A Survey on the Performance Comparison of Map Reduce Technologies and the Architectural Improvement of Spark

G S Raghavendra[1], Bezwada Manasa[2], M. Vasavi[3]
raghavendragunturi@gmail.com
manasabezawada04@gmail.com
Vasavi.lahari@gmail.com
Assistant Professors,Computer Science and Engineering,
RVR & JC College of Engineering,
Guntur, Andhra Pradesh, India

Abstract

Hadoop and Apache Spark are Apache Software Foundation open source projects, and both of them are premier large data analytic tools. Hadoop has led the big data industry for five years. The processing velocity of the Spark can be significantly different, up to 100 times quicker. However, the amount of data handled varies: Hadoop Map Reduce can process data sets that are far bigger than Spark. This article compares the performance of both spark and map and discusses the advantages and disadvantages of both above-noted technologies.

Keywords-: *Hadoop, spark, Map reduce,*

1. Introduction

The large-scale information scanning has become a remarkable platform for organizations to take advantage and exploit heaps of vital information. In the course of this vast information rise, Hadoop has progressed fiercely as an on-or cloud-based stage as the single-size solution for the huge scale problems of the corporate sector. [1] While Utilizing Hadoop has met a substantial part of the advertising, the best arrangement may be in some conditions while performing tasks on a traditional data collection. Hadoop is not an information base, but a general programming system was deliberately used to handle enormous quantities of structured and moderately information.

[1] For large-scale information evaluation, associations contemplating using Hadoop should examine if their present or future information demands require the type of capabilities that Hadoop offers. Organized information is described as information that resides in the fixed bounds of a record or document.

Due to the way structured information may be recorded, disclosed, questioned and explored, even in large quantities, in an essential and immediate method, a conventional set of data is usually implemented. [2] Unstructured data is referred to as information from a variety of sources, including communications, text archives, recordings, pictures, sound records, internet media postings.

A usual dataset cannot handle or examine unstructured information as both puzzling and voluminous. Hadoop's ability to add, Totals, and explore huge multi-source information stores without initially structuring allows associations to gain additional knowledge quickly. In this sense, Hadoop is perfect for storing, monitoring and evaluating large quantities of unstructured information for companies [3]

2. Map Reduce

MapReduce is a programming paradigm that provides enormous scalability over a Hadoop cluster's hundreds or thousands of computers. MapReduce, as the processing component, lies at the heart of Apache Hadoop. The phrase "MapReduce" refers to two independent activities performed by Hadoop applications.[2] The first type of task is the map job, which takes a collection of data and turns it into another set of data, where individual components are split down into tuples (key/value pairs).The reduction task takes as input the result of a map and merges those data tuples into a smaller collection of tuples. The reduction task is always run after the map job, as the term Map Reduce indicates.[2]

2.1 Mapper

The task of the mapper is to process the supplied data. In most cases, input data comes in the form of a file or directory, which is then stored in the Hadoop file system (HDFS). Line by line, the

Manuscript received May 5, 2022
Manuscript revised May 20, 2022
https://doi.org/10.22937/IJCSNS.2022.22.5.18

mapper function is fed the input file. The mapper parses the input and generates numerous tiny data pieces. [2]

2.2. Reducer

This is a hybrid of the Shuffle and Reduce stages. The Reducer's role is to process the mapper's data. It generates a new set of outputs after processing, which is saved in HDFS.[3]

3. Spark

Apache Spark is a free and open-source distributed computing system with high-level APIs in Java, Scala, Python, and R. It has access to data stored in HDFS, Cassandra, HBase, Hive, other Hadoop data source. it may be operated under Standalone, YARN, or Mesos cluster managers. [3]

3.1 Hadoop vs Spark

Hadoop is built on batch processing of large amounts of data. This means that the data is kept throughout time and then processed with Hadoop. Processing in Spark, on the other hand, can be done in real time. This real-time processing capability in Spark enables us to tackle the Real Time Monitoring use cases discussed in the preceding section. In addition, Spark can do batch processing 100 times quicker than Hadoop Map Reduce. As a result, Apache Spark is the industry's go-to technology for large data processing.[4]

