Big Data Processing-Beyond Batch Processing

S.Anuradha, L.Srinivasa Rao, G.Raghu Ram

Abstract: ***This paper mainly focus on analysis of large sets of students data with one of the batch processing analysis techniques Beyond batch process, analysis of data streaming is done based on program of word counting program which executes data with HDFS along with dynamic created data. To compute similar coherent strategies one can implement a schema named batch and streaming process which dynamically creates data. The architecture is reduced to serve as X-Platform which uses ample number of tools for batch and stream analysis on this proposed frame work. Here we use spark-sql, a query language which acts as interface for interactive process to have iterative processes. Real time streaming data processing involves spark streaming works. Here we focus on preliminary evaluation of results and analysis report which compares data sets performance and also achieve low latency rate with usage of RDD.***

Keywords :HDFS, Coherent strategies, X-Platform, SQL-Spark.s.

I. INTRODUCTION

In this modern world, data is being emerging like a huge storm. Wide range of technologies were introduced to handle these storms of data which is rapidly growing Big Data world over past decade. Among that several challenges were bought by cluster environment. To handle these challenges ample number of frame works have been introduced. One of frame works is Map Reduce which is invented and introduced by Google to execute extremely large applications as clusters on commodity machines. These frame works collaborates with huge sets of data in a reliable, scalable way which also handles fault tolerance too.

II. BACK GROUND

Hadoop is one among frameworks used in distributed processing of data sets of larger sized around commodity computers clusters of programming models which are simple[12]. Basically Hadoop clusters be implemented using single servers through ample number of machines in which each have capacity of local computation along with storage which comprisies of kinds of data which are different. This kind of ecological system includes with HDFS, Impala,

HBase, Pig, HIVE, Map Reduce ect.,. Hive is a Query Language with ad-hoc queries and is a database which is placed on top of Hadoop[13]. It splits tasks which are parallel using Map Reduce to perform huge stored of penta bytes, which also maintains meta store for created each table in which stores scheme and location.

As immutable data and streaming data's rapid growth, the demand for efficient new methods is also growing which is a key challenge in world of Big Data.

With historic and real time data low latency analysis performance is one of the major part of it. Spark is another platform which performs distributing along parallel computing of analysis as well as storing such data.[1][2][10]. Spark is another platform supports working sets of applications which provides similar fault tolerance and scalability of Map Reduce. Spark implemented in Scala with aim as Resilient distributed data sets(RDD) [3][8][11] which will exists as partitions of cluster across machine sets which can be retained back if it is lost. Computing users who uses memory cluster can cache a RDD among the machines which can reuse multiple times for interactive operations performance for all parallel operations.

III. PROBLEM DEFINITION AND ORGANIZATION

Hive is designed for ad-hoc queries but not sub queries and also not meeting the real time query in Big data demands as output produced from one task becomes input for other task.

Map Reduce persists the complete data set to HDFS after executing each job which is considered as limitation of Map Reduce. In this scenario there is possibility of encounter high latency using huge disc access, I/O's of number of computations going on.

Above one can be handled/overcome by Spark which is a pipe line operation in perception of holistic view When output of one operation is needed as input to another operation, it accesses data directly without intimating existing persistence storage.

Major feature of Spark is memory abstract caching (RDD) which marks Spark as ideal for workloads same data where input data access for multiple operations[4][5]' using Spark users can activate cache input data set from memory, for this reason they didn't retrive from disk for every task. Spark executes completely in memory unit, where Map Reduce has to serialize and write out put records on map side later given as input along with serialized on reduce side.

Topics from Work is organized as over view of the system, Frame work for the proposed system, Experimental analysis and conclusion respectively in the next sections.

Revised Manuscript Received on November 15, 2019

***** Correspondence Author

***S.Anuradha,**PG Student,CSE Dept., Mother Teresa Institute of Science & Technology, Sathupalli, Khammam Dt., JNTU, Hyderabad, Telengana State, India.*annuraddha29@gmail.com*

L.Srinivasa Rao, [2]Assoc.Professor CSE Dept., Mother Teresa Institute of Science & Technology,Sathupalli, Khammam Dt., JNTU Hyderabad, Telengana State, India. *Srinu.pha4@gmail.com*

G.RaghuRam, Assoc.Professor & PRO CSE Dept., G.Pulla Reddy Engineering College,Kurnool, Andhra Pradesh, India,*gprecpro@gmail.com*

IV. SYSTEM OVERVIEW

Hadoop provided as general frame work to build different types of applications of Big Data for analysis purpose, which provided with an open source implementation of Java Map Reduce. Consisting of two phases one for data storage which is (HDFS) Hadoop Distributed File system and other for data processing where Map Reduce is used. Number of tasks are scheduled and different algorithms are used to process different jobs. For repeated operations Map Reduce tasks cannot continue sharing of data frequently as data has to be stored in HDFS and write to it again which implements huge amount of accessing I/Os of disck and ample number constructing computations.

