ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 6, June 2018

Semi Dynamic Range Aggregate Point Enclosure Based Bigdata Environments

A.Raja[1], Dr. S. Prema[2]

Assistant Professor, Kavitha's College of Arts & Science, Tiruchengodu, Vaiyappamalai Rd, Chinnamanali, Tamil Nadu, India

Assistant Professor, K.S.R. College of Arts & Science, Tiruchengodu, Namakkal, Tamil Nadu, India

ABSTRACT: Big data is a broad term for huge data sets that traditional data processing applications are inadequate. Big data analysis can discover trends of various social aspects and their preferences of individual everyday behaviors .The main challenging factor is processing large amount of data within a time period .The query processing time is increased then the network communication cost and local files scanning cost can be increased simultaneously. In our existing system they use hive in range aggregate query but this could provide inaccurate results in big data environments .To overcome this limitations we propose a new technique called FASTRAQ- Range Aggregate Queries. FastRAQ first divides big data into different independent partitions with a balanced partitioning algorithm, and then generates a local estimation for each partition. If a range-aggregate query request arrives, FastRAQ obtains the result directly by summarizing local estimates from all partitions. Fast Range Aggregate Queries has time complexity of 0(1) for data updates. FastRAQ provides range-aggregate query results within a time period that are lower than that of Hive, while relative error is less than 3 percent within the given confidence time interval.

KEYWORDS: FASTRAQ, Hadoop, HDFS, MapReduce, Hive.

I. INTRODUCTION

Big data is a buzzword or huge dataset used to describe a massive volume of both structured and unstructured data. Big data is so large and this can be difficult to process using traditional database and software techniques. The most enterprise circumstance the volume of data is too big or it moves too fast or it exceeds current processing capacity. In contempt of these problems, big data has the prospective to help in many industries or companies to improve operations and make faster, more intelligent decisions.

Big data needs exceptional technologies to efficiently process large quantities of data within endurable elapsed times. The suitable technologies are crowd sourcing, data fusion and integration, genetic algorithms, machine learning, natural language processing(NLP), signal processing, simulation, time series analysis and visualization. Multidimensional big data can also be signified as tensors that can be more efficiently handled by tensor-based computation, such as multi linear subspace learning. The ancillary technologies being applied to big data include massively parallel-processing (MPP) databases, data mining, distributed file systems, search-based applications, distributed databases, cloud-based infrastructure (storage, applications and computing resources) and the Internet.

managing large datasets occupy in distributed storage .Hive provides a mechanism to assignment structure onto this data and query the data by using a SQL-like language called HiveQL[1]. Hive is a datawarehouseing infrastructure for Hadoop. The primary responsibility is to provide data summarization, query and analysis. The best part of HIVE is that it supports SQL-Like access to structured data which is known as HiveQL (or HQL) as well as big data analysis

II.HADOOP

Hadoop is an open source, Java-based programming framework that supports the processing and storage of extremely large data sets in a distributed computing environment. It is part of the Apache project sponsored by the Apache Software Foundation. It is designed to scale up from single servers to thousands of machines, this will providing local storage and computation. It is more scalable , flexible and simple fault-tolerant mechanism[1][2].

- ***Large datasets*** → Terabytes or petabytes of data
- **Large clusters** → hundreds or thousands of nodes

Hadoop is open-source implementation for Google and it is based on a simple programming model called MapReduce .

Hadoop framework consists of two main layers

- HadoopDistributed file system (HDFS)
- Execution engine (MapReduce)