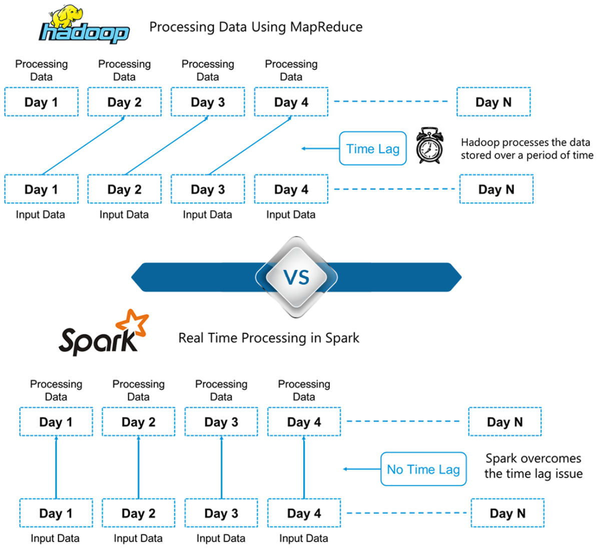

Fig 1: Hadoop vs Spark

4. **Performance Comparision Map Reduce and Apache spark.**

The speed of Apache Spark is well recognized. It outperforms Hadoop Map Reduce in memory by 100 times and on disc by 10 times. The reason for this is because Spark processes data in RAM, but Hadoop Framework must store data to disc after each Map or Reduce operation. The computational power of Spark provides near-real-time analytics, making it a perfect tool for IoT sensors, payment processing systems, advertising campaigns, security assessment, pattern recognition, social networking sites, and log surveillance.Spark offers built-in APIs for Scala, Java, and Python, as well as Spark SQL for Database users.[5] Spark also offers basic building pieces that make it simple for users to construct user-defined functions. When performing commands, you may leverage Apache Spark in interactive environment to obtain instant response.[5] Hadoop Map Reduce, on the other hand, is written in The java programming and is tough to construct. Unlike Spark, Map Reduce doesn't really support interactive use. Considering the above - mentioned characteristics, it is possible to infer that Apache Spark is more user-friendly than Map reduce.Spark, like Map Reduce, uses speculative execution and restarts for each job.[6] However, the fact that Map Reduce relies on hard disks provides it a minor edge over Apache Spark, that relies on RAM. If an unexpected incident occurs and a Map Reduce activity breaks in the middle of operation, the function may resume where it was left off. It is not really feasible with Spark since it must responsibilities with respect from the beginning. In terms of security, Map Reduce surpasses Spark Spark. For instance, Apache Spark's security is set to "OFF" by nature, making users exposed to cyber-attacks. Spark implements RPC channel verification using a secret key. It also includes event recording and the ability to protect Web User Interfaces using Javax Servlet Filters. Furthermore, because Apache Spark can operate on Yarn and leverage HDFS capabilities, it can use Hadoop File Permissions, Kerberos Authentication, and node security. Map Reduce can make advantage of all Hadoop security capabilities and interact with other Hadoop Security Projects. As a result, Map Reduce provides more security than Hadoop.

5. Limitations of Spark

Spark does not have its own file system. This does not include a filing system. It is usually dependent on other file management systems. As a result, it must integrate with one , if not Hadoop one more cloud-based data platform. This is one of Spark's core problems. Spark will not allow universal processing. The live data that enters is automatically split into bunches using Spark streaming. If such batches are of a predefined interval, each chunk of data is treated as a Spark Resilient distributed.[6]

6 .Architectural Changes in spark

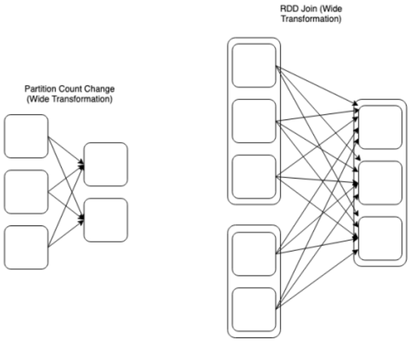

Fig 2 : Single RDD vs Multi RDD

Multiple RDDs are defined by the fact that the worker nodes will need to move data across a network to accomplish the required job. A merge is an instance of this, because we may need to acquire data from throughout the cluster to perform a comprehensive and proper join of different datasets.

Single RDDs are defined by a single input partition and have a single output partition. A filter is an illustration of this: we could have a data frame of data which we can refine down to a tiny datasets without requiring to understand any data kept on every other worker node.

7 . Data Ingestion Architecture

A. Lambda Architecture

Lambda Architecture (LA) [5is a standard system for overseeing enormous information which empowers blending of real-time information with cluster information. The fundamental design of lambda offers three layers: speed layer for constant information, batch layer for enormous volume of static authentic information pool and serving layer that incorporates continuous and batch jobs. Lambda Architecture coordinates low dormancy constant system with high throughput Hadoop batch structure. [5] information from Kafka message line gets ingested to both and stream processor structures. While stream processors can break down information, batch module stores the ingested information pool into HDFS over the time before preparing. [5] Apache Storm and Hadoop Map Reduce structures are utilized at stream and group modules separately. [5] A NoSQL information store

(Cassandra) joins the batch and ongoing perspectives at the serving layer.

