iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** VI **Month of publication:** June 2024

DOI: https://doi.org/10.22214/ijraset.2024.63171

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue VI June 2024- Available at www.ijraset.com

Solving Hadoop Drawbacks by Using Spark Framework

Akshata Parab[1], Sakshi Singh[2], Rutuja Patil[3]
Department of MCC, Mumbai University

Abstract: ***In Today's world, peta byte to exa byte data is producing every minute by social media, digital marketing, social sites. This is all generating a big data. Big data is a collection of huge and complex data sets and data volume that include the huge amount of data, data management services, social media analytics and real-time data. Big data analytics is the process of determining large volume of data. Because of generating large volume of data every second, shortage of the processing of that data techniques. Hadoop is the latest data processing framework which processes large volume of data through its HDFS and MapReduce.***

Keywords: BigData, Hadoop, Spark, MapReduce, Hive, HBase, Sql

I. INTRODUCTION

Big data as name suggests, big amount of data. Big data means the data in terms of large volume and complex data. Because of this complex data, its processing is getting difficult now-a-days. And that's why the traditional big data processing software and technologies can't handle it. Big data contains large datasets of structured and unstructured data.

II. BIG DATA CHARACTERISTICS

1) Volume – It refers to the amount of data is being collected.
2) Velocity – It refers to the speed of data coming/generating.
3) Variety – It refers to the type/kind of data.
4) Value - It refers to the usefulness of the collected data.
5) Veracity – It refers to the quality of data is being generated.

III. RESEARCH GAP

We have done literature review on some research paper, and we found below drawbacks:

1) Hadoop reads and writes data every time from disk which slows down the speed of processing.
2) Performance reduced due to slow processing.
3) Due to high latency, Hadoop does not process data interactively.
4) Hadoop can't process real-time data.
5) Hadoop is less user-friendly.
6) Limited language support.

IV. APACHE SPARK FRAMEWORK

Apache Spark is an open-source processing system. It has distributed processing system used for big data tasks. It uses optimized query execution and in-memory caching, and for fast analytic queries against data of large size. It provides development APIs in Java, Scala, Python and R, and supports code reuse across multiple services—batch processing, interactive queries, real-time analytics, machine learning, and graph processing.

Hadoop MapReduce is a programming model used for processing big data sets with a parallel, distributed algorithm. Developers can write highly parallelized operators, without having to worry about work distribution, and fault tolerance. The big thing is the sequential multi step process used to run a job is a big challenge. In each step, MapReduce reads data from disk, performs operation on it and writes data back to HDFS. Because of each step that requires a disk read, write, MapReduce slows down the process and hence slow the latency of disk input/output.

Spark was created to analyse the limitations of MapReduce, by improvizing processing in-memory, reducing the number of steps to execute a job, and by reusing data in multiple parallel operations. With Spark, where data is read into memory, operations performed, and the results written back resulting in a much faster execution.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue VI June 2024- Available at www.ijraset.com

Spark also reuses data by using an in-memory cache to highly peed up machine learning algorithms that frequently call a function on the same dataset. Data re-use is happened through the creation of Data Frames, an abstraction over Resilient Distributed Dataset (RDD), which is a collection of objects that is cached in memory and reused in multiple Spark operations. This drastically lowers the latency which makes Spark multiple times faster than MapReduce, especially when doing machine learning, and interactive analytics.

V. BENEFITS OF APACHE SPARK

There are many benefits of Apache Spark which makes it one of the most usable frameworks in any Hadoop ecosystem projects. These include:

A. Fast

Spark can run queries fast for any size of data due to in-memory caching and fast query execution.

B. Developer Friendly

Apache Spark highly supports languages like Java, Scala, R, and Python, giving you a number of languages for developing your applications. These APIs helps your developers, because they provide the simple, high-level operators which lowers the amount of code required.

C. Multiple Workloads

Apache Spark has the ability to run multiple workloads, including interactive queries, real-time analytics, machine learning, and graph processing. One application can combine multiple tasks precisely.

The Spark framework includes:

1) Spark Core is the main component of Platform.
2) Spark SQL for interactive queries
3) Spark Streaming for real-time analytics
4) Spark MLlib for machine learning
5) Spark GraphX for graph processing

D. Spark Core

Spark Core is the main functionality of the platform. It is responsible for memory management, fault recovery, scheduling, distributing & monitoring jobs, and interacting with storage systems. Spark Core is defined through an application programming interface (APIs) developed for Java, Scala, Python and R. These APIs helps your developers, because they provide the simple, high-level operators which lowers the amount of code required.

