Integrated Intelligent Research (IIR)
International Journal of Data Mining Techniques and Applications
Volume: 04 Issue: 02 December 2015 Page No.78-81
ISSN: 2278-2419

A Survey on Analyzing and Processing Data Faster Based on Balanced Partitioning

Annie .P. Kurian[1], V. Jeyabalaraja[2]
[1]Student, Dept. of Computer Science & Engg, Velammal Engg College, Chennai, India
[2]Professor, Dept. of Computer Science & Engg., Velammal Engg College, Chennai, India
Email: anniekurian13@gmail.com,jeyabalaraja@gmail.com

Abstract---Analyzing and processing a big data is a challenging task because of its various characteristics and presence of data in large amount. Due to the enormous data in today's world, it is not only a challenge to store and manage the data, but to also analyze and retrieve the best result out of it. In this paper, a study is made on the different types available for big data analytics and assesses the advantages and drawbacks of each of these types based on various metrics such as scalability, availability, efficiency, fault tolerance, real-time processing, data size supported and iterative task support. The existing system approaches for range-partition queries are insufficient to quickly provide accurate results in big data. In this paper, various partitioning techniques on structured data are done. The challenge in existing system is, due to the proper partitioning technique, and so the system has to scan the overall data in order to provide the result for a query. Partitioning is performed; because it provides availability, maintenance and improvised query performance to the database users. A holistic study has been done on balanced range partition for the structured data on the hadoop ecosystem i.e. the HIVE and the impact on fast response which would eventually be taken as specification for testing its efficiency. So, in this paper a thorough survey on various topics for processing and analysis of vast structured datasets, and we have inferred that balanced partitioning through HIVE hadoop ecosystem would produce fast and an adequate result compared to the traditional databases.

Keywords – range aggregate, big data, HIVE, HDFS, Map Reduce.

I. INTRODUCTION

Due to the enormous data in today's world, the traditional system mechanism becomes difficult to handle these large amounts of data. The traditional approaches are meant for handling structured data only to a certain size limit. Also, the processing becomes unmanageable and non-scalable after certain amount. So, that is when Big Data came into existence, which handles huge amount of data.

Big data is a broad term for data sets so large or complex that traditional data processing applications are inadequate. Challenges include analysis, capture**,** search, sharing, storage, transfer, visualization, and information privacy. Big Data is meant for "its sheer Volume, Variety, Velocity and Veracity. Big data includes handling of structured data, semi-structured and unstructured data. But the traditional system is meant only for handling the structured data. The range aggregate query in the existing system has row-oriented partitioning and has limited processing of queries. The main focus of big data is how the data are analyzed and retrieved according to the efficiency and accuracy. In today's Map Reduce framework is used for processing OLAP and OLTP. Hadoop is an open source framework that allows to store and process big data in a distributed environment across clusters of computers using simple programming models. HDFS is a distributed file system (DFS) which is highly – fault tolerant and provides high throughput access to application data and is highly suitable for large data sets. The programming model used in Hadoop is Map Reduce (MR). Map Reduce is the data processing functionality used in Hadoop which includes breaking the entire task into two parts, known as map and reduce.

In the traditional system, handling of large datasets is deficient in producing accurate and fast response. Some of the features required to overcome the disadvantages of existing system are:

- Analyzing and retrieving range aggregate query based on balanced partitioning i.e. dividing a big data into independent partitions and then generating a local estimation for each partition.
- The results retrieved should be faster and accurate.
- Fault-tolerance.
- Efficient response with minimum delay.

II. RELATED WORKS

A. Range queries in dynamic OLAP data cubes

Online Analytical Processing (OLAP) is a technology that is used to organize large business databases and OLAP application is a multidimensional database which is also called as data cube. Range query applies an aggregation operation (SUM) over selected OLAP data cubes, which provide accurate results when data's are updated. This paper mainly focuses on performing aggregation operation with reduced time complexity. The algorithm used in this paper is double relative prefix sum approach to speed up range queries from $O(n^{d/2})$ to $O(n^{d/6})$, where d is the number of dimension and n is the number distinct tuples at each dimension [5]. The pre-computed auxiliary information helps to execute ad-hoc queries. The ad-hoc query is used to obtain information only when the need arises. So the prefix some approaches are suitable for the data which is static or which are rarely updated. A research is being done for updating process, i.e.by providing simultaneous updating. Also, the time complexity for data updates has to be reduced for range aggregate queries.