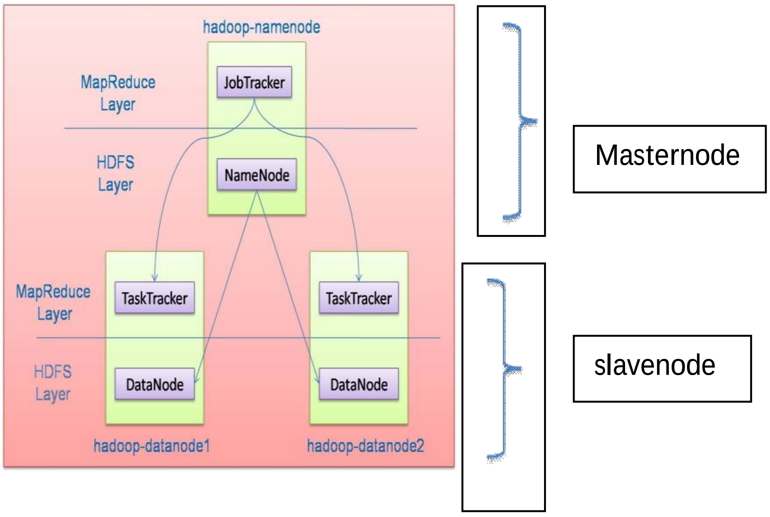


Figure1:Hadoop architecture

HDFS cluster have a single **Namenode**, a master server that manages the file system namespace and regulates access to files by clients. There are number of **DataNodes** usually one per node in a cluster. The DataNodes manages the storage attached to the nodes that they can run on. In fig[1] HDFS reveal a file system namespace and allows user to be store.A file is break into one or more blocks and set of blocks are stored in the DataNodes.

DataNodes can serves the read, write requests, deletion, performs block creation, and replication upon command from Namenode.

III.MAP REDUCE

The MapReduce is the programming model and a framework for data-intensive distributed computing used in batch mode processing. The key idea for the MapReduce model is to allow the users to focus on data processing mechanisms and hide some aspect of parallel execution.The processing of large-scale database systems to obtain the approximate results without complete execution, the online aggregation can construct into the MapReduce system. The MapReduce parallelization and Online Aggregation combined to obtain Time-to-Solutions further on the today's systems[2]**.**

To reduce fault tolerance, the output of each Map and Reduce task materialized to disk before consumed. The altered MapReduce architecture suggests pipelining the data between operators. Apart from the conventional batch-processing model, the MapReduce programming paradigm minimizes the execution time and improves the system utilization for the all types of jobs. This altered MapReduce system supports Online Aggregation, and allows the user to return the results early from a job.

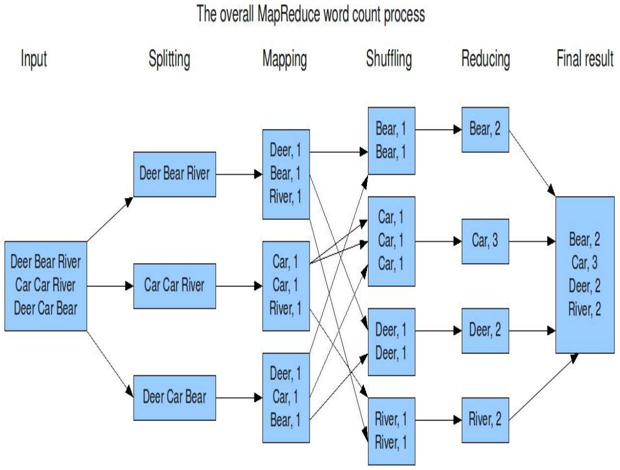


Figure2: MapReduce framework

The MapReduce programming model has developed as a famous way to combine the large clusters of computers together. MapReduce allows the programmers to think of a data-centric model and it mainly emphasis on applying transformations to sets of data records and permits the details of distributed execution, network communication, coordination and fault tolerance to be managed by the MapReduce framework. This programming model finally applied to the large batch processing computation models that centralize mainly on time to job completion. The Google MapReduce framework and the open-source Hadoop system support this model by a batch-processing application strategy. The main advantage of this programming paradigm is the ease of parallelization. The main goal of MapReduce is its fault tolerance and scalability. Apart from its ability, the MapReduce also has some pitfalls contrast to conventional DBMS.

Some of the pitfalls of the MapReduce framework are:

1**) No high-level language:** In Map and Reduce are the functions, the user should code their own operations into it. Hence, it could not support any high-level language any query optimization technique.
2**) No schema and no index:** The MapReduce works only its input is stored in the database. It is a schema-free and index-free paradigm. MapReduce needs parsing each item at reading input and transforms it into data objects for data processing, causing performance degradation.
3**) Single fixed dataflow:** The MapReduce paradigm initially designed to support only single input and generate single output. It could not work for multiple inputs and output. This provides the facility of use with a simple abstraction, yet in a stable dataflow. Hence many complicated algorithms are tough to implement with Map and Reduce functions.
4**) Low efficiency**: MapReduce operations not repeatedly optimized for I/O efficiency. As Map and Reduce are blocking operations, the transition to the following stage not done till all the tasks of the current stage is completed. MapReduce frequently shows poorer performance than DBMS.

