ISSN:0975-9646

An Efficient Multi-Dimensional Data Analysis over Parallel Computing Framework

Prof. Pramod Patil[1], Mr. Amit Patange[2]

[1,2]Department of Computer Engineering DYPIET Pimpri
SavitriBai Phule Pune University, India

Abstract: In the era of big data where data is growing double by it's size over year and year. So it is very difficult to handle and process the massive amount of data. Data storage and data handling should be done in real time and without loss of data. Cloud computing resolves the problem of storage and availability for data analysis task. Big data and parallel computing frameworks comes into picture where data analysis work need to be carried out. In old days data mining and data analysis have been doing by using traditional approach but after introducing various parallel computing framework it is been very easy to process and extract the data with these frameworks and technology. In this paper we details real world difficulties in data extraction, materialization, caching and data mining tasks. Specifically we introduce dremelcube, a dremel technology based framework for efficient cube computation and exploring the interesting cube groups on holistic measures. Here we demonstrate that, unlike existing techniques we can able to analyze the millions of tuples in real time for our datasets. Dremelcube efficiently and successfully computes the data by using holistic measutes over billion tuple datasets.

Keywords: Dremelcube, cloud computing, Data mining, Holistic Measures ,Data Analysis

1. INTRODUCTION

MASSIVE-SCALE analytical data mining and processing has now become an all over spread in companies and big industries. Every time solution we need with a low-cost storage that will enable vast business critical applications and data. Keeping the data at the fingertips for data analysis and engineering has been growing increasingly important. Data cube [1] analysis is a powerful tool for analyzing multidimensional [2][3] data. Parallel [14] computing framework has set a new standard benchmarks against which alternatives measure themselves, based on a proliferation of new benchmark testing. Parallel computing framework has been adopted by multiple vendors as their solution for letting customers do exploratory analysis on Big Data, natively and in place. CPU-intensive data retrieving queries might need to run on several thousands of core's and commodity hardware's to complete task within seconds. In parallel computing framework hadoop [9], GFS, MapReduce [7] programming paradigm we can use to process the data on various storage hardware's. Apache Pig [10], Hive, Cloudera [20][22] Impala [15], Amazon Redshift, Pivotal HAQW, Apache Drill, Dremel [18], Google Spanner and so on , these are the cutting edge databases used under the hadoop [10], MapReduce and other framework technology. It is also based on many improved file format and storage techniques.

The most structured nested data model for data processing need to be used while carrying out the data mining tasks. So we have different number of data processing framework, different frameworks has their own pros and cons with respect to time and other CPU-intensive attributes based on application requirement we choose the best. So we opted a dremelcube approach to analyze and extract the patterns from the data, this interactive analysis of massive amount of data that would surely resolve query problems in real-time.

```
CUBE ON timestamp(current), wp_namespace(query)
FROM [publicdata:samples.wikipedia] as (user, current,
query)
GENERATE reach(user), volume(user)
HAVING reach(user) > 5

reach(user) := COUNT(DISTINCT(user))
volume(user) := COUNT(user)
location(ip) maps ip to [Country, State, City]
timestamp(current)        maps        ip        to
[year,month,day,hour,minute,second]
wp_namespace(query) maps query to [MAIN, MEDIA,
SPECIAL, WIKIPEDIA,WIKIPEDIA_TALK]
```

Fig.1. Typical cubing task on a Wikipedia log, used to identify high-impact Wikipedia queries.

Increasingly, such large scale data are being maintained in clusters with thousands of commodity machines and analyzed using the dremelcube techniques. There are many most important analyses would be done over logs, involves calculating holistic i.e. non-algebraic measures such as find unique number of users or find top-k most frequent executed queries. As an example of such a query shows in above figure. One of the issue for data storage is that how can we efficiently distribute the data such that no single machine can be overwhelmed with amount of CPU-intensive work. In such way that we also concerned that how can we effectively distribute the computation [4] but with the parallel computing framework it is been easy to resolve all the two issues with data storage and computation. Here in this paper we have used partially algebraic measures, an important subset of holistic measures that are dremelcube friendly. Dremelcube materialization [11] and caching techniques also used internally by the approach.

