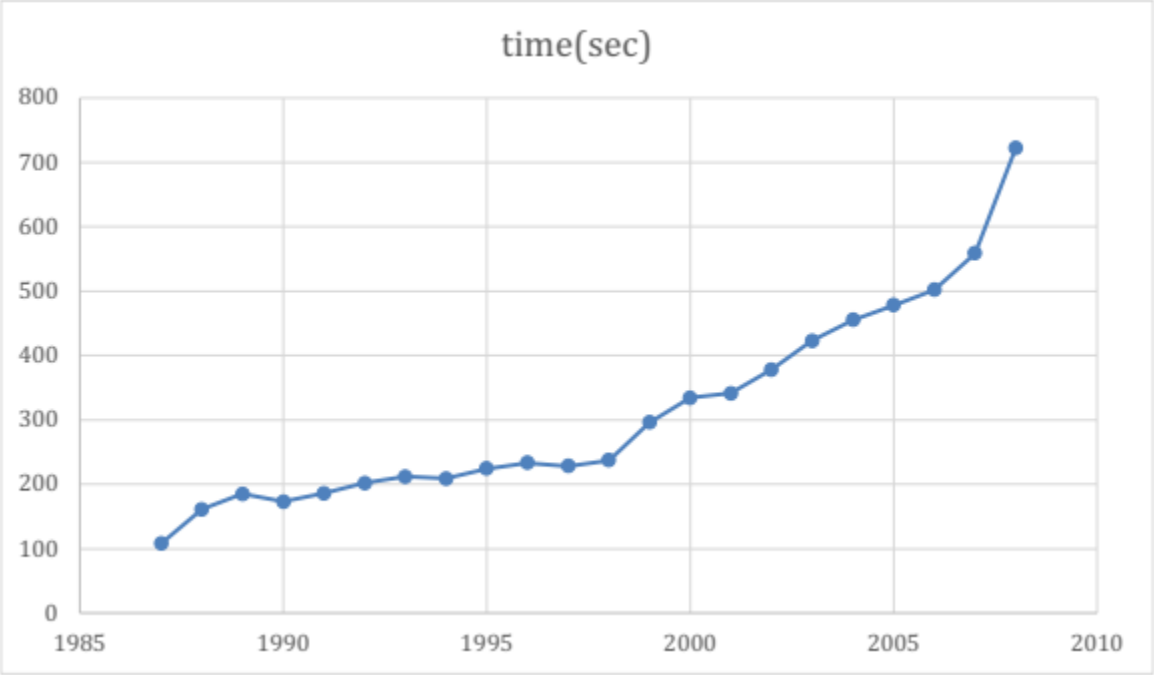


As shown in the graph, we can see that with the increase of the virtual machine number, the execution time drops significantly: from 2103 seconds with 2 VMs, to 687 second with 22 VMs. Since more nodes are taking part in the computing, less time is spent on Oozie and map-reduce.

However, as the number of the node further increases, the slope of the line is getting flat, which means the efficiency is not growing significantly. More nodes require much more

data transferring between nodes, and more data transfer significantly slows down the process as node number is greater than 15. Thus, the total time consuming is not decreasing when nodes number is over 15.

Part IV. Time Consuming with Data Expansion



According to the graph shown above, with the dataset expanded from 1987 to 2008, the time consuming is increasing as well. As the computing power for the whole program could be considered stable, the time consuming can represent the scale of the dataset during each year, and the slope between two point could stand for the degree of increase of dataset between two adjacent years. We can see a rough split point on 1998: less increase each year happened before 1998, while more increase each year happened after 1998. We can thus conclude that more and more people are choosing airplane as their transportation methods after 1998.