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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2014

Data Cube Materialization Using Map Reduce

Kawhale Rohitkumar[1], Sarita Patil[2]

Student, Dept. of Computer Engineering, G.H Raisoni College of Engineering and Management, Pune, SavitribaiPhule Pune University, India[1]

Assistant Professor, Dept. of Computer Engineering, G.H Raisoni College of Engineering and Management, Pune, SavitribaiPhule Pune University, India[2]

ABSTRACT: Data cube queries act an important class of On-Line Analytical Processing (OLAP) queries in decision support systems. To meet this needof improved performance and to handle the increase in data sizes, parallel solutions effectively we have generating the data cube. We present load balanced and communication efficient partitioning strategies which generate a sub cube computation for every processor. Sub cube computations are then carried out using existing sequential, external memory data cube algorithms. In parallel systemsbalancing the load assigned to different processors and minimizing the communication overhead are the core problems for achieving high performance. In this paper we propose a three-phase cube computation algorithm MR-Cube that employs these techniques to successfully cube billion-tuple sized data sets, and optionally surfaces interesting cube groups. In this paper, we details real-world challenges in cube materialization and mining tasks on some type of data sets. Unusually, we identify an important subset of holistic measures i.e. non-algebraic measures and bring out the MR-Cube, i.e.MapReduce-based framework for efficient cube computation and identification of interesting cube groups on holistic measures.

KEYWORDS: Big Data;Data Cube; MapReduce; Value Partitioning

I. INTRODUCTION

Data cube is acrucial concept and research direction in OLAP (Online Analytical Processing). In former years, the researches on the data cube are mainly in two aspects: firstly, how to compress the cube and store cube. Becoming to the massive data, storing all the data cubes needs a lot of space and resources. Secondly, how to choose the cube and materialize them. In order to help customers to get effective data, many learners pay attention on finding the best method of how to prefer and materialize data cubes.

Researchers have suggested a variety of algorithms on Cube compression and Storage, such as quotient cube, star cube,iceberg cube and so on. In Iceberg cube computation, the cubes which do not satisfy the requirements of pre-defined constraints will be deleted and not stored. By this suggests, the initial cubes are compressed. Quotient cube relies on equivalence partitioning. It keeps the linguistics of the initial cubes, so it will get the results of the question on any cube simply. At constant time, some learners have projected the construct of the iceberg quotient cube. This cube could be a combination of quotient cube and iceberg cube.

In this age of data explosion, parallel processing is crucial to processing a large volume of data in a timely manner. The Data cube analysis is a powerful tool for studying multidimensional data. Users typically consider the data as multidimensional data cubes.Data cube construction is a normally used operation in data warehouses. Each cell of the data cube is a view consisting of an aggregation of interest. The values of many of these cells are dependent on the values of other cells in the data cube. A familiar and occupied interrogate optimization way is to be included various or on all sides of these cells quite than compute them from raw data each time.Commercial systems differ mainly in their approach to materializing the data cube [4]. Cube analysis provides the users with a one of the best way to discover insights from the data by computing different aggregate measures. Data analysis applications typically aggregate data across many dimensions looking for strange patterns. It is the process of extraction of useful patterns from the large database.Therefore it employs methods at the cross point of statistics,machine learning and

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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2014

database systems.Data mining is the core process or the heart of knowledge discovery in database. Data mining is the application of efficient algorithms to detect the desired patterns contained within the given data.

II. RELATED WORK

As the introduction of data cube, and there has been many techniques [5], [6], [8], [9], [10] have been proposed for efficient cube computation. Data cube computation as well as materialization are one of the most essential but expensive procedures in data warehousing [1] [3]. Previous studies have developed two major approaches, top-down versus bottom-up. The former, represented by the MultiWay Array Cube algorithm, aggregates simultaneously on multiple dimensions; however, it cannot take advantage of a priori pruning when computing iceberg cubes [5].

Our work is the first comprehensive study on cube materialization for holistic measures using the MapReduce paradigm [3]. In addition to describing real world challenges associated with holistic cube computation using MapReduce for Web-scale datasets, they had some contributions that are the two-phase cube materialization algorithm MR-Cube that employs these techniques to successfully cube billion-tuple sized datasets [3].The cube computation task can be in-corporate as an operator into those languages that may be pig or hive to provide users with a friendly way to explore their data without issuing many ad-hoc aggregate queries [10].