A raw data is maintained as a set of tuples. Each tuple has a set of raw attributes, such as revision_id, title, ip and timestamp.

Revision_id	Contributor_ip	Title	Timestamp
298812790	60.51.121.221	Malaysian Chinese	1246046575
212321679	119.40.118.201	Universiti Putra Malaysia	1210761349
260142518	58.8.9.16	Malaysia	1230274603

Fig.2. Raw dataset, as maintained on cloud storage.

Revision_id	Country	State	City	Title	timestamp
298812790	US	CA	San Jose	Malaysian Chinese	1246046575
212321679	US	NY	NYC	Universiti Putra Malaysia	1210761349
260142518	US	Michigan	Detroit	Malaysia	1230274603

Fig.3. Derived dataset, by converting contributor_ip into geolocation and timestamp into actual data and time using classifiers. timestamp can be further classified into Y-M-D HH:MM:SS (Unix timestamp is recorded)

.

We have couple of terms need to understand with and those are Dimension attribute, cube lattice, Cube group and Cube region [6]. So we will explore one by one,

The dimension attributes is a term refers to the set of attributes that the end user wants to analyze or take into account. Based on user selected attributes, a cube lattice can be formed representing all possible grouping(s) of the attributes. cube region to denote a node in the lattice and the term cube group to denote an actual group belonging to the cube region. For example, tuples with id e1 and e2 both belong to the group <C*, *, *, media, image, *>, which belongs to the region <*, *, *, wp_namespace, type,*>. Extensive experimental analyses over real wikipedia or tested US air traffic data demonstrate that dremelcube significantly outperforms existing techniques in terms of efficiency and scalability.

2. Literature Survey

Decision [11] support systems frequently precompute many aggregates to improve response time of aggregation queries.

The production environment for analytical data management applications is rapidly changing. Many enterprises are shifting away from deploying their analytical databases on high-end proprietary machines, and moving towards cheaper, lower-end, commodity hardware, typically arranged in a shared-nothing MPP architecture, often in a virtualized environment inside public or private "clouds". At the same time, the amount of data that needs to be analyzed is exploding, requiring hundreds to thousands of machines to work in parallel to perform the analysis.

Aggregate measures [11] summarizing subsets of data are valuable in exploratory analysis and decision support, especially when dependent aggregations can be easily specified and computed. MapReduce [7] is a programming model and an associated implementation for processing and generating large data sets. Some times MapReduce is also not enough solution for data analysis. The another new technique can not replace Map-Reduce [8] totally but we can replace it for data extraction and data mining tasks. Cloudera Impala is also a powerful database feature that allows users to customize database functionality.

There is a growing need for ad-hoc analysis of extremely large data sets, especially at internet companies where innovation critically depends on being able to analyze terabytes of data collected every day. Parallel database products, e.g., Tera-bytes data, offer a solution, but are usually prohibitively expensive at this scale. Besides, many of the people who analyze this data are entrenched procedural programmers, who find the declarative, SQL style to be unnatural.

The run-time system takes care of the details of partitioning the input data, scheduling the program's execution across a set of machines, handling machine failures, and managing the required inter-machine communication. This allows programmers without any experience with parallel and distributed systems to easily utilize the resources of a large distributed system. Parallel computing runs on a large cluster of commodity machines and is highly scalable: a typical MapReduce/Impala/dremel/Apache drill computation processes many terabytes of data on thousands of machines on commodity hardware.

3. Limitations Of Existing Systems

Big data systems has mainly designed in accordance with their application requirement so system might have couple of limitation because of the created dependency on the application. In traditional techniques the data mining and data analysis had been doing using traditional approaches but those techniques are older in now a days as days grows exponentially time wise. So new for this domain a new technique with new paradigm approach we need to adopt to analyses and extract patterns from the data.

