ISSN: 2278-4632
Vol-11 Issue-02 2021

PERFORMANCE EVALUATION ON SPARK AND MAP REDUCE USING STREAMING DATA

Ipsita Samal, Vidyadhar
[1](Electronics & Communication Engineering, Gandhi Engineering College,Bhubaneswar)
[2](Electronics & Communication Engineering, Gandhi Engineering College,Bhubaneswar)

ABSTRACT

Dynamic information stream handling utilizing constant programming model for processing large data sets with a parallel, distributed algorithm on a cluster is at present a high worry as the measure of information being created is expanding step by step with the development of the Internet of Things, Big Data and Cloud. Big data are portrayed by immense volume that can land with a high speed and in various organizations from numerous sources. Accordingly, continuous programming model strategies ought to be fit for preparing the information to separate an incentive out of it by tending to the issues identified with these qualities that are related to information streams. In this work, we asses and break down the ability to exist Map-Reduce andSparkprocedurestodealwithdynamicinformation streamsandweintroducewhetherthecurrentsystemsareimportantin the currentcircumstance.

Keywords: *Big data, Spark, Map-reduce, Data streams.*

1. INTRODUCTION

Big data is a blanket term for the non-traditional strategies and technologies needed to gather, organize, process, and gather insights from large datasets. While the problem of working with data that exceeds the computing power or storage of a single computer is not new, the pervasiveness, scale, and value of this type of computing have greatly expandedinrecentyears.Theessentialnecessitiesofworking with huge information are the same as the prerequisites for working with datasets of any size. Nonetheless, the monstrousscale,thespeedofingestingandhandling,andthe qualities of the information that must be managed at each phase of the procedure introduce huge new difficulties when outlining arrangements. The objective of most huge information frameworks is to surface bits of knowledge and associations from huge volumes of heterogeneous information that would not be conceivable utilizing traditionaltechniques.

In 2001, Gartner's Doug Lane first presented what became known as the "three Vs of big data" to describe some of the characteristics that make big data different from other data processing:

- Volume: The sheer size of the data handled characterizes enormous information frameworks. These datasets can be requests of extent bigger than customary datasets, which requests more idea at each phase of the preparing and capacity lifecycle.
- Velocity: Another manner by which enormous information contrasts fundamentally from other information frameworks is the speed that data travels through the framework. Information isevery now and again streaming into the framework from various sources and is frequently anticipated that would be handled progressively to pick up experiences and refresh the present comprehension of the framework. This emphasis on close moment input has pushed numerous enormous information specialistsfromabunchsituatedapproachandmore like a constant gushing framework. Information is always being included, rubbed, handled, and investigated so as to stay aware of the flood of new data and to surface significant data early when it is generallypertinent.
- Variety: Data can be ingested from internal systems like application and server logs, from social media feeds and other external APIs, from physical device sensors, and from other providers. Big data seeks to handle potentially useful data regardless of where it's coming from by consolidating all information into a single system. The arrangements and sorts of mediacandifferfundamentallytoo.Richmedialike pictures,videodocuments,andsoundchroniclesare ingested close by content records, organized logs, and soon.

Because of the qualities of big data, individual computers are often inadequate for handling the data at most stages. To better address the high storage and computational needs of bigdata,computerclustersareabetterfit.Bigdataclustering software combines the resources of many smaller machines, seeking to provide several benefits:

- Resource Pooling: Combining the available storage space to hold data is a clear benefit, but CPU and memory pooling are also extremely important. Processing large datasets requires large amounts of all three of theseresources.

ISSN: 2278-4632
Vol-11 Issue-02 2021

- High Availability: Clusters can provide varying levels of fault tolerance and availability guarantees to prevent hardware or software failures from affecting access to data and processing. This becomes increasingly important as we continue to emphasize the importance of real-timeanalytics.
- Easy Scalability: Clusters make it easy to scale horizontally by adding additional machines to the group. This means the system can react to changes in resource requirements without expanding the physical resources on amachine.