III. BIG DATA MINING CHALLENGES

For an voguish or intelligent learning database system to handle Big Data, the required key is to scale up to the exceptionally large massive volume of data and provide treatments to that Big Data processing framework, which includes layers from inside out with considerations on data accessing and computing (Layer I), data privacy and domain knowledge (Layer II). The challenges at Layer I focus on data accessing and arithmetic computing procedures. Such that Big Data are often stored at many different locations and hugedata volumes may endlessly grow, an effective computing platform will have to take distributed large-scale data storage into some circumstance for computing. For example, distinctive data mining algorithms require all data to be loaded into the main memory, so this is becoming a straighten out technical barrier for Big Data because moving data across different locations is expensive even if we do have a super expectant main memory to hold all data for computing.

A. Layer I: Big Data Mining Platform

In typical systems, the mining procedures require computational intensive computing units for data analysis and comparisons. A computing platform in this system is, needed to have efficient approach to, at least, two types of resources: the data and computing processors. For small scale mining tasks, a single desktop computer, which contains CPU processors and hard disk, is sufficient. Layer 1 focuses on low-level data accessing and computing.

For Big Data mining, because data scale are huge such that to manage such massive data by commodity hardware and also the capacity of this single personal computer (PC) can handle, a typical Big Data processing framework will rely on cluster computers with a high-performance computing platform, with a mining task being deployed by running some parallel programming tools, such as MapReduce on a large number of computing nodes (i.e., clusters) [4]. The role of the software component is to make sure that a single mining task, such as finding the appropriate match of a query from a database with billions or trillions of records, is split into many small tasks each of which is running on one or multiple computing nodes.

B. Layer II: Big Data Semantics and Application Knowledge

This layer refers to numerous aspects related to the regulations, policies, domain information, and user knowledge. The two most important issues at this layer includes

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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2014

i. *Data sharing and privacy*

Information sharing is a motivation for sharing, a real-world concern is that Big Data applications which are related to sensitive information, such as medical records andbanking transactions. To protect privacy, two common approaches are to 1) restrict access to the data, for adding certification or the access control to data entries, so classified information is accessible by a limited group of users only, and 2) unknown data fields such that sensitive information cannot be pinpointed to an individual record. For the first approach, common challenges are to design secured certification or access control mechanisms, such that no conscious information can be misconducted by unauthorized individuals. The main objective is to inject randomness into the data to ensure a number of privacy goals.

ii. Domain and application knowledge.

Domain and application knowledge provides suitable information for designing Big Data mining algorithms and systems. In a primary case, domain knowledge can help to identify right features for modelling the underlying data. The domain and application knowledge can also help design and fulfil business objectives by using Big Data analytical techniques. Without having correct domain knowledge, it is a big challenge to find accurate measures to characterize the market movement, and such knowledge is often outside the mind of the data miners, although some current research has shown that using social networks.

IV. PROPOSED SYSTEM

For the groups with a large number of tuples from the cube lattice, the memory requirement for maintaining such intermediate data can become overwhelming. We come up to this challenge by identifying an important subset of holistic measures i.e. non-algebraic measures, partially algebraic measures, and introducing a value partition mechanism such that the data load on each machine can be controlled. We design and implement sampling algorithms to efficiently detect groups of cubes where such value partition is required [1] [6]. The second issue is how to effectively distribute the computation such that we impact a good balance between the amount of intermediate data being produced and the pruning of unnecessary data. We design and implement algorithms to partition the cube lattice into batch areas and effectively distribute the materialization and mining across available machines [2]. In conspiratorto describing real-world challenges linked with holistic cube computation using MapReduce for web scale data sets, we make the following main contributions:

- We formally introduce partially algebraic measures, an important subset of holistic measures that are MapReduce friendly [1].
- We intend two techniques, value partitioning and batch area identification that effectively leverages the MapReduce framework to distribute the data and computation workload [1] [2].
- We propose a three-phase cube computation algorithm MR-Cube that employs these techniques to successfully cube billion-tuple sized data sets, and optionally surfaces interesting cube groups [4].
- We are looking into the use of compressed counting data structures such as CM-Sketch as a solution to extreme data skew, which occurs if a few cube groups are unusually large [5].