To overcome the above mentioned problem Apache Spark is designed which consists of cluster computing frame work. Hadoop and Spark are compatible to each other as Spark uses all properties of it. For cashing data new concept was introduced by Spark which is named RDD(Resilient Distributed Dataset). Moto of RDD to built to handle application of currently computing frame works ineffectively. In order to increase performance rapidly tools for mining data and iterative algorithms are being kept in memory. RDD is partitioned for collecting only readable records. When users want to reuse and choose data storage RDD can create persistence storage strategy. Spark always keep these RDDs present in memory by default and also able to partioned them on disk, if it doesn't fix in RAM[1][3]. If you do any changes to existing RDDs, it will create new RDDs. In proposed system RDDs were implemented without performance of Hadoop for repetative applications and be extended for different data queries with gigabyte as capacity.

V. PROPOSED WORK'S FRAME WORK

Our proposed system is a combination of Batch, Interactive and streaming process. For Speed and time computations latency rate minimizing, output of one task will be given as input of other task which is prescribed as required architecture of our proposed work. Which is of two parts for persistence data we use batch processing and for iterative operations use interactive along streaming process.

All the above mentioned are built on single platform named as Spark which consists of inbuilt Java APIs programming along with Scala along Pytho. Spar. A DAG visualization is provided with Spark of the operations for better flow performance and analysis.

A.Bath Processing

Batch processing is depictedas following figure1. In batch processing can handle incoming data of different type where it is persistently stored. Rigid data (which cannot be change as time goes on) will be any platform like Spark input or Hadoop where data partitioned into small pieces to execute computations which are served as an output.

In the proposed work, to process immutable data we use HiveQL and SparkQL. Different query operations were performed on the data to present analysis report. In case of failures in computing process, it reprocess the process tp compute again and finally serves the required output.

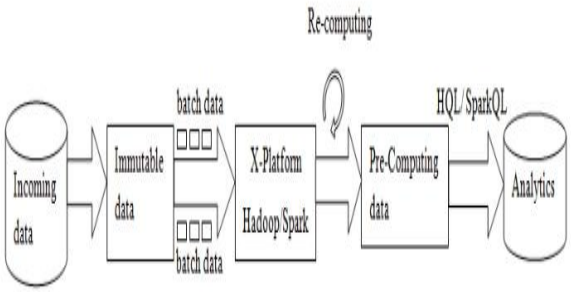

Figure1: Batch Processing

Interactive and Streaming Process

Beyond Batch processing, now discussing about interactive and streaming data analysis. In the proposed work all interactive operations are done with rigid data using Spark SQL which plays major role in processing of data residing at memory, where data is stored in the form of RDD to perform various tasks and transformations. Data takes the form of objects while storing on RDD which is static in nature, different query operations were being performed. On the final data, one performs different interactive sessions with performing number of actions and transformations. If user want to perform any action on the data stored on RDD persistently, current RDD is not going to change where as it creates different RDD for the purpose.

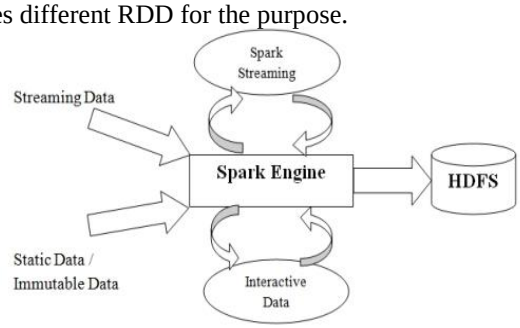

Figure2: Interactive and Streaming Process

B. Work Platform for proposed system

Single cluster concept is being used in the experiments of the proposed system. Windows 8.1 is the OS used with 64 bits with BIOS virtualization is enabled.

Set up is created with Spark and Hadoop being paired and configured in Linux based OS cloud-era.

CDH5.5 package is installed to provide latest virtual box which is a frame work of number of different tools, which consists of Hadoop of version 2.6 and Spark of version1.5.2. Scala with version 4.3.0 Vfinal with version-2.11-Linux along with Spark libraries are imported to eclipse for Streaming process of program. 8GB RAM with 50GB Hard disc are the Hardware used as the Hardware for proposed work.

Whole package is set-up to execute SparkQL, HiveQL and streaming data process

C.Design of Data base in the proposed work

Here we deal with datasets of students who are pursuing their under graduation and Post graduation courses belonging from different academic years. Tables are used as data structures which consists of different attributes like Name of the student,

College code, University id, Course, Department, branch, address and so on.
Proposed application is designed such a way that it analyze the count of the students of UG and PG courses studied in specific subject and as well as it analyzes year wise report.
These generated reports is useful for various other statical report to corresponding city, state and country where total number of students graduated each corresponding year. One can analyze rate of literacy, proficiency of students merits in particular stream which impacts on a particular country's development in various fields.