B. You can stop early with COLA: Online processing of aggregate queries in the cloud

COLA is used for answering cloud online aggregation for single and multiple join tables with accuracy. COLA supports incremental and continuous aggregation which reduces the waiting time before an acceptable estimate is reached. It uses Map Reduce concept to support continuous computing of

International Journal of Data Mining Techniques and Applications
Volume: 04 Issue: 02 December 2015 Page No.78-81
ISSN: 2278-2419

aggregation on joins which reduces the waiting time. The techniques used in [8] OLA are block –level sampling and two-phase stratified sampling. The block level sampling consists of data which is split into blocks which forms samples. The stratified sampling, divides the population/data into separate groups, called strata.

Then, a probability sampleis drawn from each group. The efficiency in terms of speed is improved when compared to existing system efficiency. The OLA provides early estimated returns when the background computing processes are still running the results are subsequently refined and the accuracy is improved in succeeding stages. But users cannot obtain an appropriate answering with satisfied accuracy in the early stages. Also it cannot respond with acceptable accuracy within desired time period.

C. HIVE- A petabyte scale data warehouse using Hadoop

Due to large data sets, traditional warehousing finds it difficult to process and handle the data. It is also expensive and produces non-scalable and inadequate result. In this paper, the authors have explored the use of HIVE an ecosystem of Hadoop. Hadoop is an open-source framework that allows to store and process big data in a distributed environment across clusters of computers using simple programming models and process model(MR). Map-reduce require hard maintenance, so the proposal system introduces a new framework called Hive on top of Hadoop framework. Hive is a data warehouse infrastructure built on top of Hadoop for providing data summarization, query, and analysis. Hive is a DB engine, which supports SQL like queries called Hive QL queries, which are then compiled into MR jobs [1]. Hive also contains a system log- meta store-which contains schema and semantics becomes useful during data exploration. HIVE ecosystem is best suitable for processing data, which is also used for querying and managing structured data built on top of Hadoop.

D. Online aggregation for large Map Reduce jobs

This paper proposes a system model that is applicable for online aggregation analytic (OLA) over map reduce in a large scale distributed systems. Hyracks is a technique involving pipelining of data processing which is faster than staged processing of hadoop process model [7]. Master maintains random block ordering. It consists of two intermediate set of files i.e., data files which stores the values and the metadata files which stores the timing information. For estimation of aggregates- Bayesian estimator- uses correlation information between value and processing time. This paper destroys locality and hence introduces overhead. Since this approach is suitable only for static data, where updating of files becomes difficult and hence the result produced is inaccurate.

E. Online Aggregation and Continuous Query support in Map Reduce

Map Reduce is a popular programming function in the Hadoop open – source framework. To resolve the issue of fault tolerance, this paper allows the output of the MR task and job function to be materialized to the disk i.e. the HDFS before it is consumed. The reformed version of Map Reduce allows data to be pipelined between operators [3]. A pipeline is a set of data processing elements which is connected in series, where the output of one element is sent as the input of the next one. So, this improves and reduces the completion time and system utilization. MR supports online aggregation. Hadoop Online Prototype (HOP) supports continues queries, where programs written as event monitoring and stream processing, thus retaining fault tolerance.

The drawback is that the output of map and reduce phase is materialized to stable storage before it is consumed by the next phase. Though the users consume results immediately, the users obtain results which are not appropriate and accurate. Also, the continuous query analysis processing is inefficient, because each new Map Reduce job does not have access to the computational state of the last analysis run, so the state must be recomputed from scratch.

F. Range Aggregation with Set Selection

The partitioning is done in large datasets in order to consume less time for processing the data and to provide output which is flexible and efficient. Range aggregation can be of any operation such as COUNT, SUM, MIN, MAX etc. The partition can be done based on different perspective such as:

- Random partition
- Hash partition
- Range Partition

Random partition is where the data's are randomly partitioned based on the size of the dataset. Hash partitioning of the data is based according to hash key value and the hash algorithm. Hash partitioning is a technique where a key value is used to distribute/partition rows equally across different partitions. Range partitioning separates the data according to range of values of the partitioning key. In range partition when the object resides on two different sets, then the operation has to be done separately on the sets and the resultant set are combined to produce the final output, based on the intersection operation. This type of process is called a range partition set operation [9]. The random partitioning might be easy to implement, but then some data's or objects maybe dependent on the other in order produce the data, so the processing of the results takes a longer time. So the range partitioning method is used in which several data are partitioned based on certain criteria and processing and analysis are done simultaneously. Therefore, it produces result with accuracy and it consumes linear space and achieves nearly-optimal query time, since the processes are in parallel.

