Faster The Slow Running RDBMS Queries With Spark Framework

Hariteja Bodepudi
Irving, USA
hariteja.bodepudi1990@gmail.com

DOI: 10.29322/IJSRP.10.11.2020.p10735
http://dx.doi.org/10.29322/IJSRP.10.11.2020.p10735

Abstract: Data is increasing day by day due to increase of the advanced internet technology, online browsing, Internet banking and online shopping. Modern Technology has helped the humankind to access and communicate anywhere in world very easily. This advanced technology drives the increase of data day by day.

Most of the companies traditionally maintain the transactional data in the Relational Data Base systems like Oracle, SQL Server, MYSQL etc. Every Organisation will have the daily reporting, weekly reporting, and monthly reporting on the collected data. Due to increase in the volumes of data from Terabytes to Petabytes, the processing of the complex queries for reporting in the RDBMS was really time taking and slow.

The usage of Spark to process these reporting complex queries will be 10 times faster than the actual processing of the data in the RDBMS. In General , the RDBMS uses the single node of the cluster to process the query and it is really time taking to process the complex queries used for daily ,weekly and monthly reporting as it got complex joins and multiple aggregate functions involved in it. But spark leverage all the nodes in the clusters to process the data very quickly by using the partitions in the table to break into small chunks and achieve the level of Parallelism to process 10 times faster than the RDBMS.

This Paper talks about how the RDBMS reporting scripts performance can be improved by incorporating the Spark framework without changing the existing queries in the RDBMS.

Index Terms- **RDBMS*;HDFS;Spark***

I. INTRODUCTION

Reporting of the data in daily basis, Weekly, Monthly, and yearly is very crucial for any organization to make the key decisions. Reporting is key for analysis of the data.

Data Reporting is a tool which is used to analyse the current, past and the future prediction for the business decisions. This data used for reporting helps in making key decisions in both operational and strategical [1]. Data Reporting will help the management to take the past data and forecast the future trends and to take the decisions how well the business is operating.

Data is increasing a lot every day due to usage of IOT devices and internet. Traditionally transactional data for any organisation is maintained and stored in the Relational Data Base Systems. RDBMS databases are stable and well established and easily to handle the data in efficient way. Data in RDBMS are consistent and reliable so most of the organisation in different sectors like banking, retail and health industry prefer RDBMS for transactional data storage. [2] Due to increase of the data it becomes hard for the RDBMS to run the queries required for daily, weekly, monthly, and yearly reporting.

Reporting is key for any organisation to make the managerial decisions whether a particular product or service is making profits or to check how it is functioning. These queries run for long time as the RDBMS runs the query on single node on the cluster. If these jobs get failed due to heavy data and this will impact the business operations and impact the key decisions to be taken for effective functionality by management. This Paper highlights the importance of reporting and comes with solution how to make these long running queries to run quickly by making use of spark without any

modifications in scripts or compromising the volumes of the data. Spark is a framework which process the data by in memory process by using the multiple nodes in the cluster by parallel processing to run the queries quickly. Spark [3] process these reporting queries 10 to 100 times faster than Relational Database Management Systems with the inbuilt factors of Parallelism and fault tolerance.

Relational Database System (RDBMS) [4] is a database that stores the data in the structured format in form of rows and columns. This helps to locate the specified data or values required in the database. The data in the database are organised in the tables It searches the data in the database by querying.

Hadoop Distributed File System (HDFS) [5]is a distributed file system that handles large volumes of data on the commodity hardware. It is highly fault tolerant and support the applications that have large datasets. It Supports both Streaming and batch data.

Spark [6] is a data processing engine which supports large volumes of data. The processing of the data through spark is 100 times faster than traditional system processing as It does the in-memory computation. It uses the built-in features like Parallelism and Fault-tolerant to scales distributed large data workload across the multiple nodes on the large cluster.

II. PROBLEM STATEMENT

A. *Existing System:*

RDBMS is capable to process the small volumes of the data very quickly. But if there is large volumes of data to be processed for reporting in daily ,weekly, Monthly and yearly basis the queries required to pull these data from database and processing will take long time and sometimes it might break as RDBMS uses single node in a cluster to process the data . As the RDBMS does not process the data in parallel and it will take long time to execute the SQL queries and get the required report. Queries need to be optimised and need lot of manual effort to optimise the scripts and need DBA help to limit the jobs to reduce the CPU usage in the server.

B. *Proposed System:*

Existing System issues can be overcome by making using of spark framework to process large volume of data without any changes in the existing SQL scripts. This Paper will explain in clear how it can be achieved and explains the implementation.

C. *Spark Architecture:*

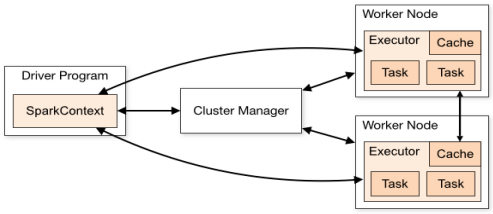

Figure1: Spark Architecture [7]

Spark is a framework which is used to process and analyse the large volumes of datasets with the distributed computing system. Spark implements the architecture of master and slave which uses one master and multiple worker nodes. Spark driver will go through its own java process and it also controls the workers machines i.e. executors have its own process.