The below figure is of the proposed system architecture of paper consisting of admin level, the client, MapReduce phase with aggregate function and giving output as analysis of large data skew as line chart, pie chart. The admin level manages the MR cube approach with value partitioning of data. The API manger manages the visualization of user interface to client whereas the DB manager manages the database output from map reduce phase that is cube formation from dataset submitted from the client. We have mangers for every level defining their roles. The dataset needed for the cubing is provided by the client side so that it can any type of large data set for the data cube formation. The application peripheral interface is providing the GUI to the client such that the mapping and reducing of data can be calculated. As the analysis of such large data set will be given by the graphs, tables, charts. Such that the client will receive report of it.

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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2014

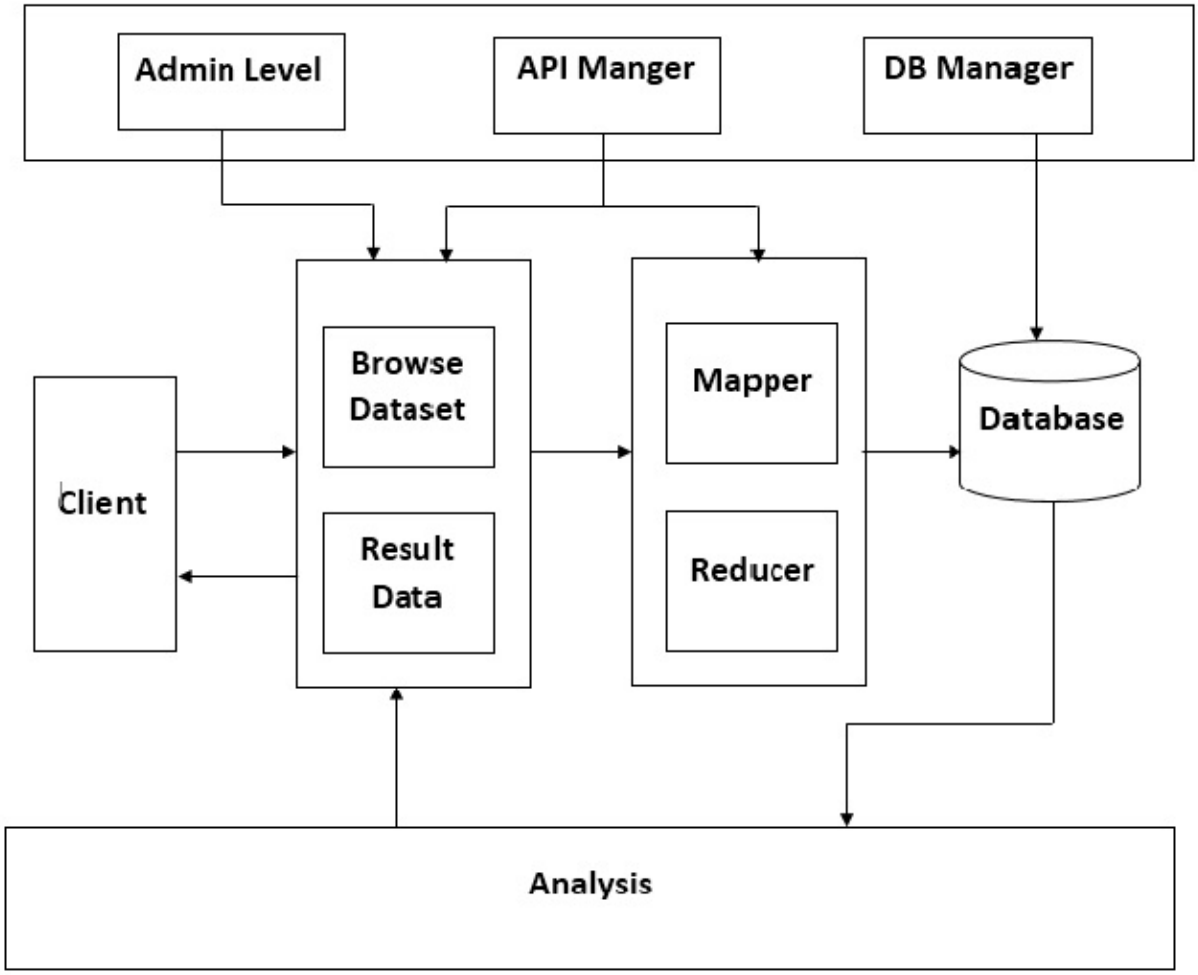

Figure: Proposed System Architecture

- *Aggregate Function:*

A function that performs computation on a *set* of values rather than on a single value. Aggregate measures summarizing subsets of data are valuable in exploratory analysis and decision support, particularly when dependent aggregations can be easily specified and computed.