E. MLib

Machine Learning

Spark contains MLlib, a library of algorithms which is used to do machine learning on data at scale. Machine Learning models can be implemented by data scientists with R or Python like languages on any Hadoop data source, saved using MLlib functionality, and imported into a Java or Scala-based pipeline. Spark was designed for fast, interactive computation that runs in memory, helps machine learning to run quickly. The algorithms include the ability to do classification, regression, clustering, filtering, and pattern mining.

F. Spark Streaming

Real Time

Spark Streaming is an effective framework that increases Spark Core's fast scheduling capability to do analytics in streaming. It ingests data in small batches and enables analytics on that data with the same application code written for batch analytics. This improves developer productivity, because they can use the common code for both batch processing as well as for real-time streaming applications.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue VI June 2024- Available at www.ijraset.com

G. Spark SQL

Interactive Queries

Spark SQL is a distributed query engine that provides low-latency, interactive queries up to 100x faster than MapReduce. It includes a code-generation, cost-based optimizer, and columnar storage for fast queries, while increasing to thousands of nodes. Business analysts can use standard SQL or the Hive Query Language for querying data. Developers can use APIs, available in Scala, Java, Python, and R.

It supports various data sources out-of-the-box including JDBC, ODBC, JSON, HDFS, Hive, ORC, and Parquet. Other storages are Amazon Redshift, Amazon S3, Couchbase, Cassandra, MongoDB, Salesforce.com, Elasticsearch, and many others can be found from the Spark Packages ecosystem.

H. GraphX

Graph Processing

Spark GraphX is a graph processing framework. It supports distributed nature of work. GraphX provides ETL, iterative graph computation and exploratory analysis to enable users to interactively develop and transform a graph data structure at scale. It comes with a selection of distributed Graph algorithms and highly flexible API.

I. Architecture in Spark

The architecture of Spark contains three main components which are listed below:

1) API

This element used by many application developers for creating Spark-based applications with a classic API interface. Spark used by API for Python, Java, and Scala programming languages.

2) Data Storage

Spark uses the Hadoop Distributed File System for various purposes of data storage. It works with any data source which is compatible with Hadoop having HDFS, Cassandra, HBase, etc.

3) Resource Management

The Spark can be referred as the stand-alone server. Also, it can be updated on any shared computing framework such as YARN or Mesos.

J. RDD in Spark

RDD stands for Resilient Distributed Dataset. It is a main component of the Spark framework. Assume RDD like any table inside the database. It could take any data type. Spark can store data in Resilient Distributed Dataset on distinct partitions. These datasets has been used to rearrange the upgrading and reusing the processing of data. These datasets are fault-tolerant due to the RDD which gives how to recreate and reuse the functions and the computations for the large datasets.

Resilient Distributed Datasets are immutable. We can change RDD using a transformation. This transformation will give the new RDD which same as the old RDD only. RDD provides its support for two main operations:

1) Action

This operation assesses and returns a newer value. Each query of data processing is updated, and the final value should returned if a function of action is called over an RDD object. A few of the operations of Action are reduce, collect, take, first, foreach and countByKey.

2) Transformation

Transformation does not return any single value. It returns a newer RDD. Nothing will be computed if we call any function of transformation. It only takes the RDD and then returns a new RDD.

Some of the operations of Transformation are reduceByKey, flatMap, filter, map, coalesce, pipe, aggregateByKey and groupByKey.