B. Stream Processing Engine it instates itself with prepared model produced from group put away into HDFS. Stream motor uses the batch information as beginning balance to begin with and expands over it, steadily at a predetermined window interim. The fundamental test spilling layer faces is to process in-flight high speed of ingested information without first putting away into a document framework or a database. [5]

C. Data Miner has two sub segments: Distributed storage and preparing by Map Reduce. HDFS or a NoSQL information stores like Cassandra, MongoDB or Base can be a capacity alternative for batch oriented jobs [5]

D. Knowledge Miner and Knowledge Base Knowledge Minder (KM) is the cutting off layer to the end client.[12] It joins the perspectives from group and stream to give a deep rooted learning system. KM perseveres the outcomes into the Knowledge Base (KB). KM likewise performs information filtration, gives the system health and execution insights for information administration and monitoring purposes. [5]

8. Data Real Time Ingestion And Machine Learning

A. Streamed Machine Learning

Batch machine learning is applied for fixed arrangement of information. Normally, these systems are likewise iterative, and we play out various ignores preparing information to join to an ideal arrangement. In opposite, web based learning predicts on each progressing window of time span. [6] In a steady manner the model persistently refreshes as new data is gotten. Be that as it may, web based learning model can be utilized alongside batch setting. Like we can utilize stochastic slope plummet (SGD) enhancement to prepare arrangement and relapse model after each preparation model. [6]

B. Streaming Regression

Training: Takes the labelled data points. Model gets trained on every batch of the input stream. It can be called repeated time to train on different stream. [6]

Predict: It also take labelled data points and tells the model to make prediction on the input stream. [6]

C. Streaming K-Means Clustering

In streaming K means clustering, model is refreshed with each passing window utilized on a blend between group focuses figured from the past batchs and the present cluster. Calculation begins with allotting information focuses to their closest batch. [6] For each new emphasis, when new information comes, register new group focuses, at that point update each batch utilizing following recipe [6]

9. Performance Compariosn Of Data Ingestion Tools

SNO	DATA INGESTION TOOL	DESCRIPTION
1	Apache Kafka	Message Broker System.Peformance lags with size of data
2	Apache Nifi	Provides directed graph of data routings.it is system mediation logic
3	Wavefront	Used for data ingesting ,visualizing and alerting metric data
4	Amazon Kinesis	Cloud Based data ingestion system.
5	Apache Samza	Message API, It maintains snapshotting and restoration of stream processor state.
6	Apache Flume	Low end data ingestion system only works well with small data with high latency
7	Apache Sqoop	Static data ingestion system, work only with data bases

10. Limitations Of Existing Ingestion Frame Works

Data Acquiring It is absurd to expect to deal with voluminous stream of streaming information. The framework must be fit for adjusting with the speed of approaching information and furthermore with assortment of information.[7] The Processing of organized information goes about as an ideal contribution for direct frameworks, while the unstructured information requires parcel of information pre-preparing like separating, extraction and association into organized configuration. The dormancy of the stream preparing framework shifts with organized and unstructured information. The right portrayal of information and information securing procedures rely upon the application based on the highest point of stream handling frameworks.[7]

B. Data Handling Second challenge is to appropriately deal with huge volume of information. The application requires examining the affectability of information, which need to store into persevering stockpiling.[10] A few applications just require putting away the combined prepared outcomes while different applications require putting away sifted and fundamentally composed handled information for later utilization and investigation.[11] The information taking care of and persevering stockpiling of information design changes with the application necessity. It should be appropriately evaluated by stream preparing frameworks.[9]

C. Data Modelling The preparing frameworks require in-stream handling capacities to have a low idleness. Thinking about the volume, assortment, speed and veracity of information, the stream preparing framework requires prescient models and effective calculations to extricate application connected to significant occasions from monstrous information streams. It likewise requires information models to perform extensive examination by joining every single accessible datum.[8]

11. Spark Improved Stream Framework

Real time processing of streaming data by using H-Stream frame work is done in two phases.

