International Journal of Computer Trends and Technology
ISSN: 2231 – 2803 / https://doi.org/10.14445/22312803/IJCTT-V69I3P103

Volume 69 Issue 3, 10-13, March, 2021

Original Article

Data Transfer Between RDBMS and HDFS by using the Spark Framework in Sqoop for Better Performance

Hariteja Bodepudi
Irving, USA

Received Date: 18 January 2021
Revised Date: 05 March 2021
Accepted Date: 08 March 2021

Abstract - *The Usage of the Internet and IOT devices has increased a lot these days. This results in an increase of the data day by day. Data has been increased from Terabytes to Petabytes which Traditional database systems cannot store and process. This data is often referred to as Big Data.*

This Big Data needs a big storage capacity which becomes more expensive to store. Companies need low commodity hardware and high reliability, and less expensive, which can be achieved and handled by the Hadoop Framework.

Organizations started to move across to the Hadoop Ecosystem to store and process large volumes of data to gain more insights out of the data. Traditionally data was stored in the RDBMS, i.e., Relational Database Management Systems. To move this data into the Hadoop Ecosystem, a tool called Sqoop become more prominent to both import and export the data from the RDBMS to Hadoop and the Hadoop to RDBMS.

This paper is going to address the importance of Sqoop and the functionality of the Sqoop how it handles the large data sets and is used as ETL to transfer the data from RDBMS to Hadoop Platform, i.e., HDFS(Hadoop Distributed File System) and Vice versa. This paper also provides recommendations on how to increase the performance and reduce the latency of the existing Sqoop processing by using Spark Framework.

Keywords - *Hadoop; HDFS; Sqoop; MapReduce; Spark*

I. INTRODUCTION

Data Analytics has become crucial for every organization to get the proper insights into how a product or service is functioning. With the increase in the usage of the internet and IOT devices, the data has been increased drastically. The increase in the data from Terabytes to Petabytes and different forms of data like Structured, Unstructured, and Semi-Structured is often refers termed as Big Data.

The term Big Data refers to extensive, massive, complex, heavy, Structured, and Unstructured datasets. Big Data was defined with three V's by Analyst Doug Laney. They are as follows:

- Volume
- Velocity
- Variety

Volume is defined as extensive large datasets. Velocity refers to the high frequency of data, and Variety refers to different forms of data like both Structured and Unstructured.

In recent days along with the three v's, big data experts proposed additional v, i.e., veracity, which refers to the quality of the data [1].

Big Data Analytics has become a key aspect for every company to generate profits out of data and for a better understanding of customer behavior. This large volume of data cannot be handled and processed by the traditional RDBMS (Relational Database Management Systems). This becomes a need for the opt of the new framework, i.e., Hadoop Framework. Hadoop can process a large volume of data and store inefficient ways.

Hadoop is defined as a framework that supports the distributed processing of large volumes of data across multiple systems by using cheap commodity hardware and high reliability and availability by maintaining the replication Factor [2].

Most of the organizations started migrating to the Big Data Platform, i.e., Hadoop Platform, which is HDFS (Hadoop Distributed File System), to store and process the large volumes of the data. For data migration and data movement between the relational data systems and the Hadoop Sqoop is used.

Sqoop uses the MapReduce program to transfer the data between the systems. This article explains how the sqoop can be used with the Spark framework to increase the performance and reduce the time for data import/export.

II. SQOOP

Sqoop is a tool used for transferring large volumes of data between the RDBMS systems and the Hadoop Platform. It works with databases like MYSQL, Oracle, MSSQL Server, PostgreSQL, etc. It is used as ETL to fetch the tables from the RDBMS and write it into HDFS. It also exports the data from the Hadoop Distributed File System to any Outside external structured database systems [3].

III. SQOOP ARCHITECTURE AND FUNCTIONALITY

Sqoop uses to import and export commands to both import and export the data. It uses the import command to import the data from the Structured database systems, i.e., RDBMS to Hadoop and uses the export command to export the data from the HDFS to RDBMS.

It uses the functionality of Map Reduce to store the data into the Hadoop Distributed File System.

Hadoop uses the MapReduce Program, i.e., Framework, to store and process the data. In General, Most of the Jobs in the Hadoop will go through both the map and reduce phases. Map Phase takes the input of the data and splits the input data to achieve parallelism across the Cluster. The input of each phase is a key-value pair. In the Map Phase, each input split of the data is passed to the map function to get the output value and then followed by shuffling and sorting of the output from the map phase. Reduce Phase will combine the values and return the output. MapReduce will be completed in four steps like Splitting, Mapping, Sorting, and Reduce [4].