Using clusters requires a solution for managing cluster membership, coordinating resource sharing, and scheduling actual work on individual nodes. Cluster membership and resourceallocationcanbehandledbysoftwarelikeHadoop's YARN (which stands for Yet Another Resource Negotiator) or ApacheMesos.

Sentiment analysis (SA), also known as opinion mining, has attracted an increasing interest. It is a hard challenge for language technologies, and achieving good results is much more difficult than some people think. The task of automatically classifying a text written in a natural language intoapositiveornegativefeeling,opinionorsubjectivity[1], is sometimes so complicated that even different human annotators disagree on the classification to be assigned to a given text. Personal interpretation by an individual is different from others, and this is also affected by cultural factorsandeachperson'sexperience.Andtheshorterthetext, andtheworsewritten,themoredifficultthetaskbecomes,as in the case of messages on social networks like Twitter or Facebook.

The rest of the paper is organized as follows: Section 2 presents related work. Section 3 discusses the available methodologies and their drawbacks. Section 4 analyses current capabilities of current Map Reduce to handle a huge amount of data. Section 5 focuses on the conclusion and related works.

2. RELATEDWORK

A. Extracting realtime social media data from data streams usingFLUME.

Flume is composed of the following components. Flume Event: It is the main unit of the data that is transportedinside theFlume(Typicallyasinglelogentry).Itcontainsapayload ofthebytearraythatistobetransportedfromthesourcepath to the destination path which could be accompanied by optionalheaders.

FlumeAgent:IsanindependentJavavirtualmachinedaemon process which receives the data (events) from clients and transports to the subsequent destination (sink or agent). Source:IsthecomponentofFlumeagentwhichreceivesdata from the data generators say, Twitter, Facebook, weblogs fromdifferentsitesandtransfersthisdatatooneormore channels in the form of Flume event. The external source sends data to Flume in a format that is recognized by the target Flume source. Example, an Avro Flume source can be used to receive Avro data from Avro clients or other Flume agents in the flow that send data from an Avro sink, or the Thrift Flume source will receive data from a Thrift sink, or a Flume Thrift RPC client or Thrift Clients are written in any language generated from the Flume thrift protocol. Channel: Once, the Flume source receives an Event, it stores this data into one or more channel and buffers them until they are consumedbysinks.Itactsasabridgebetweenthesourceand sinks.Thesechannelsareimplementedtohandleanynumber of sources andsinks.

Sink: It stores the data into the centralized stores like HDFS and HBase.

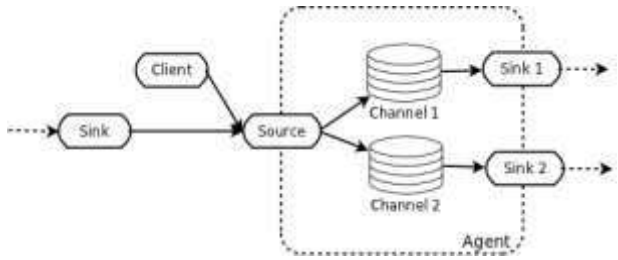

Fig 1: FLUME Architecture

B. Sentiment Analysis for social mediadata

Semantic approaches are characterized using dictionaries of words (lexicons) with the semantic orientation of polarity or opinion. Systems typically preprocess the text and divide it intowords,withproperremovalofstopwordsandalinguistic normalization with stemming or lemmatization, and then check the presence or absence of each term of the lexicon, usingthesumofthepolarityvaluesofthetermsforassigning the global polarity value of the text. Typically, systems also include;

ISSN: 2278-4632
Vol-11 Issue-02 2021

i) Anadvancedtreatmentofmodifierterms(suchasvery,too, little) that increase or decrease the polarity of the accompanyingterms

ii) Inversion terms or negations (such as no, never), which reverse the polarity of the terms to which theyaffect.