G. Hyper Log Log in practice: Algorithmic Engineering of a State of the art Cardinality Estimation Algorithm

Hyper Log Log is an algorithm for the count-distinct problem, approximating the number of distinct elements in a multi set. Calculating the exact cardinality of a multiset requires an amount of memory proportional to the cardinality, which is impractical for very large datasets. This paper introduces algorithm for selection of range cardinality queries. The Hyper Log Log ++ algorithm is used for accurately estimating the cardinality of a multi set using constant memory [4].Hyper Log Log++ has multiple improvements over Hyper Log Log, with a much lower error rate for smaller cardinalities. It serializes the hash bits to bytes array in each bucket as a cardinality estimated. The Hyper Log Log++ uses 64- bit hash function instead of 32- bits in Hyper Log Log in order to improve the data-scale and estimated accuracy in big data environment. Hence this algorithm decreases the memory usage; accuracy increased and also reduces error.

International Journal of Data Mining Techniques and Applications
Volume: 04 Issue: 02 December 2015 Page No.78-81
ISSN: 2278-2419

H. Fast data in the era of big data: Twitter's real – time related query suggestion architecture

The big data as significant increase in data volume, and the preferred tuples maybe located in different blocks or files in a database. On the other hand, real time system aims to provide appropriate results within seconds on massive data analysis. This paper presents the architecture behinds Twitter real- time related query suggestion and spelling correction service. It describes two separate working systems that are built to solve the problem. They are: 1) Using hadoop implementation,2) Using in-memory processing engine [6]. A good related query suggestion should provide Topicality and Temporality. Using hadoop, Twitter has robust and production Hadoop cluster. It uses Pig which is a scripting language to aggregate users search session, compute term and co-occurrence statistics. Due to this hadoop causes two bottlenecks that are log import and latency problem. Then the in-memory processing stores query co-occurrence and statistics, it stores the results in HDFS after the process and also contains replication factor. It becomes easier to fetch data from in- memory because it reduces the time complexity, and therefore it improves the efficiency.

I. Effective use of Blocks- Level Sampling in Statistics estimation

Due to the large datasets i.e. petabyte of data's, it becomes hard to scan the whole data and retrieve the result and also it s expensive. In order to overcome this, approximate statistics are built based on sampling. Block level sampling is more productive and efficient than the uniform-random sampling over a large datasets, but accountable to significant errors if used to create database statistics. In order to overcome it, two approaches are used Histogram and Distinct- Value estimation. Histogram uses two phase adaptive method in which sample size is decided based on first phase sample, and this is significantly faster than previous iterative methods.

The 2-phase algorithm consists of the sort and Validate mechanism, and from it the required histogram is merged. The distinct value estimates appears as part of Histogram, because in addition to the tuple counts in buckets, histogram also keeps a count of the number of distinct value in each bucket [2]. This gives a density measure, which is defined as the average number of duplicates for the distinct value. The bucket weight is returned as the estimated cardinality of query.

III. COMPARISON BETWEEN RDBMS AND BIG DATA

A comparison is made between the traditional RDBMS and the emerging technology i.e. big data, to show which is efficient in handling the structured data. This shows that the big data has various advantages when compared to RDBMS.

CRITERIA	RDBMS	BIG DATA
Description	Used for transactional system, reporting and archiving	Stores and process data in parallel
Data Type	Structured	Structured, Unstructured and semi-structured
Availability	Not a Fault tolerant system	Fault tolerant
Storage	Stores the data in rows and columns	Stores in Distributed File System (HDFS)
Data Size	Terabyte	Petabyte
Read/Write limits	1000 query/sec	Millions query/sec
Transaction/ Processing speed	Takes more time to process and also not that accurate and efficient	Fast and efficient

IV. SYSTEM MODEL

Many organization and industries generate large amounts of data in today's world which consist of both the structured and unstructured data. Handling big data using the open- source Hadoop framework would consist of using HIVE for analyzing and processing structured data with the help of Map Reduce function.

Range-aggregate queries for structured data execute the aggregate function depending on number of rows/columns simultaneously in a given query ranges. The processing of large amount of data in the traditional system is difficult, since it processes only limited set of tuples and also takes a long time to access and generate the accurate result. Therefore to efficiently handle the query on big data, we would use the range partition/ aggregation mechanism. The data are partitioned in such a way that the searching and processing of data becomes easier.

According to large data record field, the big data is partitioned and then a local estimation sketch is generated for each partition. When a user enters range-aggregate query according to requirements, the system quickly try to fetch the respective data from the respective partitions, instead of scanning the whole data. This method tries to bring a holistic improvement in handling structured data with accuracy, availability, consistency and with fast response.