The architecture of the Sqoop is explained in Fig.1.

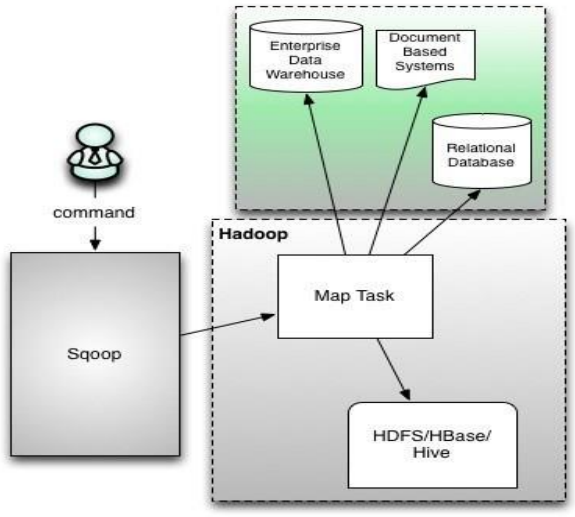

Fig.1 Sqoop Architecture

Sqoop makes use of the above-mentioned MapReduce program functionality to import and export the data. Sqoop uses Mapper to take the input of the data from external sources like RDBMS and uses a reducer to copy the data to the destination, i.e., HDFS(Hadoop Distributed Filesystem) as Hive/HBase Tables [5].

From the Fig.1. it explains how Sqoop takes command to initiate Map Task to retrieve the data from the source and then copies the data to the destination.

IV. SQOOP CONNECTOR

Sqoop is used for transferring the tables from RDBMS to the HDFS. It uses the JBDC connection to connect to any RDBMS databases with the hostname by specifying the username and password of the database [6].

Sqoop is operated with two commands. They are

- Sqoop Import
- Sqoop Export.

V. SQOOP IMPORT

Sqoop Import command is used to import the data from traditional RDBMS /Datawarehouse systems to the Hadoop Distributed File System. It reads the table rows from the RDBMS tables and imports the rows into HDFS as multiple files. The process of importing is done in parallel, so output at HDFS will contain multiple files in the HDFS. These files typically have multiple formats like Text, Avro, Sequential, etc., with fields terminated by commas, tabs, etc. [7].

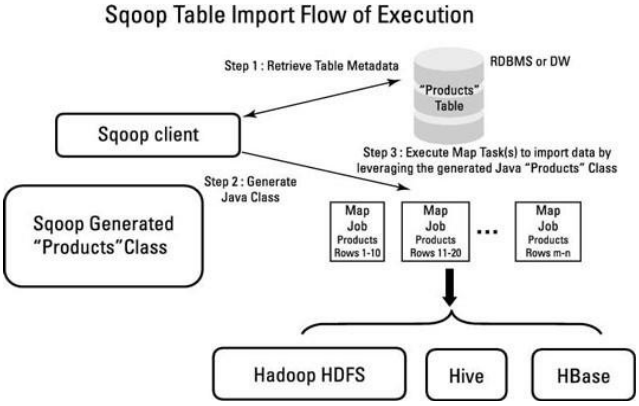

Fig.2 Sqoop Import Functionality

From the Fig.2. it is clear how the Sqoop import command works. It takes Sqoop import as a command and retrieves the metadata of the table from Traditional Database Systems, launch the Java Class and executes the Map Tasks, which is the phase of MapReduce to get the input data and then either use Reduce or without Reduce to copy the data across to the Hadoop Cluster as File, Hive and HBase Tables.

```
sqoop import –jdbc:mysql://dbname \
--table tablename –split-by primary key \
--fields-terminated-by- "\t" \
--username user -P \
--target-dir directoryname
```

A. Sample command to do the Sqoop import

-P refers to a Password that needs to be specified on a command line or as a separate file [8].

It also uses num of mappers as options in sqoop import to increase the parallelism.

B. Steps Involved

- Read the data from the RDBMS systems like MYSQL in the above import code snippet and do the Map-Reduce Program, which is in Java in the background
- Tries to Understand the data, and the code generation is performed to fetch the data
- It creates the Java File and will compile it. After the compilation, a Jar File is generated
- Jar File will help the sqoop import to apply the structure to the data it takes from the RDBMS
- If we specify to delete the target directory, if it exists, it will delete it and then import the data. While importing data, it connects to the Resource Manager in the Hadoop Framework to get resources allocated and then launches Application Master. To achieve the Parallelism and perform equal distribution, it takes a number of mappers and then uses the boundary query on a primary key to see the min and max of the rows to equal distribute among the mappers we specify in the sqoop Import or else default it takes 2

VI. SQOOP EXPORT

Sqoop Export is used to Export the Data from the HDFS to any RDBMS Systems like MYSQL, Oracle, etc. [9].

