IPL DATA ANALYSIS

A MINI PROJECT REPORT SUBMITTED BY

Roystan Dsouza
4NM17CS151
VI Semester, C/D Section

Rohith Naik
4NM17CS149
VI Semester, C/D section

UNDER THE GUIDANCE OF

Mrs. Divya Jennifer D'souza
Assistant professor GD I
Department of Computer Science and Engineering

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF

Bachelor of Engineering in Computer Science & Engineering

from

Visvesvaraya Technological University, Belagavi

N.M.A.M. INSTITUTE OF TECHNOLOGY
(An Autonomous Institution under VTU, Belgaum)
AICTE approved, (ISO 9001:2015 Certified), Accredited with 'A' Grade by NAAC
NITTE -574 110, Udupi District, KARNATAKA.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

B.E. CSE Program Accredited by NBA, New Delhi from 1-7-2018 to 30-6-2021

April 2020

NITTE EDUCATION TRUST | N.M.A.M. INSTITUTE OF TECHNOLOGY
(An Autonomous Institution affiliated to Visvesvaraya Technological University, Belagavi)
Nitte – 574 110, Karnataka, India
(ISO 9001:2015 Certified), Accredited with 'A' Grade by NAAC
☎: 08258 - 281039 – 281263, Fax: 08258 – 281265

Department of Computer Science and Engineering
B.E. CSE Program Accredited by NBA, New Delhi from 1-7-2018 to 30-6-2021

CERTIFICATE

"IPL DATA ANALYSIS" is a bonafide work carried out by Roystan Dsouza (4NM17CS151) and Rohith Naik(4NM17CS149) in partial fulfilment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering prescribed by Visvesvaraya Technological University, Belagavi during the year 2019-2020.

It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report. The Mini project report has been approved as it satisfies the academic requirements in respect of the project work prescribed for the Bachelor of Engineering Degree.

Signature of Guide

Signature of HOD

ACKNOWLEDGEMENT

We believe that our project will be complete only after we thank the people who have contributed to make this project successful.

First and foremost, our sincere thanks to our beloved principal, Dr. Niranjan N. Chiplunkar for giving us an opportunity to carry out our project work at our college and providing us with all the needed facilities.

We sincerely thank Dr. K.R. Udaya Kumar Reddy, Head of Department of Computer Science and Engineering, Nitte Mahalinga Adyantaya Memorial Institute of Technology, Nitte.

We express our deep sense of gratitude and indebtedness to our guide Mrs. Divya Jennifer D'Souza, Assistant Professor GD I, Department of Computer Science and Engineering, for her inspiring guidance, constant encouragement, support and suggestions for improvement during the course of our project.

We thank all the teaching and non-teaching staff members of the Computer Science and Engineering Department and our parents and friends for their honest opinions and suggestions throughout the course of our project.

Finally, we thank all those who have supported us directly or indirectly throughout the project and making it a grand success.

Roystan Dsouza
(4NM17CS151)

Rohith Naik
(4NM17CS149)

ABSTRACT

In today's date big data analysis is needed for every data analytics to examine sets of data to extract the useful information from it and to draw conclusion according to the information. Analyzing Big Data is a very challenging task since it involves the processing of huge amount of data. Prominent data analysis tools are required to analyze the data in order to gain value out of it. Hadoop is a sought -after open source framework that uses MapReduce techniques to store and Process huge datasets. However, the programs written using MapReduce techniques are not flexible and also require maintenance. This problem is overcome by making use of HiveQL. In order to execute queries in HiveQL, the platform required is Hive.

Here we are considering the dataset of Indian Premier League(IPL) and analysing it using HiveQL.

Table of Contents

INTRODUCTION

Big data is a well known term used to portray the exponential development, openness and availability of data, both unstructured and structured. As and when the data generated, it has to be processed and analyzed in order to extract meaningful information from the data, else the data generated is useless. Since more than 80% of the data produced is unstructured, traditional database systems fail to analyze such kind of data. Big Data analytics overcomes this disadvantage and helps in processing and analyzing such unstructured or semi-structured data in a short amount of time. Built-in tools are used for this purpose. Big Data Analytics internally uses Hadoop framework for processing the data. Hadoop framework consists of MapReduce and Hadoop Distributed File System (HDFS). MapReduce framework provides a way of processing the data in a distributed manner and HDFS provides a way to store the huge amount of data. Hadoop framework also provides various tools like Hive, Pig, Cassandra etc for Big Data analytics.