VI. EXPERIMENTAL ANALYSIS

A.Batch Queries

For Batch Processing, files will be saved in local file system
A student data set which is in the form of CSV is created our systemproposed which is stored locally. SparkQL is used for batch processing on dataset. Different query based operations are inserted to get expected results
Tables were created using Spark Query Language and loaded. Loaded tables were retrived by "SELECT * FROM table name" where result stored in RDD as "Val object name =sql context.sql("SELECT * FROM….)
One can perform number of actions on it using RDD like "object name GROUPBY("subject").count().show()" where it gives the count of subjects studied by the students.

B. Interactive Queries

Text files were used for execute Interactive operations and various actions are formed to produce the transformed result. Initiate file from hhdfs://…")".
Operations which are interactive are performed with text file where various actions are converted to produce the result.
Retrieve file from localfile system save text file to hdfs using RDD "val rddl = sc.textFile(" cal system and stored/saved text file flat map and reduce the text file pair to <key, value> then stored result on RDD as "Val rdd2= rdd. Flat map(line=> linesplit(")).map(word=>word1).reduceBykey((a,b)=>(a+b) " resultant word count is retrieved using rdd2.show() operation

Using RDD interactive operations were performed on it. As mentioned earlier object rdd2 stores out/result of operation performed. Various actions can be done using that object like rdd2.first(), rdd2.count(), rdd2.collect(), rdd2.filter() and so on. As mentioned earlier output will stored in object, now this output will be fetched as input for another operation. Spark frame work is considered as best interactive data processing.

Next step is analysis of streaming data. Here Scala program is created to access the real time data streaming.
Here we analysed two types of streaming data
1)Streaming of dynamically created data
2)Streaming data from HDFS and then store the result tphdfs.

Word count program is performed with connection to local server and data time for streaming is set to 10 sec where one can create data dynamically.

C.Analysis of experimental results

Here students both UG and PG data sets are analysed based on subject wise and year wise. Figures 3 and 4 depicts the analysis of data of Computer science subject wise analysis of UG and PG during specific academic year.
Report generated based on dataset used which depicts result of PG students. Here status of literacy rate of UG students, which depicts whether it is increasing or decreasing year wise. This survy helps to take towards to raise literacy rate on the respective fields.

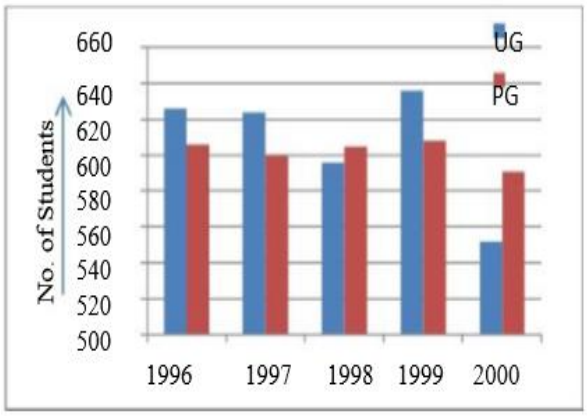

Figure3. Data Analytics

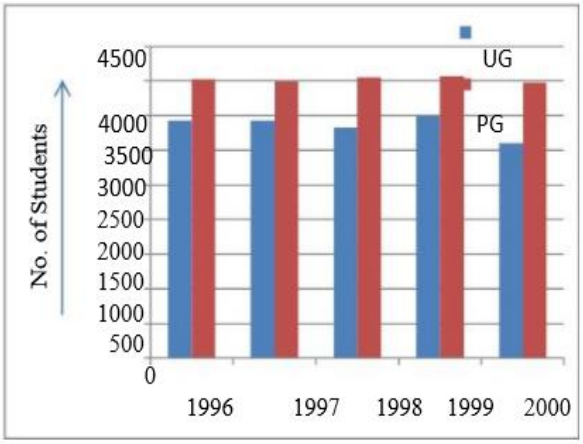

Figure 4. Computer Science

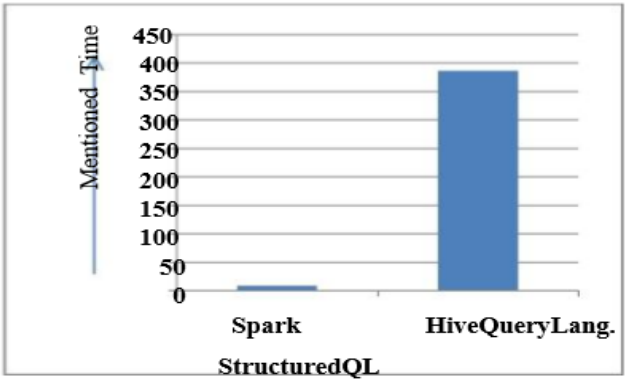

Figure5. Spark's and Hadoop's Latency

Figure 5, depicts the Hadoop and Spark performance analysis. Here multiple join operations on the data sets of students using Spark Structured Query Language and HiveQuery Language parallel to retrieve the data. From the above figure one can conclude that HiveQL having more latency than SparkSQL which implies that Spark requires less time to execute its operations than Hive