A. Sample Code Snippet for sqoop Export+

```
Sqoop export –connect jdbc:mysql://dbname –
username user \

-P –export-dir dir\filename \

--table tablename \

--input-fields-terminated-by “,” \

--direct
```

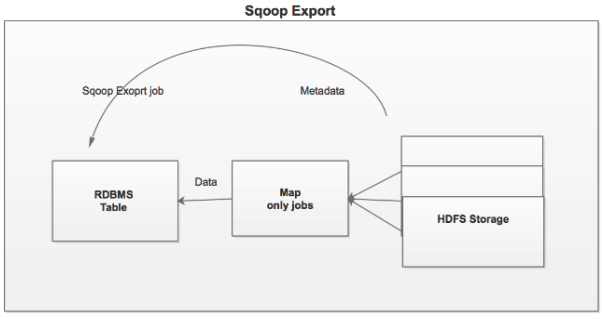

Fig.3 Sqoop Export Functionality

Sqoop Exports performs the same steps used in the sqoop Import, but only the difference is the source becomes HDFS, and the destination becomes RDBMS [10] . Another major difference is it uses boundaries from block size instead of the primary key used in sqoop import. As the Hive Tables/ HDFS does not have the concept of primary key and split will be taken care of by the sqoop internally [11].

VII. PROPOSED SOLUTION

In the functionality of Sqoop, we clearly saw that Sqoop uses the Map Reduce Program to both import and export the data. Map Reduce is a program that runs at disk level based on the Hadoop Framework. So, the bulk data transfer, like Terabytes of data transfer over the network, will be slow and take a long time to complete.

This Paper explains how the Performance of the sqoop can be increased. The proposed solution was to use the Spark Framework for data transfer by replacing the MapReduce for better Performance.

Spark uses the in-memory computation rather than a disk which is 100 times faster than the MapReduce programming. This will increase the performance of the data transfer between the system and reduces the time to complete the Job

VIII. IMPLEMENTATION

Sqoop by default uses the MapReduce with the connector API to connect to the RDBMS for the data transfer

In my new proposed solution, we will use the **Spark Framework to create, Submit and execute the Job**.

A. Sqoop Spark Job Creation Steps

- Spark Context needs to be created from the configs
- Launch the Sqoop Spark Job and call the SqoopSparkJob.init
- Use the Create Job API to launch the sqoop Job
- Call the SqoopSparkJob.Execute to execute the Job

B. Sqoop Spark Submission Sample Code on a command Line

```
bin/spark-submit --class
org.apache.sqoop.spark.SqoopJDBCHDFSJob
Driver --master yarn  pathofthejar --confDir
sqoopconfdirectory -- jdbcString
jdbc://typeofdatabase/databasename –u
username –p password  --target-dir
pathofdirectory --numE 4  --numL 4
```

The main concept of the new proposed system is Sqoop Jobs have been handled and managed by the **Spark Context.**

The New Proposed system can claim 100 times faster than the existing system because the framework we are implementing is Spark. Spark Framework is an engine used for processing large volumes of data. Spark is faster because the computation is done at the memory level of the worker nodes and does not perform the Input-output operations at the disk level, but Hadoop MapReduce Framework does make use of disk [13].

IX. RESULTS

I performed experiments on performance benchmarks on importing MYSQL tables with 250k records without partitioning and 1 million Records with Partitioning. In both scenarios, Sqoop with Spark Framework processes the data faster than the Sqoop with Map-Reduce Framework. Results of both MapReduce and Spark Frameworks in processing 250k records and 1 million records are clearly shown with histograms in Fig.4 and Fig.5.