VI. PERFORMANCE ANALYSIS

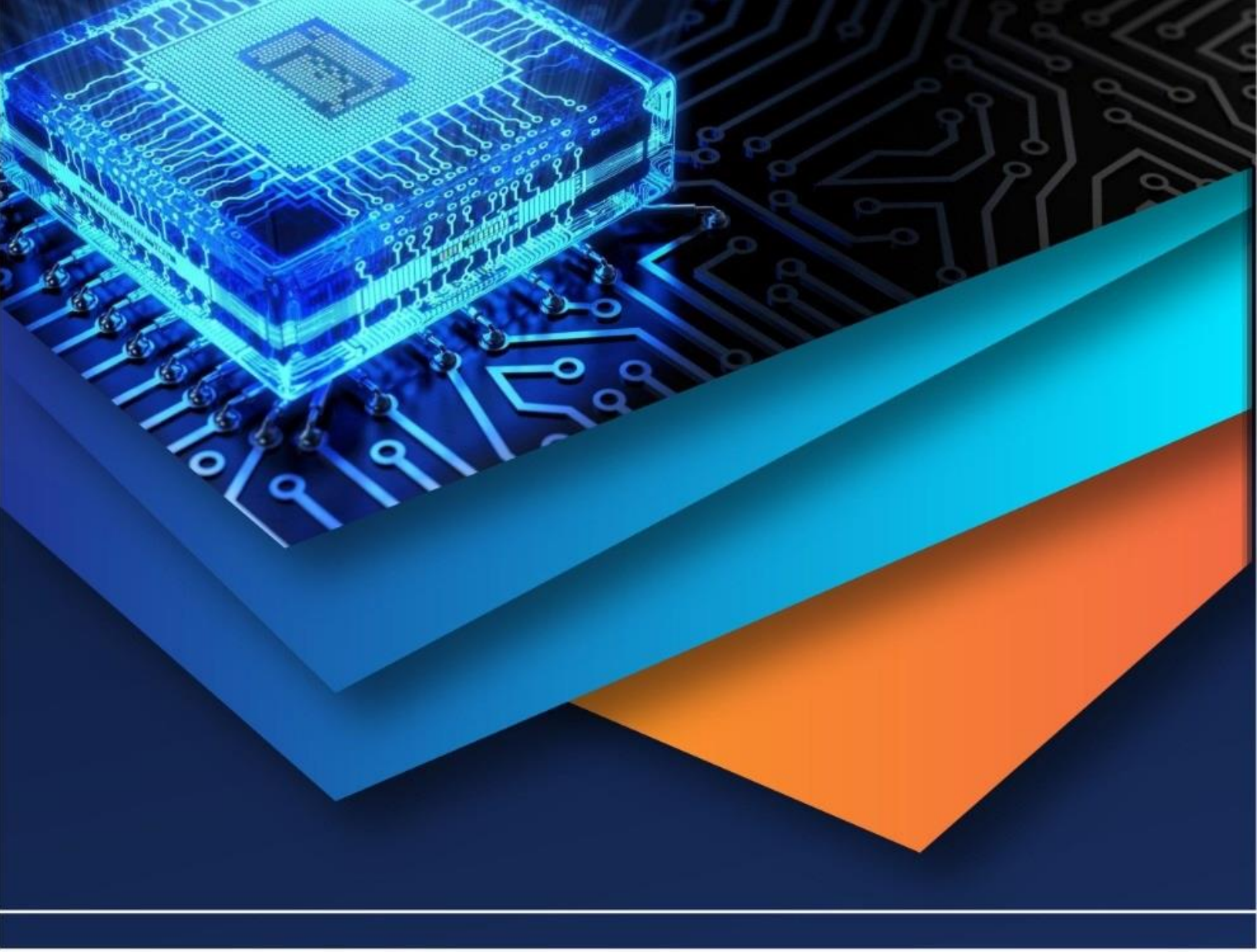
1) Spark gives a powerful architecture capable of handling very large amounts of data. There are many Spark optimization techniques which processes and data handling, including performing tasks in memory. It also used for storing frequently accessed data in a cache, which reduces latency during retrieval. Spark is also designed for data processing and scalability can be distributed across multiple computers, increasing the available computing power.
2) Spark's fast processing speeds are achieved by RDDs. While many frameworks depend on external storage systems such as a Hadoop Distributed File System (HDFS) for reusing and sharing data between computations, RDDs support in-memory computation.

VII. CONCLUSION

1) In Hadoop, we read from and writes on disk every time which increases the load on system and hence slows down the process.
2) Due to slow process, latency increases which gives loss of data sometimes.
3) In Spark, number of cycles of reading and writing data is low which increases speed. And hence it has low latency.
4) In Hadoop, we can use JAVA or PYTHON but in Spark, we can use other languages also like R, Scala, Spark SQL.
5) Hadoop only processes data in batches and can't process real time data.

REFERENCES

[1] https://journalofbigdata.springeropen.com/articles/10.1186/s40537-020-00388-5
[2] https://www.researchgate.net/publication/281403776_Big_Data_And_Hadoop_A_Review_Paper
[3] https://www.ijcst.com/vol74/1/11-iqbaldeep-kaur.pdf
[4] https://www.researchgate.net/publication/339176824_Apache_Spark_A_Big_Data_Processing_Engine
[5] https://www.academia.edu/34540716/Hadoop_The_Definitive_Guide
[6] https://www.geeksforgeeks.org/difference-between-hadoop-and-spark/



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY