COSC 6339 Big Data Analytics

Python MapReduce

1st homework assignment

- **Lavanya.S.S**

  1616056

Problem description:

Given a data set containing all flights which occurred between 2006 to 2008 in US ,data related to 21 million flights are provided , each line of the data contains different information about the flight , we have to implement a map reduce job which determines the percentage of delayed flights per origin airport and we have to implement a map reduce job which determines the percentage of delayed flights per origin airport and month , also we have to determine the execution time of the code for data set using 1,2,4,8 reducers.

Solution strategy:

Solution strategy for the problem is, the complete problem is divided into mapper and reducer function where the initial operations such as finding the flight delay time and origin airport name and corresponding month is done in the mapper function , the value are presented to reducer in the ("key","value") pair format , the reducer will then use these information to further process the data and calculate the percentage of delayed flights per origin airport .

I used array data structure for storing the intermediate values,

- initially the delayed flight information is obtained by splitting the file line by line using the contxt.value.split(",") function , the variables are separated by comma to fetch each variable , the delayed time is obtained by fetching the 16^{th} column , and the origin airport is obtained by 15^{th} column, so these values are checked if they are NA values or 0,
- If the delay time is NA or 0 , the variable is assigned a value of 0
- else the variable is assigned a value of 1
- the context.emit() will transfer the key value pair, in this case key is the origin airport key, value is 1 or 0 based on whether there was a delay in flight or not, these values are transferred to the reducer function
- in the reducer function two array variables are assigned - one for calculating the total number of flights in the data , one for calculating the total number of flights that were delayed
- percentage of delayed flights is calculated by dividing the number of delayed flights by total number of flights multiplied by 100.
- As we increase the number of reducers the work gets shared and the time taken to execute the operation will decrease. We can observe the output file of each reducer to see the amount of work executed by each reducer.
- For operation percentage of delayed flights per origin airport and month the month values along with airport is taken as key and delay time is the value.

Description of how to run the code

The code is executed by using the command

pydoop submit --num-reducers 1 --upload-file-to-cache A1_mapreduce_flightdata.py A1_mapreduce_flightdata /cosc6339_hw1/flights-longlist/allflights.csv trial91

The file name is -**A1_mapreduce_flightdata.py**

The input data file is- **/cosc6339_hw1/flights-longlist/allflights.csv**

The output data is **trial91**

after execution of the program the log files will be created, we can see the list of log files as shown below,

hdfs dfs -ls trial91

-rw-r--r-- 3 bigd27 hadoop 0 2018-09-30 17:55 trial91/_SUCCESS

-rw-r--r-- 3 bigd27 hadoop 12004 2018-09-30 17:55 trial91/part-r-00000

We can see the output using the command

Hdfs dfs -cat trial91/part-r-00000

The output data files are submitted with the source code, named in the format - largefile_log_1RA.txt

Results

Description of resources used

1.Whale Cluster:

- The clusters consist of a login node (whale.cs.uh.edu) and several compute nodes
- The only access method to whale from the outside world is by using ssh (Example: Putty).
- The login nodes are to be used for editing, compiling and submitting jobs.
- Program runs are submitted through Hadoop (framework that supports the distributed execution of large scale data processing) on this cluster.
- Jobs are submitted from the login node and run on 1 or more compute nodes. Jobs then run until they terminate in some way, e.g. normal completion, timeout, abort.

2. Pydoop:

- It is a Python interface to Hadoop that allows to write MapReduce applications in Python.
- Pydoop offers several features not commonly found in other Python libraries for Hadoop:
    - o a rich HDFS API
    - o a MapReduce API that allows to write pure Python record readers / writers, partitioners and combiners
    - o transparent Avro (de)serialization
    - o easy installation-free usage

3. Python: version 2.7

- Python is an easy to learn, powerful programming language.
- It has efficient high-level data structures and a simple but effective approach to object oriented programming.
- Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.
- The Python interpreter is easily extended with new functions and data types implemented in C or C++.
- Python is also suitable as an extension language for customizable application

Description of measurements performed

Operation - percentage of delayed flights per origin airport

Sl no	Operation	Number of reducers	Time taken	Log file name [attached with src code file]
1	percentage of delayed flights per origin airport	1	4 min 10 seconds	Largefile_log_1RA.txt
2	percentage of delayed flights per origin airport	2	2 min 47seconds	Largefile_log_2RA.txt
3	percentage of delayed flights per origin airport	4	2 min 15 seconds	Largefile_log_4RA.txt
4	percentage of delayed flights per origin airport	8	1 min 40 seconds	Largefile_log_8RA.txt