There are four main limitation in the existing techniques,

1. Large amount of data storage and processing [10].
2. Existing techniques only designed considering single machine or small number of nodes.
3. In existing techniques holistic measures [12] were not considered for data analysis means they were lacking in.
4. Detection and avoidance of extreme data skew problem [11].

4. Proposed Work

Big data [10] describes a holistic information management strategy that includes and integrates many new types of data and data management alongside traditional data.

Big data has also been defined by the four Vs:

Volume : The amount of data. While volume indicates more data, it is the granular nature of the data that is unique.

Velocity : The fast rate at which data is received and perhaps acted upon. The highest velocity data normally streams directly into memory versus being written to disk.

Variety : New unstructured data types. Unstructured and semi-structured data types, such as text, audio, and video require additional processing to both derive meaning and the supporting metadata.

Value : Data has intrinsic value-but it must be discovered. There are a range of quantitative and investigative techniques to derive value from data-from discovering a consumer preference or sentiment, to making a relevant over by location, or for identifying a piece of equipment that is about to fail.

As per defined and used in the existing system the Hadoop Distributed File System (HDFS) [9][22] is designed to store very large data sets reliably, and to stream those data sets at high bandwidth to user applications. In a large cluster, thousands of servers both host directly attached storage and execute user application tasks. By distributing storage and computation across many servers, the resource can grow with demand while remaining economical at every size. Hadoop is derived from GFS (Google File System). Apache Hadoop is a batch oriented solution that has a lack of support for ad-hoc, real-time queries. Many of the players in Big Data have realized the need for fast, interactive queries besides the traditional Hadoop approach. Columnar storage, one the key solution vendors in Big Data/Hadoop domain has just recently launched dremel that addresses this gap. Dremel [18] is inspired from the parallel databases concept [07]. Dremel provides a SQL-like query language for wide variety of SELECT statements with WHERE, GROUP BY, HAVING clauses, with ORDER BY – though currently LIMIT is mandatory with ORDER BY external tables, etc. It also supports arithmetic and logical operators and it's built-in functions such as COUNT, SUM, LIKE, IN or BETWEEN. It can access data stored on situ but it does not use MapReduce, instead it is based on its own distributed query engine.

Cloud computing platform [18] offers hosting on the same supporting infrastructure. Storage is reliable with protection of data failure. Anytime time anywhere we can access the data and cloud programs over the network. Large data set storage and their availability for application makes it easy process for data mining and data analysis. With cloud computing environment we can build complex web application for business purposes. There are many cloud service api's available by Google, Amazon etc for academic purpose.

We propose the dremelcube approach that addresses the challenges of massive data set scale cube computation with holistic measures. By using dremelcube approach we have reduced intermediate data size and computational cost. Here we are dealing with data partitioning and cube lattice partitioning.

Dremelcube is not a complete replacement for MapReduce paradigm but it is often used in conjunction with it to the analyze outputs rapidly prototype larger computation. In this approach various variety of commodity hardware's has been used for processing and computation of data so we are defining server node architecture for dremelcube approach,

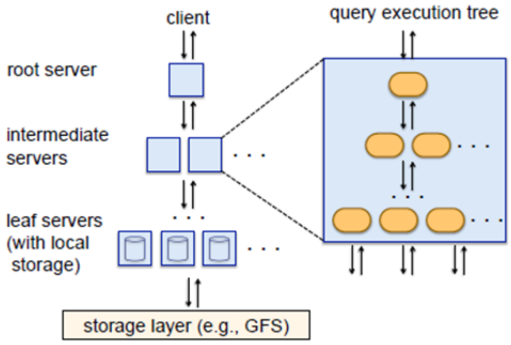

Fig.4 System execution inside the server node.

We are briefing (in next figure) below the system architecture and work flow of the system, dremelcube approach interacts with dremel technology developed by Google inc. and also it is dealing with cloud storage for billions of tuples from the data set. This is approach is based on the columnar data storage representation structure which gives it benefit in data retrieval and executing real-time queries. Dremelcube approach considers partially algebraic measures, result materialization and data mining techniques [11].

