CS 644: Introduction to Big Data
Dr. Chase Wu, Qianwen Ye

Airline Analysis with Big Data

Prepared by Tianyu Hou, Gerald Wrona

A) Direct Acyclic Graph (DAG)

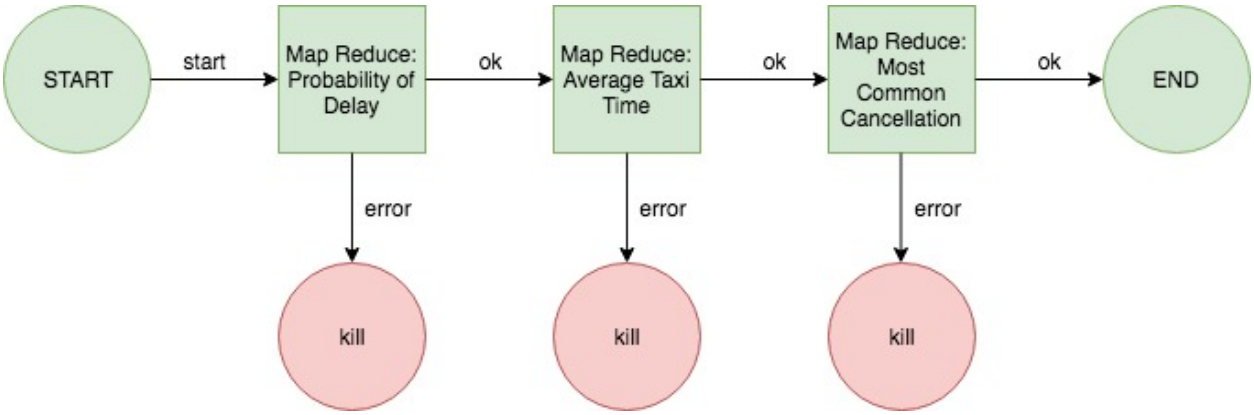


Fig. 1

To create our diagram we followed the instructions in the official Apache Oozie documentation found here (https://oozie.apache.org/docs/4.0.1/DG_Overview.html).

We created the diagram using the in-browser diagram tool Draw found here (https://www.draw.io/).

B) Description of Algorithms

On-Schedule with MapReduce

- i) 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.

- ii) reducer calculate the addition of all the ‘1’ appear for each unique carriers. Then the on-time sum divided by the sum of unique on-time and none-on-time. That will get the on time rate of a unique carrier
- iii) reducer than sort the on-time rate for all unique carriers. And return the top3 and worst3 of the results.
- iv) if the no output, print ” No output.”.