A. Phase 1

In this phase developing of a frame work known as H-Stream is done. The importance of this Frame work it can handle any type of data and can process data better than previous techniques

The tool may process data as following

1.Input data can be given from various sources like social media, YouTube etc

2.Then the data may first have compressed by using Map-Reduce version 2 (YARN).[13]

3. HIPI is a picture preparing library intended to be utilized with the Apache Hadoop Map Reduce system. It gives an answer for how to store an enormous assortment of pictures on the Hadoop Distributed File System (HDFS) and make them accessible for productive appropriated handling. [13]The essential info item to a HIPI program is a Hipi Image Bundle (HIB). [13]A HIB is an assortment of pictures spoke to as a solitary record on the HDFS. The HIPI conveyance incorporates a few valuable apparatuses for making HIBs, including a Map Reduce program that fabricates a HIB from a rundown of pictures downloaded from the Internet. The main handling phase of a HIPI program is a separating step that permits sifting the pictures in a HIB dependent on an assortment of client characterized conditions like spatial goals or criteria identified with the picture metadata. This usefulness is accomplished through the Culler class. Pictures that are winnowed are rarely completely decoded, sparing preparing time. The pictures that endure the separating stage are doled out to singular guide errands such that endeavours to augment information territory, a foundation of the Hadoop Map Reduce programming model.[13]

B. Phase 2

In this phase the data which will be the output of H-Stream may be analysed by using machine learning algorithm like KNN, SVM. After successful categorization of data. Then data is analysed by using of Hadoop tools like Hive, pig. Advanced data visualization techniques like isoclines, iso-surface, Oracle Visual Analyser, Microsoft Power BI for 2D, 3D visualization of processed data are used.[13].

Conclusion

This article compares the performance of both spark and map and discusses the advantages and disadvantages of both above-noted technologies and provides the required archietctural changes needed for the improvement of spark performance. The Stream Framework will be well suited for those applications which require and demand for real-time processing. Traditional methods of ingestion don’t support real time low latency processing. In future This frame work can be extended for all multimedia applications which require real time analysis and frame work

Acknowledgments

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors. The authors would like to thank the editor and anonymous reviewers for their comments that help improve the quality of this work

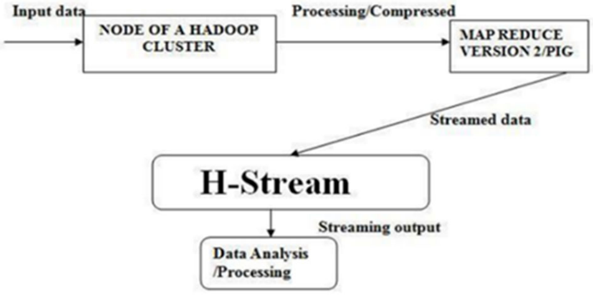

Fig 3: Architecture of Spark Framework

Expected Results of the Proposed Method

1.Real time querying helps users to take accurate instant decision support.
2.Need for additional hardware and tools for processing of large data can be decrease
3.Highly robust frame work which makes querying/processing easy.
4.They may be no need of separate tools for capturing streaming data processing and visualization since everything is encapsulated as a single frame-work.
5.Works on both structured and unstructured data.

References

[1] “Apache Map Reduce” IBM technologies 2020.

[2] “Apache Spark Tutorial for Beginners” Data Flair 2020.

[3] “Real Time Cluster Computing Framework” Sandeep Dayananda, 2020

[4] “Hadoop MapReduce vs Spark: A Comprehensive Analysis “Nicholas Samuel on Data Integration, ETL

[5] “Apache Spark Pros and Cons” Knowledge Hut. 2020

[6] “Limitations of Apache Spark” techvidvan 2020

[7] Adesh Chimariya B. Professor Mika Mäntylä, “Streaming Data AnalyticsBackground, Technologies, and Outlook,” Master’s Thesis, University of Oulu

[8] Ovidiu-Cristian Marcu , Alexandru Costan , Gabriel Antoniu , Mar´ıa S. Perez-Hern ´ andez ´ Bogdan Nicolae† , Radu Tudoran, Stefano Bortoli 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS) ,pp.1480–1485.

[9] UnGyu Han and Jinho Ahn, “Dynamic Load Balancing Method for Apache Flume Log Processing,” in Advanced Science and Technology Letters, Vol.79 (IST 2014), pp.83-86

[10] Yang Ruan, Zhenhua Guo, Yuduo Zhou, Judy Qiu, Geoffrey Fox, “HyMR: a Hybrid MapReduce Workflow System,” ACM 978-1-4503-1339-1/12/06.

[11] Gautam Pal, Gangmin Li, Katie Atkinson “Multi-Agent Big-Data Lambda Architecture Model for E-Commerce Analytics” „mdpi ,pp.1-15.

[12] Gautam Pal, Gangmin Li, Katie Atkinson “Big Data Real Time Ingestion and Machine Learning”, IEEE Second International Conference on Data Stream Mining & Processing,

[13] Gunturi S Raghavendra,Prof Shanthi Mahesh, Prof MVP Chandrasekhara Raohttps://www.ijrte.org/portfolio-item/e6045018520/