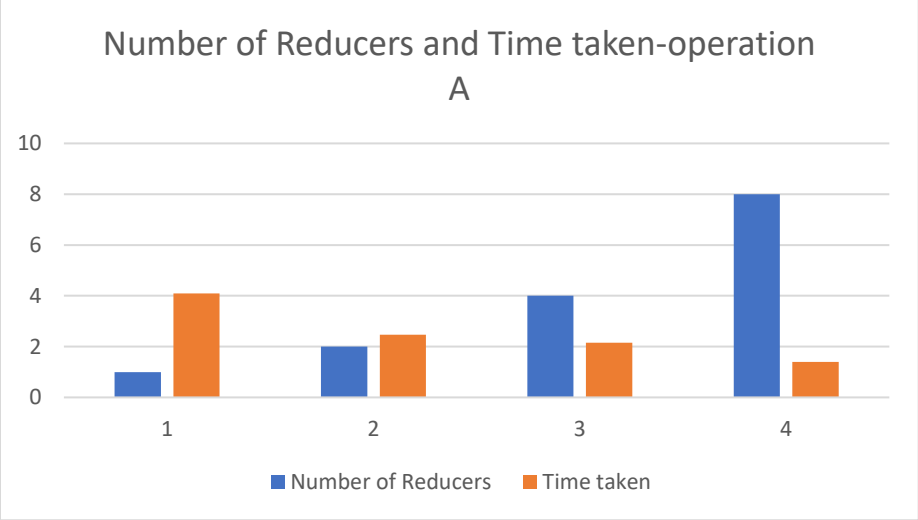
Operation- percentage of delayed flights per origin airport and month

Sl no	Operation	Number of reducers	Time taken	Log file name
1	percentage of delayed flights per origin airport and month	1	4 min 9 seconds	Largefile_log_1RB.txt

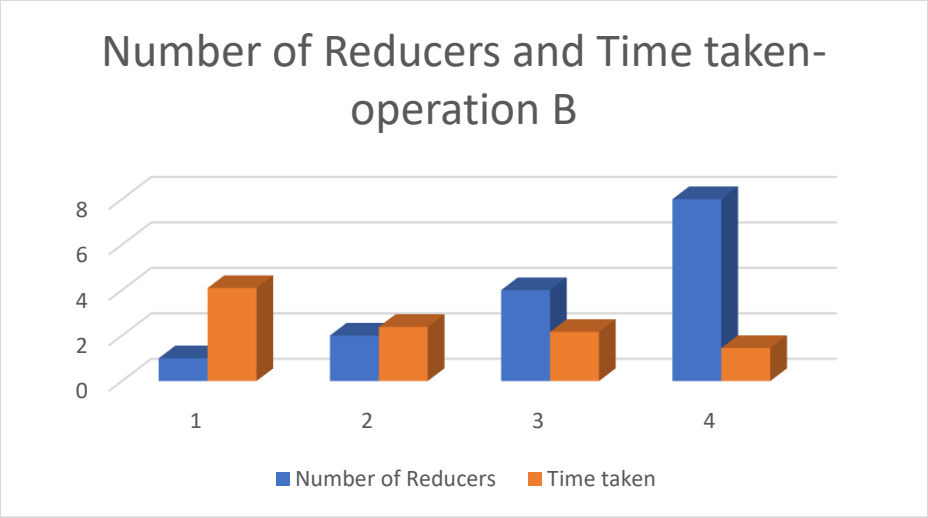
2	percentage of delayed flights per origin airport and month	2	2 min 38 seconds	Largefile_log_2RB.txt
3	percentage of delayed flights per origin airport and month	4	2 min 16 Seconds	Largefile_log_4RB.txt
4	percentage of delayed flights per origin airport and month	8	1 min 45 seconds	Largefile_log_8RB.txt

Results

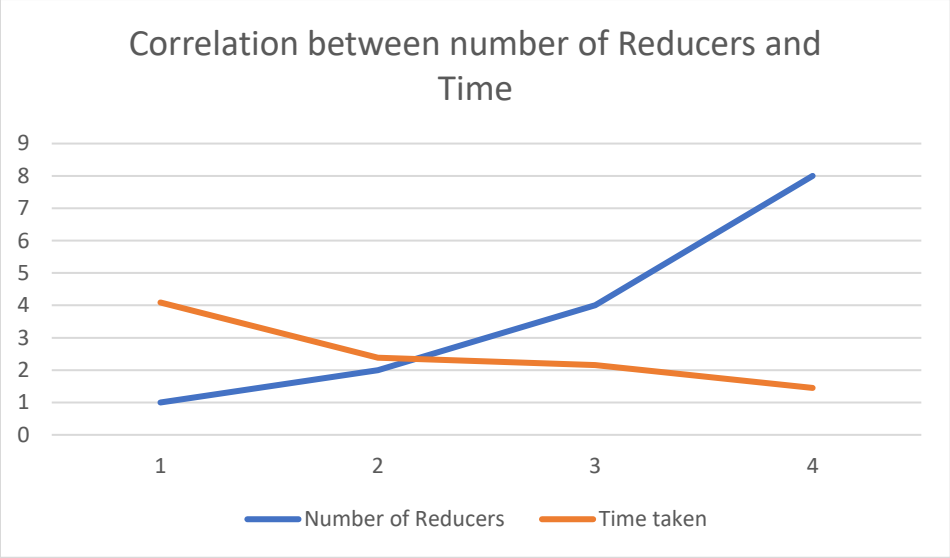
1. percentage of delayed flights per origin airport



2. percentage of delayed flights per origin airport and month



3. Correlation between number of Reducers and Time



As we can see from the above results, as the number of Reducers increases the amount of time taken reduces as the work gets shared between different reducers , so more the number of reducers faster is the operation .