The Indian Premier League(IPL) is a Twenty-20 cricket tournament league established with objective of promoting cricket in India. The league is annual event where representing different India cities compete with each other. The teams for IPL are selected through auction. Big data creates a new way in the field of cricket to analyse the numerous amount of data generated in every match of cricket.

Here we have used Hive tool with the Hadoop Framework in order to process the Indian Premier League (IPL) dataset. HiveQL is used to write SQL like queries to process the data. These queries are internally converted to MapReduce operations to process and analyze the data. SQL is mostly suited for smaller datasets whereas HiveQL is most suited for larger datasets.

SOFTWARE REQUIREMENTS

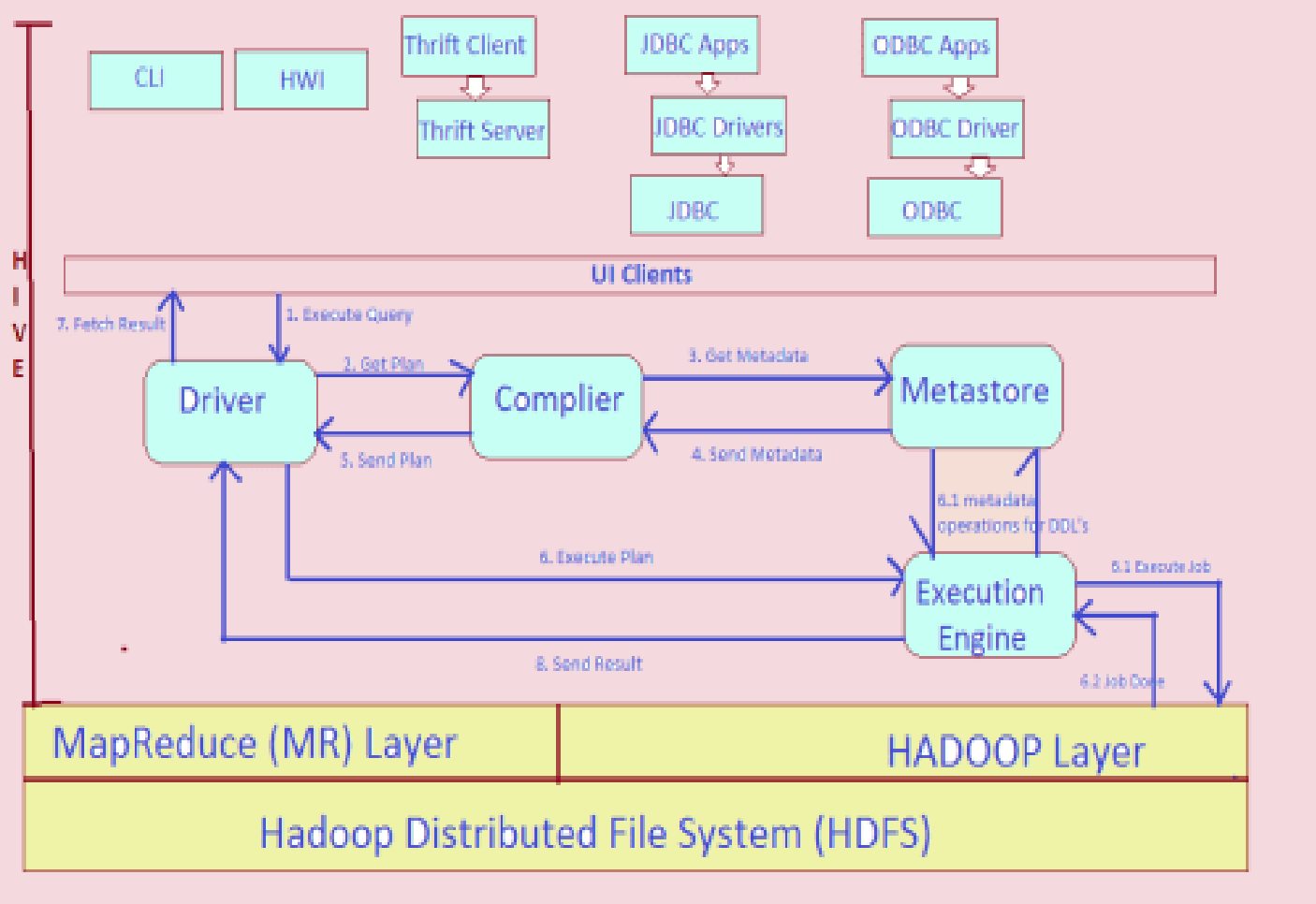
- Apache Hadoop: Apache Hadoop is a collection of open-source software utilities that facilitate using a network of many computers to solve problems involving massive amounts of data and computation. It provides a software framework for distributed storage and processing of big data using the MapReduce programming model.

