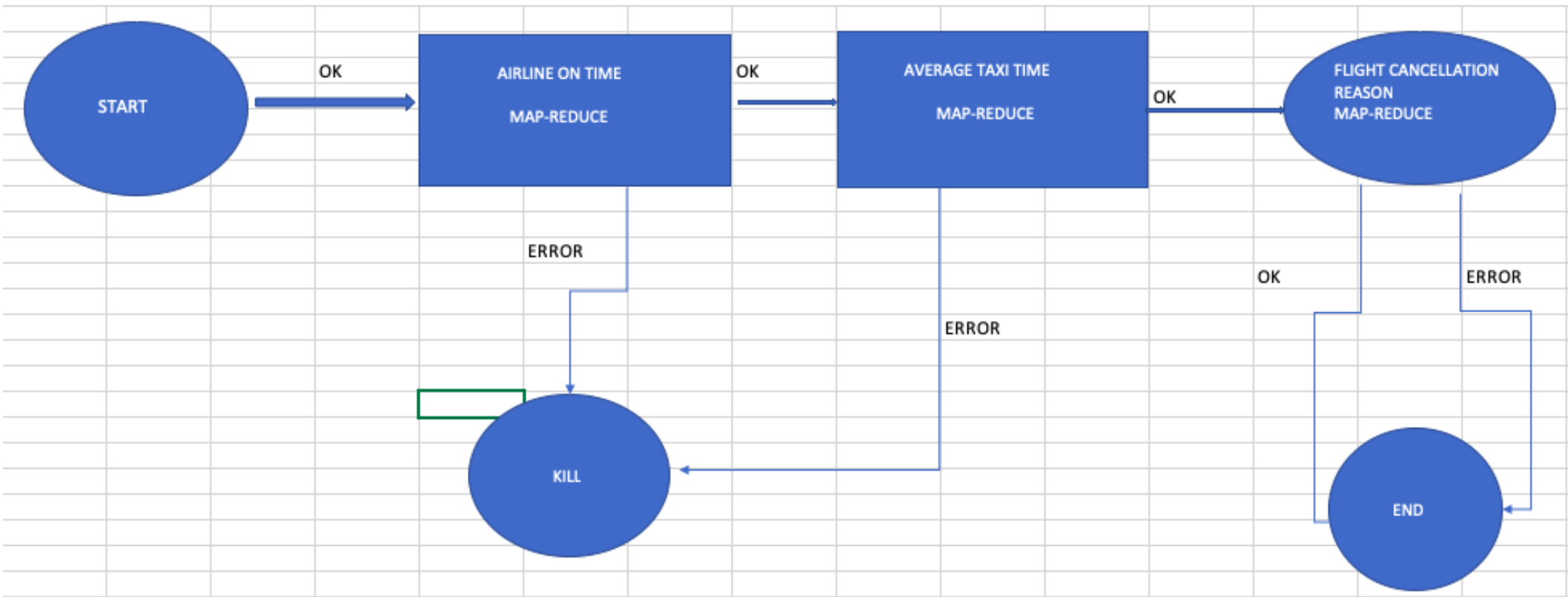
Project-CS644

Flight Data Analysis

STUDENT

Baljinder Kaur Smagh(bks46)
Suraj Kumar Jha(skj29)

(a)Structure of Oozie Workflow



(b)Algorithm—

Problem: the 3 airlines with the highest and lowest probability, respectively, for being on schedule

Program Name: AirlineOnTime.java
Formula :

column 8- UniqueCarrier unique carrier code(airline)
column 14 ArrDelay arrival delay, in minutes
column 15 DepDelay departure delay, in minutes

Delayed flights: Which has ArrDelay/DepDelay >=10 min, consider it as delay,

So delay probability = Total times delayed/ total times flight flews

Map Class:

Input : Key: some number <LongWritable>, Value: sentance <Text>
Output: Key: * Flight or Flight <Text>, Value:1 <LongWritable>

Define: A Mapper class to filter delayed flights. *Flight is for delayed flights and Flight is for counting number of alirlines flew.

Combiner Class:

Input : Key: * Flight or Flight <Text>, Value:1 <LongWritable>
Output: Key: Flight or Flight <Text>, Value: count of Flight OR Flight <LongWritable>

Define: A Combiner class to count the number of airlines delayed and number of airlines flew.

Reducer Class:

Input : Key: Flight or Flight <Text>, Value: count of Flight OR Flight <LongWritable>
Output: Key: Flight <Text>, Value: probablity <Text>

Define: Reducer Class is calculating Probablity of delayed flights and adding top 3 and least 3 flights to a tree sets of MyDataType(user Define datatype)

---x

Problem: the 3 airports with the longest and shortest average taxi time per flight (both in and out), respectively

Program Name : AverageTaxiTime.java

Formula used:

column 16- origin airport code
column 17-destination airport code
column 19-tax in a column in min for destination
column 20-taxi out the column in min for the origin

Average = total taxi In nd Out time of airport(taxi in for origin and taxi out for destination)/ number of arlines flew or landed from that particular airport

Map Class:

Input : Key: some number <LongWritable>, Value: sentance <Text>
Output: Key: Airport or Airport <Text>, Value:1 for airport count and taxi time for Airport <LongWritable>

Define: A Mapper class to filter airports and there taxi time (both in and out). * Airport is for taxi time(in or out) of airports and Airport is for number of flight airport has hosted.