We begin with partially algebraic technique that first we identify holistic measures that can be easy to compute in parallel. We have an example like to find number of unique users from the dataset, so approach can find or store pre-calculated value as total number distinct users.

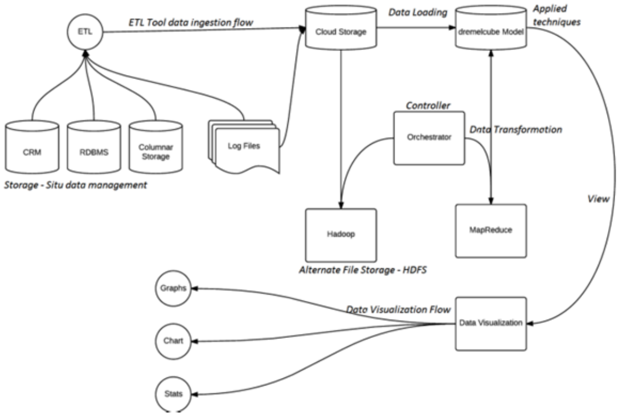

Fig.5 System Architecture - dremelcube.

We are briefing (in next figure) below the system architecture and work flow of the system, dremelcube approach interacts with dremel technology [18] developed by Google inc. and also it is dealing with cloud storage for billions of tuples from the data set. This is approach is based on the columnar data storage representation structure which gives it benefit in data retrieval and executing real-time queries. Dremelcube approach considers partially algebraic measures, result materialization and data mining techniques.

We begin with partially algebraic technique that first we identify holistic measures that can be easy to compute in parallel. We have an example like to find number of unique users from the dataset, so approach can find or store pre-calculated value as total number distinct users.

Another technique includes in dremelcube approach is value partitioning and forming a batch areas. Partitioning a large group based on algebraic group called value partitioning. There are many mapper function when we got large data set then we have to work on multiple reducer to get the result but in the reducers there many reducers which unfriendly So we need to know the partition factor so we can reduce reducer un-friendly group from computation. To know the partition factor we need detect reducer un-friendly groups on the fly or another approach is scan all the data and compile it. So we need to adopt sampling approach, We estimate the reducer-unfriendliness of each cube region based on number of groups it is estimated to have and perform partitioning for all groups within the list of cube regions that are estimated to be reducer-unfriendly. We declare group G to be reducer-unfriendly if we observe more than 0.75rN tuples of G in the sample.

N = Sample size

r = c/|D| denotes max no. of tuples a single reducer can handle.

c = reducer limit

|D| = total no. of tuples in the data.

Batch areas [11], we group cube regions into areas such that, A region that is reducer friendly must belong to a batch area that contains at least one of it's parent.

No two region whose parents are reducer unfriendly can belong to same area. The maximum difference in number of regions between any pairs of areas is 2, a heuristic to balance the load. So in this way batch areas are formed.

Algorithm 1

```
dremelcube(Cube Lattice C, Dataset D, Measure M)
    Dsample = SAMPLE(D)
    RegionSizes R = ESTIMATE - MapReduce(Dsample, C)
    Ca = ANNOTATE(R,C) #value part. & batching
    while(D)
    do R <- R UNION dremelcube-MapReduce(Ca, M, D)
        D <- D' # retry failed groups D' from dremelcube-Reduce
        Ca <- INCREASE-PARTITIONING(Ca)
    Result <- MERGE(R) #post-aggregate value partitions
    return Result
```

Result or cube materialization are done over the result set and result is being stored on the physical hard drive to fetch the data next time if same request is raised by the end users. After done all techniques data cube mining is done at the end to get desired result also we can find out the patterns from the data by using apache S4 i.e. simple scalable streaming system by giving result input to s4.