3. ANALYSIS BETWEEN MAP REDUCE AND SPARK

Authors of [2], consider major challenges of MapReduce on Data Storage, Analytics, Online Processing, Privacy, and Security. Customary information preparing and capacity approaches are confronting numerous difficulties in meeting theconsistentlyexpandingfiguringrequestsofBigData.This work concentrated on MapReduce, one of the key empowering approaches for taking care of Big Data requests by methods for very parallel handling on countless hubs. Schema-free challenges the data storage of the Map-Reduce which is addressed by No-SQL stores – MR with various indexing approaches. Analytics is one the challenges of MR which is caused by Statistical challenges of learning along withinteractiveanalyticsandscalingcomplexlinearalgebra. But, Data preprocessing could be the solution for the Statistical Learning whereas, Map interactive query processing addresses the challenges of Inter activeanalysis.

Authors of [3], performs a comparison study of MapReduce and Spark with respect to the processing of batch jobs and iterative jobs. The methodology performs the comparison by evaluating the architecture components in the MapReduce and Spark frameworks and concludes that spark is much faster with respect to different analytical workloads then MapReduce because of hash-based aggregation component for combiner and RDD based caching.

In this paper [4], authors compare the performance of MapReduce and Spark frameworks by using standard K- Means machine learning algorithms. They mention that though K-Means has random nature of computing, machine learninglibraryofSparkdidthelearningmuchbetterthanthe Map Reduce in terms of processing time. Their results show that Spark executes 3 times faster than theMapReduce.

Authorsofthispaper[5],usesHealthCarestreamingdatafor analyzing the performance of Map-Reduce framework. A new task-level adaptive Map-Reduce frameworks have been introduced. They have used smoothing and Kalman filter to estimate the workloadcharacteristics.

Here [6], the author shows the Hadoop, Map Reduce, and HDFS in the view of the developer. The aim was to build a framework to sustain a load of scalability and fault-tolerant Hadoop MapReduce programming worldview and HDFSare progressively being utilized for handling expansive and unstructured informational indexes. Hadoop empowers communicating with the MapReduce programming model while concealing the multifaceted nature of conveying, designing and running the product parts in general society or private cloud. Hadoop empowers clients to make a group of ware servers. MapReduce has been displayed as an autonomous stage as a-benefit layer appropriate for various necessity by cloud suppliers. It likewise empowers clients to comprehend the information preparing andinvestigating.

In this paper [7], give an investigation of the entanglements of current theoretical execution techniques in MapReduce. They exhibit scenarios which affect the performance of the strategies: data skew, errands that begin non-concurrently, uncalled for configuration of stage rate and sudden asset rivalries. In view of the examination, they have built up another theoretical execution procedure called MCP to deal with these situations. MCP considers the cost execution of group processing assets, going for diminishing the activity execution time as well as enhancing the bunch throughput. Their analyses demonstrate that: MCP can accomplish up to 39% upgrades over Hadoop MCP fits well in both heterogeneous and homogeneous situations; MCP can deal with the information skew case well; MCP is very versatile, which performs exceptionally well in both little bunches and substantial groups; MCP has less overhead than Hadoop-LATE and can be effortlessly executed into new forms of Hadoop.

Authors here [8], technique Ant can be stretched out to differentstructures,forexample,Spark,however,someextra exertion is required. Not quite the same as Hadoop, which executes singular undertakings in independent JVMs, Spark utilizes agents to have various assignments on laborer hubs. To stretch out Ant to Spark, they have to powerfully change agentsizeswithoutrestartingapropelledwork.Sincerunning Spark on another nonexclusive group administration middleware, for example, YARN, turns out to be progressively mainstream, it is conceivable to empower flexibleagentsutilizingassetcompartments.Allthings considered, Ant can screen the fruition times of individual undertakingsandutilizesuchdataasinputtodecidetheideal size of Sparkagents

Here[9]inthispaper,theymentionaboutthemostinfluential articlesaddedtotheenhancementsinMapReducesystemfor extensivedatasetsislikewiseinvestigatedinviewofthemost influential articles chose from papers cover the period 2006–2015 and depicted reception and arrangements that expect to reduceaportionoftheissues.Foreachpart,theyhavepresent the general foundation, examine the specialized difficulties, and audit the most recent advances. Explored a few open

ISSN: 2278-4632
Vol-11 Issue-02 2021

research challenges, including Energy efficiency, Resource assignment, handling enormous information in distributed computing, ongoing preparing, stack adjusting,mapping.