Combiner Class:

Input : Key: * Airport or Airport <Text>, Value:1 <LongWritable>
Output: Key: Airport or Airport <Text>, Value: count of Airport OR Airport <LongWritable>

Define: A Combiner class to count the number of airlines flew or landed from that airport and total taxi time .

Reducer Class:

Input : Key: Airport or Airport <Text>, Value: count of Airport OR Airport <LongWritable>
Output: Key: Airport <Text>, Value: average <Text>

Define: Reducer Class is calculating Average taxi time on an airport and adding top 3 and least 3 airports to a tree sets of MyDataType(user Define datatype)

---x

Problem: the most common reason for flight cancellations.

Program Name: CancellationReason.java

Formula used:

column 21 - checks if flight canceled or not values 0 or 1 respectively
column 22 - cancellationCode - A or B or C or D

reason for cancellation
(A = carrier,
 B = weather,
 C = NAS,
 D = security)

Map Class:

Input : Key: some number <LongWritable>, Value: sentance <Text>
Output: Key: reason <Text>, Value:1 <LongWritable>

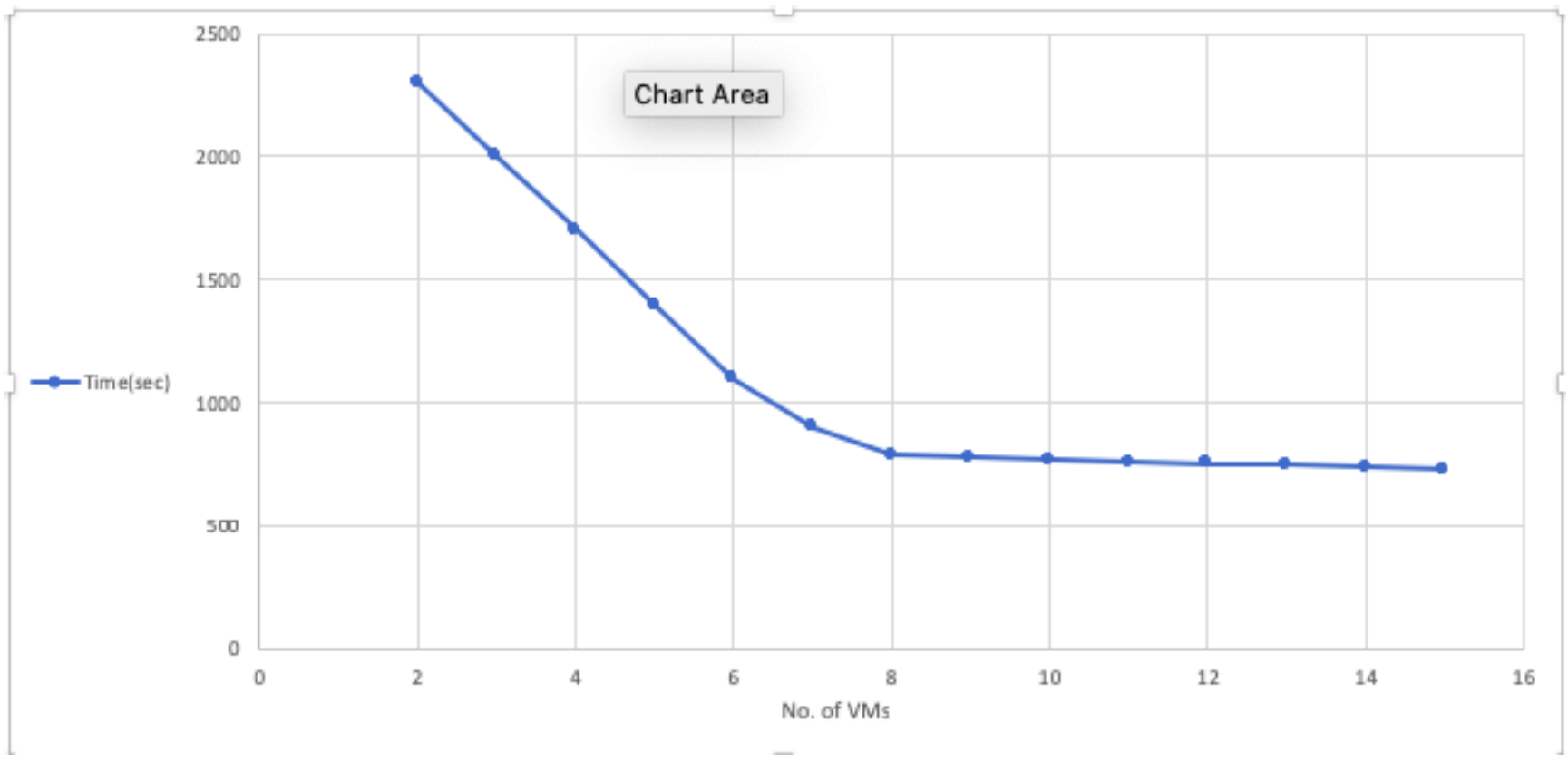
Define: A Mapper class to get different type of cancellation reason. A = carrier, B = weather, C = NAS, D = security