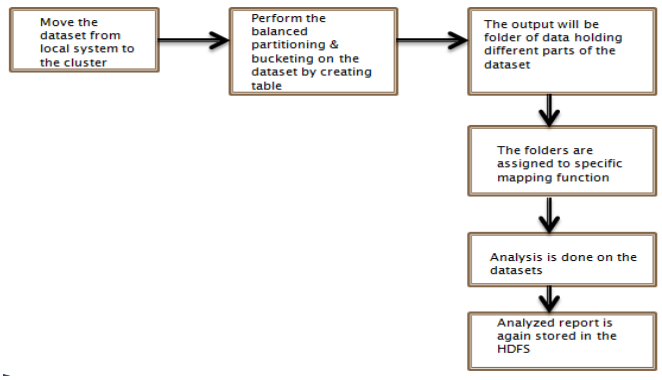

Fig1. Block Diagram

V. PROBLEM SCENARIO

Since the traditional relational database finds it difficult to handle the large amount of data, therefore the system is

International Journal of Data Mining Techniques and Applications
Volume: 04 Issue: 02 December 2015 Page No.78-81
ISSN: 2278-2419

inadequate, non-scalable, causes relative errors and increased time complexity. Here, an idea is introduced, which uses balanced partitioning on range aggregate queries, which was not that scalable and reliable with the traditional system. The partitioning is the process which is done in order to reduce time in scanning the whole data. Partition is a division of permissible database or its peripheral elements into distinct independent parts. This is normally done for performance, accuracy reasons. The new approach would provide result with accuracy andreduced relative errors when compared to the existing system. Thus the system aims to provide:

- Accurate results.
- To increase the processing speed of range-aggregate query.
- To achieve scalability.
- Updating of data done simultaneously.
- Latency and errors are reduced.

Since our idea uses range partitioning technique in order to process the data, it reduces the time required to access the data and the results are retrieved faster. The Map Reduce is the processing function in hadoop which does the analysis of the datasets and stores the results in the HDFS. The results are stored in a separate folder, in order to reduce the time.

VI. CONCLUSION

In this paper, we have reviewed all the different types of techniques which have its application effective in use for our fast range aggregate queries in big data environment. The traditional system models are non–scalable and unmanageable. Also, the cost increases as the growth of data increases. We would try to equally partition the data and provide sample estimation for each partition, which would make the processing of query easier and effective. The Hive is a data warehouse query language which is used for handling the structured data. The system which we analyzed would produce the data with high efficiency result and with less time complexity for data updates. This system would provide a good performance with high scalability and accuracy. This approach would be good techniques for developing real – time answering methods for big data analysis and processing.

REFERENCES

[1] Ashish Thusoo ,Joydeep Sen Sarma, Namit Jain "Hive – A Petabyte Data Warehouse using Hadoop." *Proc. 13th Int'l Conf .Extending* Database *Technology(EDBT '10)* ,2010.

[2] Chaudhuri .S, Das .G, and Srivastava .U, "Effective use of blocklevel sampling in statistics estimation," in Proc. ACM SIGMODInt. Conf. Manage. Data, 2004, pp. 287–298.

[3] Condie .T, Conway .N, Alvaro .P, Hellerstein .J .M, Gerth .J, Talbot .J, Elmeleegy .K, and Sears .R, "Online aggregation and continuous query Support in Map Reduce," in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2010, pp. 1115–1118.
Heule .S, Nunkesser .M, and Hall .A, "Hyperloglog in practice:Algorithmic engineering of a state of the art cardinality estimation Algorithm," in Proc. 16th Int. Conf. Extending Database Technol.,2013, *pp.* 683–692.

[4] Liang .W, Wang .H, and Orlowska .M .E, "Range queries indynamic OLAP data cubes," Data Knowl. Eng., vol. 34, no. 1,pp. 21–38, Jul. 2000.

[5] Mishne .G, Dalton .J, Li .Z, Sharma .A, and Lin .J, "Fast data in theera of big data: Twitter's real-time related query suggestionarchitecture," in Proc. ACM SIGMOD Int. Conf. Manage. Data,2013, pp. 1147–1158.

[6] Pansare .N, Borkar .V, Jermaine .C, and Condie .T, "Online aggregationfor large MapReduce jobs," Proc. VLDB Endowment, vol. 4,no.11, pp. 1135–1145, 2011.

[7] Shi .Y, Meng .X, Wang .F, and Gan .Y, "You can stop early withcola: Online processing of aggregate queries in the cloud," in Proc.21st ACM Int. Conf. Inf. Know. Manage. 2012, pp. 1223–1232.

[8] Yufei Tao, Cheng Sheng, Chin-Wan Chung, and Jong-Ryul Lee," Range Aggregation with Set Selection," *IEEE transactions on knowledge and data engineering,* VOL. 26, NO. 5, MAY 2014.