5. Experimental Setup

In this section we evaluate Dremelcube performance on several datasets used in the experiments and examine the effectiveness of this approach for nested data. We adopted three data sets. The Real data set contains real life click streams obtained from the logs of Wikipedia. We have also examined the United states Air-traffic data set. And third dataset is also under observation taken from Github commits and log details. The main dataset we are using for all data analysis task is Wikipedia search query logs and those are classified as follows, timestamp, contributor_ip. We establish two dimension containing total six levels. The contributor_ip dimension contains three derived leves (from lowest to highest, with cardinalities) : city-> state-> country and is derived from contributor_ip.

This data set in full contains 314M click tuples for a size of 36 GB. The number of unique users and queries are in the range of eight to tens of millions. We can also generate synthetic example data set. We have run some of the base techniques on different parallel computing framework distributed databases and here is the below results analysis done with the help of graphs as follows,

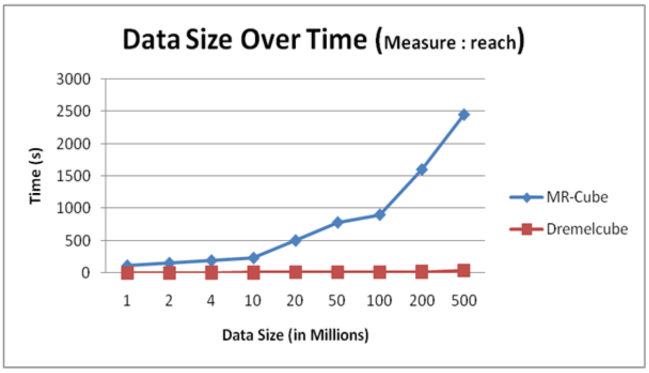

Fig.6 Data set running over time (Measure : reach).

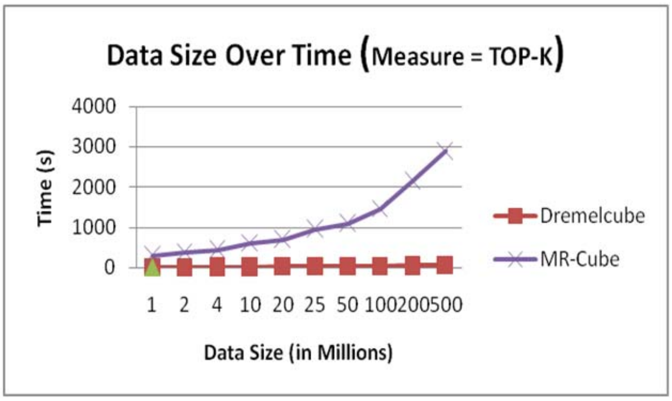

Fig.7 Data set running over time (Measure : TOP-K)

These above graphs are based on the prepared model queries and executed using existing systems resources and technology and it is compared with dremelcube approach. As we have stated dremelcube uses columnar data storage and situ data management algorithms so it drastically reduced time as compared to used hadoop's native databases.

6. CONCLUSION

In this paper, we study data analysis using various data exploration techniques and subsequent mining of holistic measures over extremely massive amount of data such as based on dremel technology in distributed data storage. We identify a subset of holistic measures that partially algebraic. We design on technique in that way we can easily analyze the data and their patterns. We design algorithms that partition the cube lattice into batch areas. Then after that cube materialization is to be carried out. And finally we find interesting cube groups as a part of data cube computation. We have also minimized the data skewness by using dremelcube approach.

ACKNOWLEDGEMENT

I would like to take this opportunity to acknowledge the contribution of certain people without which it would not have been possible to complete this paper work. I would like to express my special appreciation and thanks to my advisor Professor Pramod Patil, you have been a overwhelming mentor for me.