Authors of this paper [10], As a greatly parallel handling structure, MapReduce is very much perceived for its scalability, flexibility, fault tolerance and several other attractive features. It encourages parallelization of a class of uses, usually alluded to as embarrassingly parallelizable. Be that as it may, as has been normally recognized, MapReduce has not been intended for expansive scale complex information administration undertakings. For instance, the first system does not give abnormal state dialect bolster that is natural to and expected by database clients; thus, clients need to exclusively develop various processing logics and programs.Italsodoesnothavebuilt-inindexingandquestion streamlining bolster required for database inquiries. This

has normally driven to along stream of research that attempts to address the lack of database functionality. In this study, our emphasis is on the upgrade and expansion of MapReduce framework for databaseapplications.

4. CONCLUSION

Spark has fantastic execution and is very financially savvy because of in-memory information handling. It's good with much of Hadoop's information sources and record organizations, and because of amicable APIs that are accessible in a few dialects, it likewise has a speedier expectation to learn and adapt. Start even incorporates chart handling and machine-learning abilities.

Hadoop MapReduce is a more develop stage and it was worked for bunch preparing. It can be savvier than Spark for genuinely Big Data that doesn't fit in memory and furthermore because of the more prominent accessibility of experienced staff. Besides, the Hadoop MapReduce biological community is as of now greater because of numerous supporting ventures, devices and cloud administrations.

In any case, regardless of whether Spark resembles the enormous victor, the odds are that you won't utilize it all alone—despite everything you require HDFS to store the information and you might need to utilize HBase, Hive, Pig, Impala or other Hadoop ventures. This implies regardlesswe will have to run Hadoop and MapReduce nearby Spark for a full Big Databundle.

5. REFERENCES

[1] Bo Pang and Lillian Lee: “Opinion mining and sentiment analysis” Foundations and Trends in Information Retrieval Vol. 2, No 1-2 (2008)1–135.
[2] KatarinaGrolinger,MichaelHayes,WilsonA.Higashino, Alexandra L 'Heureux, David S. Allison “Challenges for MapReduce in BigData”
[3] Juwei Shi, Yunjie Qiu, Umar Farooq Minhas, LimeiJiao, Chen Wang, Berthold Reinwald, and Fatma Ozcan “Clashof the Titans: MapReduce vs. Spark for Large Scale Data Analytics”
[4] SatishGopalani,RohanArora,“ComparingApacheSpark and Map Reduce with Performance Analysis using K-Means”. International Journal of Computer Applications (0975 – 8887) Volume 113 – No. 1, March 2015
[5] Balaji Palanisamy, Aameek Singh, Ling Liu “Cost- effective Resource Provisioning for MapReduce in aCloud”
[6] Qi Chen, Cheng Liu, and Zhen Xiao “Improving MapReduce Performance Using Smart Speculative ExecutionStrategy”
[7] Dazhao Cheng, Jia Rao, Yanfei Guo, Changjun Jiangand Xiaobo Zhou “Improving Performance of Heterogeneous MapReduce Clusters with Adaptive TaskTuning”
[8] Ibrahim Abaker Targio Hashem, Nor Badrul Anuar, Abdullah Gani,Ibrar Yaqoob,Feng Xia, Samee Ullah Khan “MapReduce: Review and openchallenges”
[9] FENG LI, BENG CHIN OOI, M. TAMER ÖZSU, SAI WU “Distributed Data Management UsingMapReduce”
[10] Fan Zhang, Junwei Cao, Samee U.Khan, Kequi Li, Kai Hwang “ A task level adaptive Map-Reduce framework for real-time streaming data in health careapplication.”