IV.FASTRAQ FRAMEWORK

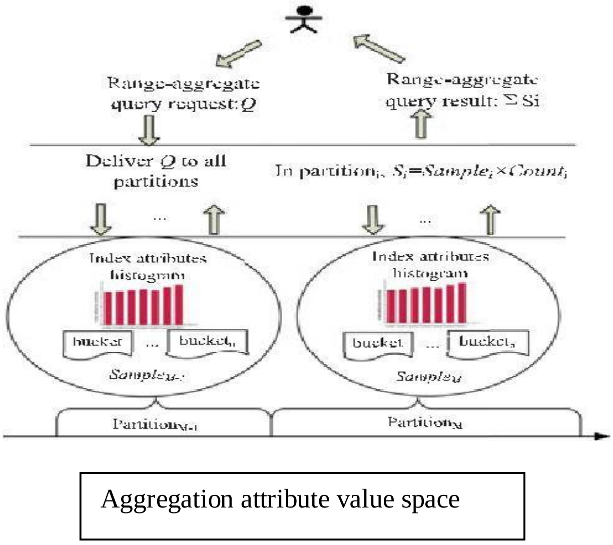
In FastRAQ, the attribute values can be numeric or alphabetic. One example of the range aggregate problem is shown as follows:

Select exp(AggColumn), other ColName where
li1 < ColNamei < li2 opr

lj1 < ColNamej < lj2 opr. . . ;

In the above query, expression is an aggregate function such as SUM or COUNT; AggColumn is the dimension of the aggregate operation; li1 < ColNamei < li2 and lj1 < ColNamej < lj2 are the dimensions of ranges queries; opr is a logical operator including AND and OR logical operations. In the following discussion, AggColumn is called Aggregation-Column, ColNamei and ColNamej are called Index-Columns[3].

Figure3: The FastRAQ framework.


Range-aggregate query request: Q
Range-aggregate query result: Σ Si
Deliver Q to all partitions
In partition$_i$, S_i=$Sample_i \times Count_i$
...
Index attributes histogram
bucket
...
bucket$_n$
$Sample_{M-1}$
Index attributes histogram
bucket
...
bucket$_n$
$Sample_M$
Partition$_{M-1}$
Partition$_M$
Aggregation attribute value space

The distributed range-aggregate queries cost primarily includes two parts. i.e.,the cost of network communication and the cost of local files scanning. The first cost is produced by data transmission and synchronization for aggregate operations when the selected files are stored in different servers. The second cost is produced by scanning local files to search the selected tuples. When the size of a data set increases continuously, the two types of cost will also increase simultaneously. Only when the two types of cost are minimized, can we obtain faster final range-aggregate queries results in big data environments.

V.ALGORITHM USED FOR ANALYZING BIGDATA

A. Pattern Matching Algorithm

In dissecting the information examples imagine a vital part in that each proceed towards information is analyzed. For a set of examples of a set of articles with a particular end goal to focus all possible matches system exploit Rete Match Algorithm.

It stores state data of articles which are matched and somewhat match till the article is show in the memory. There is an another example matching calculation likewise specific example matching which exploit seeking of associated examples in giving content. Knuth-Morris-Pratt is an alternate calculation which is likewise on scanning for examples exploits Java procedures. RE matching and graph calculations are on normal declarations and they give more than one result for a related example[7].

The Brute energy correct example matching calculation uses hunt systems down to find the ex-demonstration information. Applications of this calculation are for parsers, web crawlers, advanced libraries, screen scraper. Different calculations use DFA, punctuation and stable statement for assessment of designs into the straight time protects no reinforcement stream. This RE example matching calculation gives disparate events of examples in the content file. One of the liveliest researched areas of computer science is Patten matches with many papers still being published. This type of category of matching problems includes such as:-

1) Exact string matching: - for example, in a handling issue.
2) Approximate string matching: - for example, in a identification and optical chaste restructure.
3) Largest typical substring

Execution of any matching algorithm is determined, as various string correspondence measures have been constructed that can come close to compare strings and concentrate on a quantitative measure of the level of likeness normally communicated in extends. The accurate pattern matching algorithm is implemented on the basis of searching and matching include with other data that is present at that time in the system. Patterns are also thoughtful for finding the degradation or intruders in the system.

B. Clustering algorithm

Clustering is an unhandled learning process where grouping of physical or abstract objects into classes of same object takes place. A cluster is a kind of data objects which are similar to each other within the same cluster and are different to the objects in different clusters. Clustering is also known as data segmentation in definite applications as clustering partitions data sets into groups construct on their resemblance. Unlike to classification, clustering could not depend on predefined classes and class tag training examples[4].