From the Figure1 it is clear spark applications are launched on the multiple machines by using the cluster manager. Cluster Manger can be standalone or the YARN, Mesos, Kubernetes etc and these are managed by the Resource Manager [8].

D. *Functionality of the Spark Framework:*

A spark driver launches the job, and it coordinates between the Master and the Executors. It invokes the main applications and launches the spark context. Spark driver has DAG scheduler, task scheduler, backend scheduler and block manager.

When a job is submitted to the Spark for processing spark context is created by the spark driver and it request for the resources from the cluster manager and it launches the executors. Spark driver runs the main application and the actual codes and split the tasks among the executors. Executors will run the tasks and sends the output back to the driver [9].

III. IMPLEMENTATION

In this Paper I am taking the implementation of running the MYSQL queries with Spark. This can be extended to any RDBMS databases.

Spark can be connected to any RDBMS database by using the required database jdbc driver. Through that jdbc driver spark can connect to the database and can start running the queries on top of it.

A. Connecting Spark to MYSQL Database:

To connect Spark Engine to the MYSQL database mysql-connector jdbc driver [10] need to be installed either on the hdfs i.e. Hadoop distributed file system or on the spark to call from the spark shell.

The below code needs to be executed to connect spark to MYSQL

```
Spark-shell –jars location of the jar
```

B. Configuring the Connection String to connect to MYSQL:

After connecting to the MYSQL with spark shell. Username and password of the database needs to authenticate to connect to a specific database to read the table from the database.

Below code snippet needs to be executed to read the table from the MYSQL database

```
val url = “url of the mysql”

val prop = new java.util.Properties

prop.setProperty(“username”,”username of MYSQL “)

prop.setProperty(“password”,”password of MYSQL “)

prop.setProperty(“driver”,” com.mysql.jdbc.Driver”)
```

C. Reading the MYSQL Table and Storing data in Database:

By using the above configuration string, we can connect to the specific database and table in the MYSQL and the data can be stored in the memory temporarily.

Below code snippet needs to be executed to read and store the data in data frame

df in the code is the dataframes which saves the MYSQL data in the memory

```
val df = spark.read.format(“jdbc”).option(url,table,prop)
```

D. Creating temporary table on Data Frame:

We can register the dataframes as temporary table or view and can perform the SQLK operation on top of it.

Below is the code snippet to connect to the create temporary table

```
df.createTempView(“temptablename”)
```

We can do any SQL operations on top of this temporary table which we used to run in the MYSQL databases.

Below is the sample code snippet how to run the SQL queries on top of the temp view/table

```
val result = spark.sql("select count(*) from temptable")
```

These processed query result can be written back to the MYSQL database or even can be save as any type of file format like csv, excel etc.

Writing Result back to MYSQL data frame:

```
val table =" dbname.tablename"
result.write.mode("append").jdbc(url,tablename,
```

[11]

IV. FINDINGS

The processing of the same query in the MYSQL i.e. RDBMS is slower when compared with the spark processing. If the query takes minutes to process the data in RDBMS it will be completed in seconds with the spark due to the in-memory computation

V. CONCLUSION

This paper talks about the research I performed how the performance of the long running queries can be improved in RDBMS by using the Spark Framework. Reporting is a key factor for any organisation and most of the transactional data is stored in the RDBMS but RDBMS takes long time to process the long reporting data so the proposed solution of connecting spark to the RDBMS database and then process the same queries without any chances will solve the problem of data delay and slow processing data .

This Paper clearly explains the implementation with the code which can be used to connect any RDBMS database to improve the performance of processing the data.

REFERENCES

[1] S. Durcevic, "What is a Data Report," Oct 2019. [Online]. Available: https://www.datapine.com/blog/data-report-examples/.

[2] S. D. Thomson, "Preserving Transactional Data," Digital Preservation Coalition , 2016.

[3] "Apache Spark," June 2020. [Online]. Available: https://www.ibm.com/cloud/learn/apache-spark?lnk=hm. [Accessed October 2020].

[4] "RDBMS Definition," [Online]. Available: https://techterms.com/definition/rdbms#:~:text=Stands%20for%20%22Relational%20Database%20Management,format%2C%20using%20rows%20and%20columns..

[5] "Hadoop Introduction," [Online]. Available: https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html.

[6] "Apache Spark," [Online]. Available: https://www.ibm.com/cloud/learn/apache-spark?lnk=hm.

[7] "Spark Components," [Online]. Available: https://spark.apache.org/docs/latest/cluster-overview.html.

[8] "Cluster Overview," [Online]. Available: https://spark.apache.org/docs/latest/cluster-overview.html.

[9] P. Nayak, "Spark Architecture and Internal Working," [Online]. Available: https://medium.com/@psnayak90/spark-architecture-internal-working-few-insights-on-internal-working-of-spark-a357a5248367.

[10] "JDBC Driver for MySQL (Connector/J)," [Online]. Available: https://www.mysql.com/products/connector/.

[11] "JDBC to Other Databases Using Spark," [Online]. Available: https://spark.apache.org/docs/latest/sql-data-sources-jdbc.html.

AUTHORS

Author - Hariteja Bodepudi, Masters in Information Systems,
hariteja.bodepudi1990@gmail.com