REFERENCES

1) Bosworth, Gray, Chaudhuri, Reichart, Pellow, Venkatrao, "Data cube: a relational operator generalizing group by, cross-tab and sub-totals," Proc. 12th Int’l Conf. Data Eng. (ICDE), 1996.
2) Deshpande, S. and R. Agarwal, Gupta, Naughton J., Sarawagi ,Ramakrishnan, "On the computation of multidimensional aggregates," Proc.22nd Int’l Conf. Very Large Data Bases (VLDB), 1996.
3) M. Deshpande , F. Naughton, Zhao.”An array based algorithm for simultaneous multidimensional aggregates”. In SIGMOD'97.
4) Srivastava D., Ross, "Fast computation of sparse data cubes," Proc. 23rd Int'l Conf. Very Large Data Bases (VLDB),1997.
5) Han , Xin D., X. Li, and W. B. Wah Starcubing: Computing iceberg cubes by top-down and bottom-up integration. In VLDB'03.
6) S.Wagner, R.T. Ng and Yin Y., "Iceberg-cube computation with PC clusters," Proc. ACM SIGMOD Int’l Conf. Management of Data, 2001.
7) Yury K. and Sergey K., “Applying map-reduce paradigm for parallel closed cube computation,” Proc. First Int’l Conf. Advances in Databases, Knowledge, and Data Applications (DBKDA), 2009.
8) S. Ghemawat, Dean J.. MapReduce: Simplified Data Processing on Large Clusters. OSDI, 2004.
9) Abouzeid A. et al., “HadoopDB: an architectural hybrid of mapreduce and DBMS technologies for analytical workloads,” Proc. VLDB Endowment, vol. 2, pp. 922-933, 2009.
10) Murthy A.C. and Shvachko K.V., “Scaling hadoop to 4000 nodes at Yahoo!,” Yahoo! Developer Network Blog, 2008.
11) Yu Cong, Arnab Nandi, Bohannon Philip, and Ramakrishnan Raghu “Data cube materialization and mining over map-reduce “ IEEE transaction on Knowledge and Data Engineering, vol. 24, no. 10, Oct 2012.
12) C. Yu, Bohannon P., Nandi A. and Rama krishnan R., “Distributed cube materialization on holistic measures,” Proc. IEEE 27th Int’l Conf. Data Eng. (ICDE), 2011.
13) Chen Y., Dehne F. K. H. A., Eavis T., and Rau Chaplin A.. PnP: sequential, external memory, and parallel iceberg cube computation. Distributed and Parallel Databases, 2008.
14) Sergey K. and Yury K.. Applying Map Reduce Paradigm for Parallel Closed Cube Computation. DBKDA, 2009.
15) http://www.cloudera.com/content/cloudera/en/documentation/core/latest/topics/introduction.html
16) http://theprofessionalspoint.blogspot.in/2012/12/google-dremel-vs-apache-hadoop-big-data.html
17) http://vision.cloudera.com/impala-v-hive/
18) http://static.googleusercontent.com/media/research.google.com/en//pubs/archive/36632.pdf
19) https://wishkane.wordpress.com/2013/07/06/cloudera-impala-fast-interactive-queries-with-hadoop/
20) http://blog.cloudera.com/blog/2012/10/cloudera-impala-real-time-queries-in-apache-hadoop-for-real/
21) http://www.tutorialspoint.com/hbase/hbase_overview.htm
22) https://wishkane.wordpress.com/2013/07/06/cloudera-impala-fast-interactive-queries-with-hadoop/
23) http://research.google.com/archive/bigtable.html

AUTHORS

[1]Pramod Patil, Research Scholar, computer department, College of engineering, Pune
[2]Amit Patange, PG student, Computer department, DYPIET, Pune

Prof. Pramod D.Patil obtained his Bachelor’s degree in Computer Science and Engineering from Swami Ramanand Tirth Marathwada University , India. Then he obtained his Master’s degree in Computer Engineering and pursuing PhD in Computer Engineering majoring in Mining Data Streams both from Pune University, INDIA. Currently, he is a Research Scholar in Department of Computer Engineering at COEP, Pune University, INDIA. His specializations include Database Management System, Data Mining, and Web Mining. His current research interests are Mining Data Streams.

Mr. Amit Patange obtained his Bachelor’s degree in Computer Science from University of pune, India. Now pursuing Master’s degree in Computer Engineering University of pune, INDIA. His dissertations work on Data Mining.