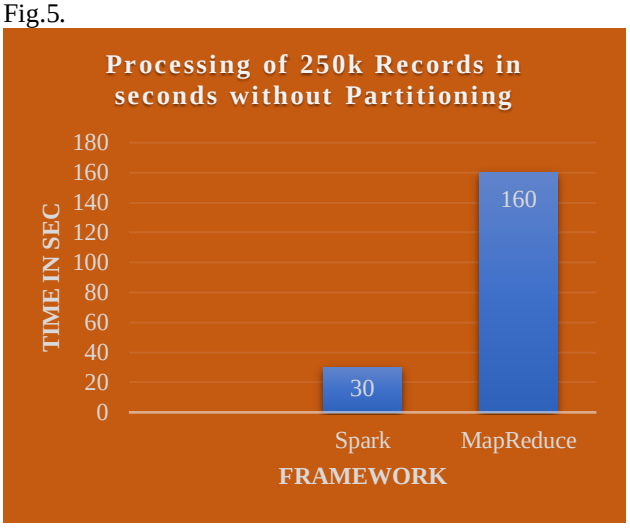

Fig. 4 Results of Processing 250K Records Table

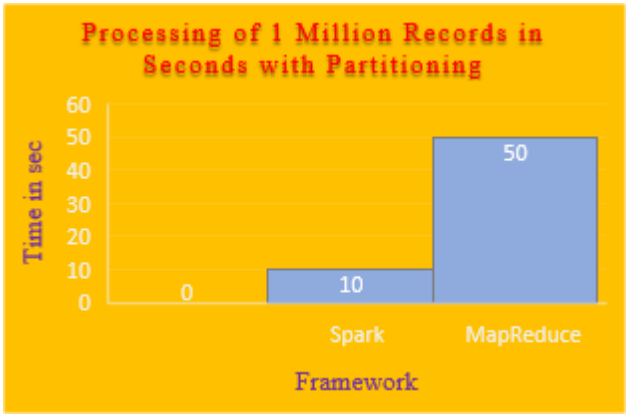

Fig. 5 Results of Processing 1 Million Records Table

X. CONCLUSION

Sqoop is a great tool that is used for bulk data transfer between the RDBMS and Hadoop Platform. It also does the data transfer between the Hadoop Platform to the RDBMS systems. Sqoop tool uses the MapReduce program for the data transfer.

This paper comes up with solutions and recommendations for using the sqoop on spark framework to increase the performance and reduce the time for the data transfer between the systems and also explains how the traditional sqoop works and its functionality.

REFERENCES

[1] What is BigData, [Online]. Available: https://datasciencedegree.wisconsin.edu/data-science/what-is-big-data/.

[2] Apache Hadoop Overview, [Online]. Available: https://hadoop.apache.org/.

[3] What is Sqoop, [Online]. Available: https://www.ucartz.com/clients/index.php?rp=/knowledgebase/833/Hadoop-What-is-Sqoop-and-Flume.html.

[4] MapReduceOverview, [Online]. Available: https://docs.marklogic.com/guide/mapreduce/hadoop#:~:text=Status%20and%20Logs-,MapReduce%20Overview,step%20map%20and%20reduce%20process.&text=The%20top%20level%20unit%20of,reduce%20phase%20can%20be%20omitted..

[5] ApacheSqoop-a-means-to-work-with-traditional-database, [Online]. Available: https://blogs.perficient.com/2016/08/11/apache-sqoop-a-means-to-work-with-traditional-database/#:~:text=Sqoop%20uses%20export%20and%20import,as%20well%20as%20fault%20tolerance..

[6] SqoopUserGuide v1.4.2, [Online]. Available: https://sqoop.apache.org/docs/1.4.2/SqoopUserGuide.html.

[7] SqoopUserGuide v1.4.1, [Online]. Available: https://sqoop.apache.org/docs/1.4.1-incubating/SqoopUserGuide.html#:~:text=With%20Sqoop%2C%20you%20can%20import,process%20is%20performed%20in%20parallel..

[8] Sqoop Import Command, [Online]. Available: https://docs.cloudera.com/runtime/7.2.1/migrating-data-into-hive/topics/hive_create_a_sqoop_import_command.html.

[9] Sqoop Export Command, [Online]. Available: https://community.cloudera.com/t5/Support-Questions/Using-direct-option-in-Sqoop-import-export/m-p/66930.

[10] P. S. V. Naresh Kumar, Modern Big Data Processing With Hadoop, Packt Publishing, (2018).

[11] J. Reddy, Introduction to Sqoop Architecture, [Online]. Available: https://www.freecodecamp.org/news/an-in-depth-introduction-to-sqoop-architecture-ad4ae0532583/.

[12] Spark-Sqoop-Job, [Online]. Available: https://www.wikitechy.com/tutorials/sqoop/spark-sqoop-job.

[13] M. Williams, Apache Spark vs. Map Reduce, 30 August 2017. [Online]. Available: https://dzone.com/articles/apache-spark-introduction-and-its-comparison-to-ma#:~:text=The%20biggest%20claim%20from%20Spark,O%20operations%20with%20the%20disks.