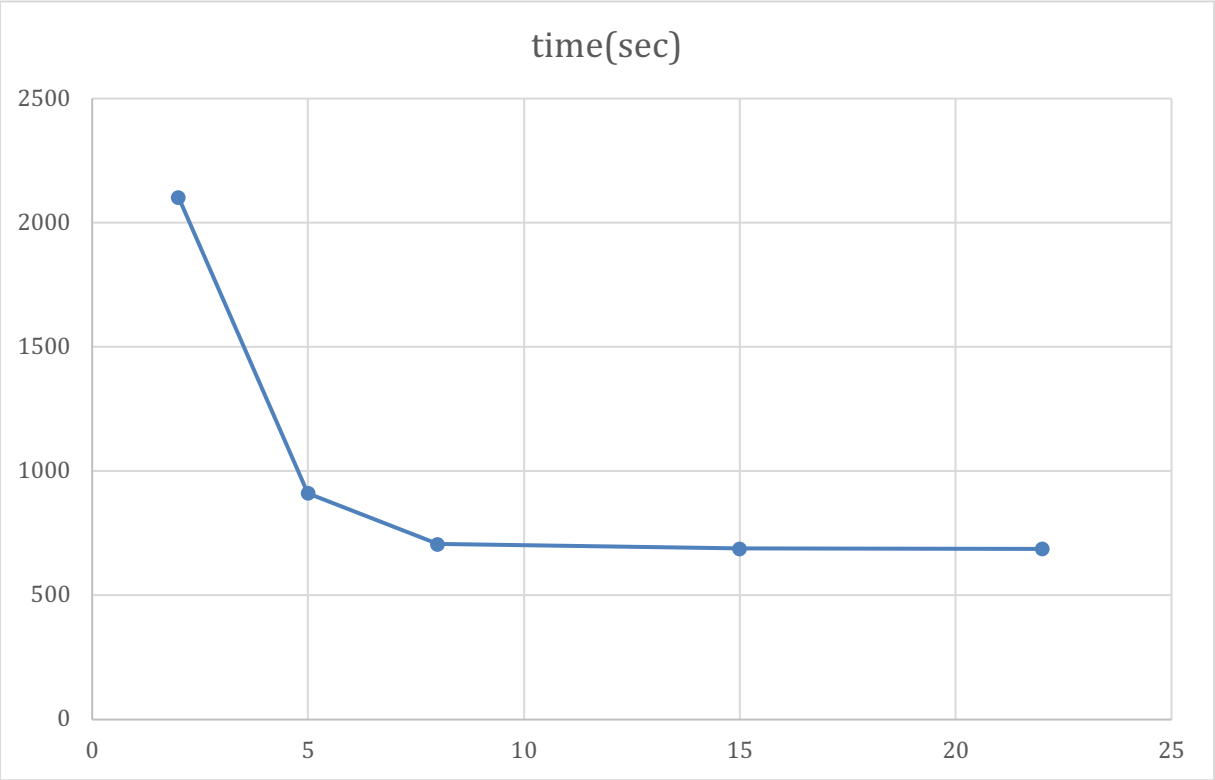
Average Taxi Time with MapReduce

- i) mapper scan the dataset, and get all the taxi-in or taxi-out which is not NULL. And combine all the data as pair [original airport, taxi out] and [dest-airport, taxi in].
- ii) reducer receive the pairs, then calculate the average of a airport, and get the a pair of [airport, taxi time] for the each valid airport.
- Iii) reducer sort all the valid airport taxi time pairs. And return the top3 and worst3 of the airport with taxi time.
- iv) if the no output, print ” No output.”.

Most Common Cancellation with MapReduce

- i) Mapper scan the dataset, get all the “cancelled” is ‘1’ and the “cancel code’ is not ‘NA’. and combine them as pair.
- ii) Reducer collect the pairs, and calculate the sum of times appear for each cancel code. Reduce the pairs from mapper to [cancel code, happen times].
- iii) Reducer sorts all the pairs of [cancel code, happen times] , and get the top1 reason of the cancelation.
- iv) if the no output, print ” No output.”.

C) execution time with the number of nodes increase

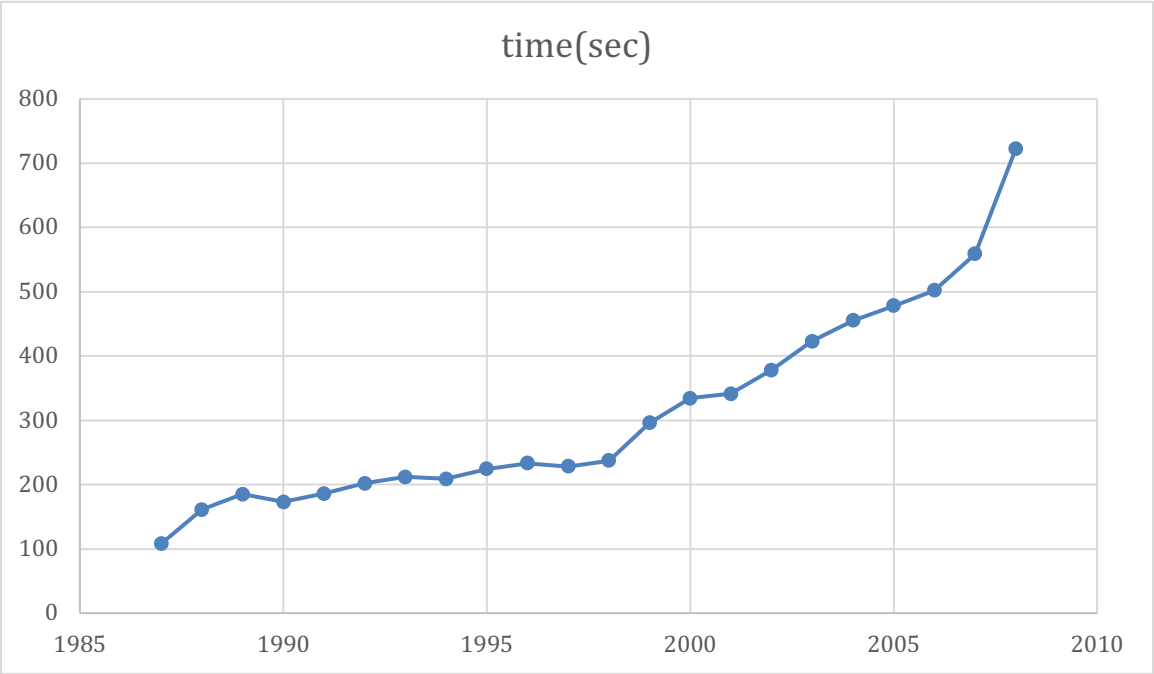


As shown in the graph, it is obvious that with the increment of the VM's number, the execution time drop significantly: from 2103 second with 2 VMs, to 687 second with 22 VMs. Because more nodes are taking part in the computing, the time for the Oozie and map-reduce is getting less.

However, as the number of the node increase, the slope of the line is getting flat. Which means, after the number of nodes reach 15, we can't get more time efficiency. Because data and computing missions transferring between the nodes needs time too, and more node means more time will be spent for the networking.

Thus, it's the reason why the number of the VMs is growing but the total time consuming couldn't be reduced.

D) time consuming with the dataset expand



According to the graph shown, with the dataset expand from 1987 to 2008, the time consuming is increasing too. As the computing power for the whole program could be considered stable, the time-consuming could stand for the scale of the dataset during typical years, and the slope between two point could stand for the increment of dataset.

So, we found that from 1987 to 1998 the increment of the flight each year is not very large. But after 1998 the slope getting bigger, which means the during 1998 to 2008, more flights are taken off than before.

Finally, we can conclude that, after 1998, more people choose plane as transportation.