- Apache Hive: Apache Hive is a data warehouse software project built on top of Apache Hadoop for providing data query and analysis. Hive gives a SQL-like interface to query data stored in various databases and file systems that integrate with Hadoop.

HARDWARE REQUIREMENTS

- Minimum RAM required: 4GB (Suggested: 8GB)
- Minimum Free Disk Space: 25GB
- Minimum Processor i3 or above
- Operating System of 64bit (Suggested)

DESIGN AND ANALYSIS



IMPLEMENTATION

Creating table 'iplmatches' :

CREATE TABLE IPLMATCHES (ID INT, SEASON STRING, CITY STRING, DATE_OF_MATCH DATE, TEAM1 STRING, TEAM2 STRING, TOSS_WINNER STRING, TOSS_DECISION STRING, RESULT STRING, DL_APPLIED INT, WINNER STRING, WIN_BY_RUNS INT, WIN_BY_WICKETS INT, PLAYER_OF_MATCH STRING, VENUE STRING, UMPIRE1 STRING, UMPIRE2 STRING, UMPIRE3 STRING) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n';

Loading the data from hdfs to 'iplmatches' :

LOAD DATA INPATH '/user/hduser_/bdaproj/matches.csv;

HiveQL queries:

1.To determine the total matches played since the inception of IPL :

```
hive> select count(id) from iplmatches;
```

2.To determine the winner of each season of the IPL :

```
hive> SELECT
    >   season  AS SEASON,
    >   winner AS WINNER,
    >   IF(win_by_runs='0',CONCAT('Won by ',win_by_wickets,' Wickets'),
    >   CONCAT('Won by ',win_by_runs,' Runs')) AS `WINNING MARGIN`,
    >   IF(winner=team1,team2,team1) AS RUNNER,
    >   player_of_match
    > FROM
    >   iplmatches
    > WHERE
    >   id IN(SELECT MAX(id) FROM iplmatches GROUP BY season) ;
```

3.To determine the winner of the 1st match of every season of the IPL :

```
hive> SELECT
    >   season  AS SEASON,
    >   winner AS WINNER,
    >   IF(win_by_runs='0',CONCAT('Won by ',win_by_wickets,' Wickets'),
    >   CONCAT('Won by ',win_by_runs,' Runs')) AS `WINNING MARGIN`,
    >   IF(winner=team1,team2,team1) AS RUNNER,
    >   player_of_match
    > FROM
    >   iplmatches
    > WHERE
    >   id IN(SELECT MIN(id) FROM iplmatches GROUP BY season) ;
```

4.To determine the information of the tied matches in IPL :

```
hive> select * from iplmatches where result='tie';
```

RESULT

1.Result of the 1st query :

```
hive> select count(id) from iplmatches;
Query ID = hduser__20200516002533_277969d8-0802-46e5-8d49-a2d2e8ab2d88
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Job running in-process (local Hadoop)
2020-05-16 00:25:36,635 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_local1791294298_0001
MapReduce Jobs Launched:
Stage-Stage-1:  HDFS Read: 280226 HDFS Write: 0 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
756
Time taken: 3.421 seconds, Fetched: 1 row(s)
```