There are lots of clustering methods accessible, and every one of them may give a dissimilar grouping of a dataset. The selection of a certain method will depend on the kind of output required. Some clustering analysis techniques are grid based clustering, density based clustering, model based clustering, partition based clustering, and hierarchical clustering.

Partitioning method generates k partitions (clusters) of the known dataset, where all partitions represent a cluster and each cluster can be depict by a centroid or a cluster representative which is some kind of synopsis explanation of all the objects available in a cluster.

(i) K-means:

K-medoids as well as K-means are the best examples of the partitioning methods. Both the k-medoids and k-means algorithms are partitioned in addition both attempts to minimize squared error. In the k-means clustering problem, the centroid is not available in the actual points in the most of cases[4][5].

In comparison to the k-means algorithm k-medoids choose data points as centers which make k-medoids more strong in the existence of noise and outliers than k-means, Hence a medoid is not as much affect by outliers or any other extreme values than a mean.

(ii) Hierarchical clustering:

Hierarchical clustering gain one after another by either splitting larger clusters, or by merging smaller clusters into larger ones .We categorize hierarchical method as being either divisive or agglomerative, based on how the decomposition take place. This agglomerative approach begins with formation of a separate group by each object. It continuously merges the groups or objects that are close to one another, until a preferred number of clusters are obtained[5].

The divisive approach, it is also called as top-down approach, starts with all objects in the same cluster. In successive iteration, a cluster is breaks up into smaller clusters, until a desired number of clusters are acquired. In hierarchical methods the step is complete, it cannot be undone sometimes this may lead to erroneous decisions. DIANA and AGNES are some of the examples hierarchical clustering.

(iii) Density based clustering:

Density based clustering techniques are based on a local cluster standard. Clusters are examined as regions in the data space where objects are dense, and are divided by regions where objects density is less. The general proposal is to grow the cluster as much as the density which is the number of objects or data points in the neighborhood overreaches certain threshold that is, for every data point inside a given cluster, the locality of a certain radius need to have at least a minimum number of points[7].

As a result these regions organized may consist of an arbitrary shape. DBSCAN is a density based method which increase clusters on the basis of a density-based connectivity analysis. OPTICS is one more density-based method which produces an augmented order of the clustering structure of data.

(iv)Grid-based clustering algorithm:

Grid-based clustering algorithm is used to divide multidimensional data space into a particular number of cells, and after that clustering operation is applied on it. The major advantage of this approach is fast processing time, which is actually self-determining of the amount of data objects along with dependent purely on the numeral of cells in every dimension of the quantized space[4][5]. STING (STatistical INformation Grid) is an example of a grid-based method based on statistical information stored in grid cells. It use to divide data space into rectangular cells, and then these cells forms a hierarchical structure and can splits the high-level cells into a number of low-level cells.

The data statistical information (for example mean , minimum, maximum, count and data distribution, etc.) of each cell is precalculated for the subsequent query processing.

(v) CLIQUE algorithm:

CLustering In QUEst(**CLIQUE**) are clustering algorithms both are density-based as well as grid-based. CLIQUE is include an algorithm based on grid and density. It divides M- dimensional data space into rectangular cells. When the quantity of data points in a cell is more than a threshold (user input), it will be called as a dense cell. A cluster is the huge collection of dense cells.

CLIQUE algorithm automatically recognizes high dimensional space along with high dense data points, and is self-determining of data input order and data distribution.

VI.PROPOSED WORK

In our proposed work we introduce the distributed partitioning algorithm, clustering based histogram, range cardinality(RC) tree to reducing the query processing time and can improve the performance by minimize both network communication cost and local files scanning . Partitioning is a process of allocating each record in a large table to a smaller table based on the value of a certain field in a record. It is used in data center networks to improve manageability and availability of big data. The partitioning step has turn in to a key determinant in data analysis to boost up the query processing performance .All these works authorize each partition to be processed independently and more efficiently.

Stratified sampling is a method of sampling from independent groups of a population, and selecting sample in each group to improve the delegate of the sample by reducing sampling error. We implementing our partitioning algorithm based on the idea of stratified sampling to create the maximum relative error under a threshold in each partition. At the same time, the sum of the local result from each partition can also achieve satisfied accuracy for any ad-hoc range-aggregate queries. We first divide the value of numerical space into different groups and subdivide each group into different partitions according to the number of available servers. The partition algorithm can be expressed as follows for data sets R:

Partitioning(R)=(g,p)=(V_e, random[1, V_r])

The stratified sampling is a method to subdivide the numerical value space into independent intervals with a batch of logarithm functions, and each interval stands for a group[7]. When the number of logarithm functions is fixed, an arbitrary natural integer N can be mapped into a unique group g.