VII. CONCLUSION

Big Data has created a new trend and effective solutions have been provided. Hadoop is well known which is one among those solutions. It is designed specifically for batch processing and executing high through put jobs. It is also suitable for jobs with large volumes of data in longtime. Hadoop can not only process streaming data but also more efficient and reliable.

In the proposed work we also used Spark as it works effectively using RDD. Due to high requirement of interactive queries and Big data handles both streaming data and real-time data requirements. Spark serves as very good example for the cases which supports memory computing using RDDs Performance analysis have been done for Hadoop and Spark.

REFERENCES

1. Altti Ilari Maarala, Mika Rautiainen, Miikka Salmi, Susanna Pirttikanges and Jukka Riekki "Low latency analytics for streaming traffic data with Apache Spark" 2018
2. Lei Gu, Huan Li, "Memory or Time:Performance Evaluation for Iterative Operation on Hadoop and Spark ", High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing (HPCC_EUC), 2013 IEEE 10th International Conference on Date of Conference:13-15 Nov. 2013Page(s):721 – 727.
3. "Spark: A Big Data Processing Platform Based on Memory Computing" by Zhijie Han, Yujie Zhang, 2015-IEEE , DOI: 10.1109 /P AAP .2015.41
4. M. Satish Gopalani, Rohan Arora "Comparing Apache Spark and Map Reduce with performance analysis using K-Means" international Journal 2015.
5. "Performance comparision of hive, impala and spark SQL" by Xiaopeng Li, Wenli Zhou, doi:10.11109/IHMSC.2015.95
6. "Performance Prediction for Apache Spark Platform" by Kewen Wang, Mohammad Maifi Hasan khan @ 2015 IEEE 17th international conference on HPCC.
7. Xinyi Liao, Zhiwei Gao,Weixing Ji, Yizhuo Wang "An Enforcement of Real Time Scheduling in Spark Streaming" doi:10.1109/IGCC.2015.739373
8. "Streaming Twitter data analysis using spark for effective job search" Lekha R.Nair , Sujala D. Shetty , Journal of Theoretical and applied information technology -2015
9. "Lambda Architecture for cost-effective Batch and speed big data processing" by Mariam Kiran, Peter Murphy, Inder Monga, Jon Dugan, 2015-IEEE,doi:10.1109/BigData.2015.7364082
10. M. Zaharia, M. Chowdhury, S. S. Michael J. Franklin, and I. Stoica, "Spark: Cluster computing with working sets," *In HotCloud*, June 2010.

AUTHORS PROFILE

Dr. Anuradha Sanike obtained her PhD in Computer Science and Engineering from Sri Krishna Devaraya University, Ananthapur, Andhra Pradesh in 2011. Persued her B.Sc, bachelors and MCA masters degrees from Osmania University, Hyderabad in the year 1994 and 1997 respectively.
She had 14 years of teaching experience. She is awarded Post Doctoral Fellowship from University Grants Commission, New Delhi, India in 2015.
She is pursuing her M.Tech(2018-2020) in Computer science at Mother Teresa college of Science and Technology affiliated to JNTU Hyderabad.
She Published 23+ Papers in Reputed International Journals, International and National conferences

L.Srinivasa Rao Persuing PhD from JNTUH, Graduated B.Tech from Lakireddy Balireddy college of Engineering JNTUH in 2004 and Post Gaduated (M.Tech) from JNTUH in 2008. He had 10 years of teaching experience.
He is working as Associate Professor in CSE Dept at Mother Teresa college of Engineering, Sathupalli, Khammam Dt. He Published 9 papers in reputed International, National Journals and Conferences. His Area of Interest is Cloud Computing and Computer Networks.

Dr.G. Raghuram obtained his PhD from Rayalaseema University, Kurnool in 2017. Graduated from Sri Krishna Devaraya University in the year 1994, MCA from Osmania University in the year 1997 and M.Tech from Manav Bharti University in the year 2014.
He had 20+ years of teaching experience.
He is working as Associate Professor and Public Relations Officer at G. Pulla Reddy Engineering College, Kurnool, Andhra Pradesh, India. He presented 24+ research papers in various International Journals and national, International conferences. His research areas include Artificial Intelligence and Computer Networks.