2.Result of the 2nd query :

```
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Job running in-process (local Hadoop)
2020-05-16 00:27:18,890 Stage-2 map = 100%,  reduce = 100%
Ended Job = job_local1645811991_0004
Stage-5 is selected by condition resolver.
Stage-1 is filtered out by condition resolver.
SLF4J: Found binding in [jar:file:/home/roystan/Downloads/apache-hive-3.1.2-bin/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/roystan/Downloads/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
2020-05-16 00:27:22     Starting to launch local task to process map join;      maximum memory = 239075328
2020-05-16 00:27:22     Uploaded 1 File to: file:/tmp/hduser_/0cb13293-6118-4ee6-9afe-1f05f9d037cf/hive_2020-05-16_00-27-17_618_705454888247314500-1/-local-10005/Has
hTable-Stage-3/MapJoin-mapfile11--.hashtable (494 bytes)
Execution completed successfully
MapredLocal task succeeded
Launching Job 3 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Job running in-process (local Hadoop)
2020-05-16 00:27:24,075 Stage-3 map = 100%,  reduce = 0%
Ended Job = job_local2092880290_0005
MapReduce Jobs Launched:
Stage-Stage-2:  HDFS Read: 1120904 HDFS Write: 0 SUCCESS
Stage-Stage-3:  HDFS Read: 700565 HDFS Write: 0 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
2017    Mumbai Indians  Won by 1 Runs   Rising Pune Supergiant  KH Pandya
2008    Rajasthan Royals        Won by 3 Wickets        Chennai Super Kings     YK Pathan
2009    Deccan Chargers Won by 6 Runs   Royal Challengers Bangalore     A Kumble
2010    Chennai Super Kings     Won by 22 Runs  Mumbai Indians  SK Raina
2011    Chennai Super Kings     Won by 58 Runs  Royal Challengers Bangalore     M Vijay
2012    Kolkata Knight Riders   Won by 5 Wickets        Chennai Super Kings     MS Bisla
2013    Mumbai Indians  Won by 23 Runs  Chennai Super Kings     KA Pollard
2014    Kolkata Knight Riders   Won by 3 Wickets        Kings XI Punjab MK Pandey
2015    Mumbai Indians  Won by 41 Runs  Chennai Super Kings     RG Sharma
2016    Sunrisers Hyderabad     Won by 8 Runs   Royal Challengers Bangalore     BCJ Cutting
2018    Chennai Super Kings     Won by 8 Wickets        Sunrisers Hyderabad     SR Watson
2019    Mumbai Indians  Won by 1 Runs   Chennai Super Kings     JJ Bumrah
Time taken: 6.47 seconds, Fetched: 12 row(s)
```

3.Result of the 3rd query :

```
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Job running in-process (local Hadoop)
2020-05-16 00:28:00,875 Stage-2 map = 100%,  reduce = 100%
Ended Job = job_local1779125889_0006
Stage-5 is selected by condition resolver.
Stage-1 is filtered out by condition resolver.
SLF4J: Found binding in [jar:file:/home/roystan/Downloads/apache-hive-3.1.2-bin/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]

SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
2020-05-16 00:28:04     Starting to launch local task to process map join;      maximum memory = 239075328
2020-05-16 00:28:04     Uploaded 1 File to: file:/tmp/hduser_/0cb13293-6118-4ee6-9afe-1f05f9d037cf/hive_2020-05-16_00-27-59_618_3703585785245895706-1/-local-10005/Ha
shTable-Stage-3/MapJoin-mapfile21--.hashtable (492 bytes)
Execution completed successfully
MapredLocal task succeeded
Launching Job 3 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Job running in-process (local Hadoop)
2020-05-16 00:28:06,024 Stage-3 map = 100%,  reduce = 0%
Ended Job = job_local300853537_0007
MapReduce Jobs Launched:
Stage-Stage-2:  HDFS Read: 1681356 HDFS Write: 0 SUCCESS
Stage-Stage-3:  HDFS Read: 980791 HDFS Write: 0 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
2017    Sunrisers Hyderabad     Won by 35 Runs  Royal Challengers Bangalore     Yuvraj Singh
2008    Kolkata Knight Riders   Won by 140 Runs Royal Challengers Bangalore     BB McCullum
2009    Mumbai Indians  Won by 19 Runs  Chennai Super Kings     SR Tendulkar
2010    Kolkata Knight Riders   Won by 11 Runs  Deccan Chargers AD Mathews
2011    Chennai Super Kings     Won by 2 Runs   Kolkata Knight Riders   S Anirudha
2012    Mumbai Indians  Won by 8 Wickets        Chennai Super Kings     RE Levi
2013    Kolkata Knight Riders   Won by 6 Wickets        Delhi Daredevils        SP Narine
2014    Kolkata Knight Riders   Won by 41 Runs  Mumbai Indians  JH Kallis
2015    Kolkata Knight Riders   Won by 7 Wickets        Mumbai Indians  M Morkel
2016    Rising Pune Supergiants Won by 9 Wickets        Mumbai Indians  AM Rahane
2018    Chennai Super Kings     Won by 1 Wickets        Mumbai Indians  DJ Bravo
2019    Chennai Super Kings     Won by 7 Wickets        Royal Challengers Bangalore     Harbhajan Singh
Time taken: 6.419 seconds, Fetched: 12 row(s)
```