In clustering based histogram we measure the data distributions by clustering values of all index-columns and use the learned knowledge to build our histogram. A feature vector of clustering is expressed as <tag, vector>, where tag is the attribute value, and vector is the frequency for the tag occurring in each dimension. For example, the feature {tag=ad, vector=<10,2>} indicates that the value of ad occurs in the first index column 10 times and the second index column 2 times.

RC-Tree includes three types of nodes, which are root node, internal nodes, and leaf nodes. The root node or an internal node points to its children nodes and keeps their values of spreads. A leaf node is for one bucket in the histogram. The leaf node only keeps the statistical information, and tuples values are stored in bucket data files. Because the buckets are independent of each other, the RC-Tree structure and its construction process are similar to the B+ Tree.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 6, June 2018

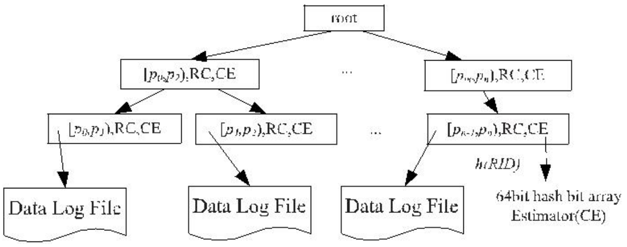


Figure4. A typical RC-Tree structure.

In order to improve throughput of RC-Tree, a hash table for newly incoming data is introduced for incremental updating process. The hash table consists of multiple nodes which are identical to the RC-Tree's leaves nodes. If a new record is coming, it first writes into the hash table, creates node if it does not exist, and then appends the tuples values into a temporary data file. When the number of nodes in the hash table reaches a threshold, the hash table flushes nodes into the RC-Tree, and appends the temporary files to the for-mal bucket data files. The incremental updating process will greatly improve the throughput of RC-Tree in big data environments.

VI.CONCLUSION AND FUTURE WORK

We propose a new approach FASTRAQ that acquires accurate estimations quickly for range-aggregate queries in big data environments. FastRAQ has O (1) time complexity for data updates. If the ratio of edge-bucket cardinality (h0) is small enough, the FastRAQ even has O (1) time complexity for range aggregate queries. FastRAQ gives the good starting spot for developing the real time answering methods for big data analysis. There are also some interesting directions for our future work. First, FastRAQ can solve the 1:n format range aggregate queries problem, i.e., there is one aggregation column and n index columns in a record. We plan to investigate how our solution can be extended to the case of m:n format problem, i.e., there are m aggregation columns and n index columns in a same record. Second, FastRAQ is now running in homogeneous environments. We will further explore how FastRAQ can be applied in heterogeneous context or even as a tool to boost the performance of data analysis in DBaas.

REFERENCES

[1]. A. Thusoo, J. S. Sarma, N. Jain, Z. Shao, P. Chakka, N. Zhang, S. Antony, H. Liu, and R. Murthy, "Hive—a petabyte scale datawarehouse using Hadoop," in Proc. IEEE 26th Int. Conf. Data Eng.,2010, pp. 996–1005.
[2]. Hadoop MapReduce distribution.Available: http:// hadoop.apache.org, 2015.
[3]. W. Liang, H. Wang, and M. E. Orlowska, "Range queries in dynamic OLAP data cubes," Data Knowl. Eng., vol. 34, no. 1,pp. 21–38, Jul. 2000.
[4].Ying Pei, Jungang Xu, Zhiwang Cen, Jian Su ,"IKMC: An Improved K-medoids Clustering Method for Near-duplicated Records Detection", 2009 IEEE conference.
[5].Mehrdad Mahdavi. Hassan Abolhassani,"Harmony K-means algorithm for document clustering", 11 December 2008 Springer Science+Business Media, LLC 2008.
[6].Prasadkumar Kale et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 6 (3), 2015, 2871-2875.
[7].S. Chaudhuri, G. Das, and U. Srivastava, "Effective use of blocklevel sampling in statistics estimation," in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2004, pp. 287–298.

DOI: 10.15680/IJIRCCE.2018.0606016