Reducer Class:

Input : Key: reason <Text>, Value:1 <LongWritable>
Output: Key: reason <Text>, Value: totalCount <int>

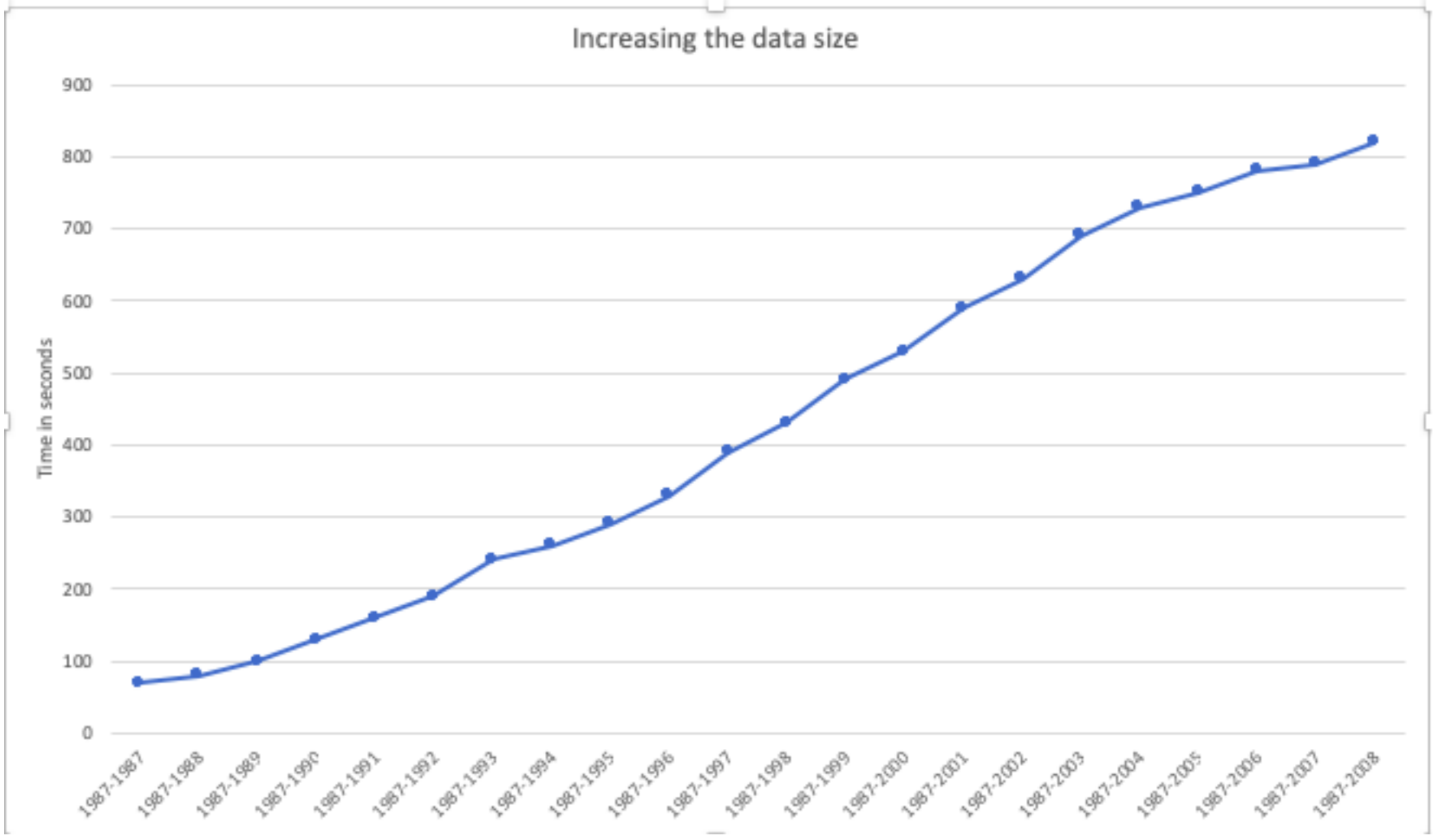
Define: Reducer Class is calculating number of flight cancelled due to following reason and then getting the most common reason.

(c)Increasing no. of VMs—



As we can observe from the graph that increasing number of VMs helps in reducing the execution time. This will also help in increasing the processing ability of Hadoop cluster and execution time of map-reduce job will also be less than before, so execution time for Oozie will also reduces. When no. of VMs reaches to some certain level, there will not be much change in execution time and it reaches to some constant level.

(d)Increasing the data Size –



As we can see from the Figure, increasing the Data size increases the execution time for Oozie workflow. In the starting the data is not much so the execution is increasing slowly but after 90s as data starts increasing at fast phase the execution time also starts increasing rapidly. But we can see there is slower raise in execution time once data reaches its ultimate level and after that increase in data not much rapidly