4.Result of the 4th query :

```
hive> select * from iplmatches where result='tie';
OK
34      2017    Rajkot  2017-04-29      Gujarat Lions   Mumbai Indians  Gujarat Lions   bat     tie     0       Mumbai Indians  0       0       KH Pandya       Saura
shtra Cricket Association Stadium       AK Chaudhary    CB Gaffaney
126     2009    Cape Town       2009-04-23      Rajasthan Royals        Kolkata Knight Riders   Kolkata Knight Riders   field   tie     0       Rajasthan Royals    0
0       YK Pathan       Newlands        MR Benson       M Erasmus
190     2010    Chennai 2010-03-21      Kings XI Punjab Chennai Super Kings     Chennai Super Kings     field   tie     0       Kings XI Punjab 0       0       J The
ron     "MA Chidambaram Stadium  Chepauk"       K Hariharan     DJ Harper
388     2013    Hyderabad       2013-04-07      Royal Challengers Bangalore     Sunrisers Hyderabad     Royal Challengers Bangalore     bat     tie     0       Sunri
sers Hyderabad  0       0       GH Vihari       "Rajiv Gandhi International Stadium     Uppal" AK Chaudhary    S Ravi
401     2013    Bangalore       2013-04-16      Delhi Daredevils        Royal Challengers Bangalore     Royal Challengers Bangalore     field   tie     0       Royal
 Challengers Bangalore  0       0       V Kohli M Chinnaswamy Stadium   M Erasmus       VA Kulkarni
476     2014    Abu Dhabi       2014-04-29      Rajasthan Royals        Kolkata Knight Riders   Rajasthan Royals        bat     tie     0       Rajasthan Royals    0
0       JP Faulkner     Sheikh Zayed Stadium    Aleem Dar       AK Chaudhary
536     2015    Ahmedabad       2015-04-21      Rajasthan Royals        Kings XI Punjab Kings XI Punjab field   tie     0       Kings XI Punjab 0       0       SE Ma
rsh     "Sardar Patel Stadium    Motera"        M Erasmus       S Ravi
11146   2019    Delhi   NULL    Kolkata Knight Riders   Delhi Capitals  Delhi Capitals  field   tie     0       Delhi Capitals  0       0       P Shaw  Feroz Shah Ko
tla Ground      Anil Dandekar   Nitin Menon     Marais Erasmus
11342   2019    Mumbai  NULL    Mumbai Indians  Sunrisers Hyderabad     Mumbai Indians  bat     tie     0       Mumbai Indians  0       0       JJ Bumrah       Wankh
ede Stadium     S Ravi  O Nandan        Nanda Kishore
Time taken: 0.147 seconds, Fetched: 9 row(s)
```

CONCLUSION

Analyzing Big Data is a very challenging task since it involves the processing of huge amount of data. Hadoop-Hive is well suited for the analysis of huge datasets. The performance of Hadoop-Hive in case of smaller datasets is low but it increases as the size of the dataset grows. However, SQL is well suited for smaller datasets.

This work analyzed IPL dataset using HiveQL for four different cases. The first case the total matches played since the inception of IPL. The second case determined the winner of each season of the IPL. The third case determined the winner of the 1st match of every season of the IPL. The fourth case determined the the information of the tied matches in IPL.

As a part of future work, further queries can be written to retrieve other meaningful information like team statistics, player consistency, strike rate and other parameters which can be of use to the bidders in careful selection of their IPL teams.

REFERENCES

1. Indian Premier League Dataset Analytics using Hadoop-Hive(International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-9 Issue-2, December, 2019)

2. https://github.com/thabaresh-s/cricket-data-analysis-using-Hive-and-Pig