Algebraic aggregates are tougher to compute than distributive aggregates. An algebraic aggregate saves its computation in a handle and produces a result in the end. The super-aggregate needs these intermediate results rather than just the raw sub-aggregate. An algebraic aggregate must assert a handle tuple for each element of the cube.In case for example, we can use a weblog analysis problem to illustrate the concept of queries with correlated aggregates. In this task, the goal is to understand user's behaviour on the search results and the advertisements by analysing massive web search session logs.

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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2014

- *Map Reduce Phase:*

Data partitioning, fault tolerance, execution scheduling are provided by MapReduce framework itself. MapReduce was designed to handle large data volumes and huge clusters (thousands of servers). MapReduce is a programming framework that allows to execute user code in a large cluster. All the user has to write two functions: Map and Reduce.

During the Map phase, the input data are distributed across the mapper machines, where each machine then processes a subset of the data in parallel and produces one or more <key; value> pairs for each data record. Next, during the Shuffle phase, those <key, value> pairs are repartitioned (and sorted within each partition) so that values corresponding to the same key are grouped together into values {v1; v2; :::}. Finally, during the Reduce phase, each reducer machine processes a subset of the <key, {v1; v2; :::}> pairs in parallel and writes the final results to the distributed file system. The map and reduce tasks are defined by the user while the shuffle is accomplished by the system.

Even though the former pseudo code is written in terms of string inputs and outputs, conceptually the map and reduce functions supplied by the user have associated types.

Map (k1, v1) → list (k2, v2)
Reduce (k2, list (v2)) → list (v2)

Two programmer specified functions:
✓ Map
Input: key/value pairs ! (k1,v1)
Output: intermediate key/value pairs !list(k2,v2)

✓ Reduce
Input: intermediate key/value pairs ! (k2,list(v2))
Output: List of values !list(v2)

That is, the input keys and values are drawn from a different domain than the output keys and values. The k1, k2 are the two different keys used in MapReduce phase and same as v1, v2 are the different values. The intermediate keys and values are from the same domain as the output keys and values.

V. MR CUBE APPROACH

We propose the MR-Cube approach that addresses the challenges of large scale cube computation with holistic measures i.e. non algebraic measures. Our goal is to divide the computation into pieces such that no reducer has to deal with extremely massive data groups, and the total intermediate data size is controlled.

A. Partially Algebraic Measures

We begin by identifying a subset of holistic i.e. non algebraic measures that are easy to compute in parallel than an arbitrary holistic or non-algebraic measure. We call them partially algebraic measures. For example, to compute the measure reach (i.e., unique number of users) of billions of search tuples, a known practical approach is to first group the tuples by the user id (uid) and then count the number of such groups produced. It is as if the holistic measure (non-algebraic measures) has become algebraic for the attribute user id.

B. Value Partitioning

We want to perform value partitioning only on some groups that are likely to be reducer-unfriendly and dynamically adjust the partition factor. This approach is to detect reducer unfriendly groups on the fly and perform partitioning once they are detected. This is to scan the cube data and compile a list of potentially reducer-unfriendly groups, for which the mapper will perform partitioning in MR cube approach. Finally, extreme data skew can occur in some sort of datasets, which will cause value partitioning to be applied to most of the cube regions [1].

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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2014

Value Partitioning is used for distribute data for that we are running Naïve Algorithm. We want to perform value partitioning only on groups that are probably to be reducer unfriendly and dynamically adjust the partition factor. We assume a sampling approach where we figure out the reducer unfriendliness of each cube region based on the number of groups it is calculated to have, and perform partitioning for all groups within the list of cube regions (a small list) that are estimated to be reducer unfriendly.

C. Batch Areas

Partitioning technique is used for effectively distribute computation. It is also called as Batch Area. Where each batch area represents a collection of regions that share a common ascendant region. We suggest to combine regions into batch areas. Mappers can nowlet out one key-value pair per batch for each data tuple, thus drastically reducing the amount of intermediate data [2]. Reducers, on the other hand, instead of simply applying the measure function, execute a traditional cube computation algorithm over the set of tuples using the batch area as the local cube lattice.

Intuitively, the three constraints state that: (i) a region with at least one parent that is also reducer friendly must belong to a batch area that contains at least one of its parents; (ii) no two regions whose parents are reducer-unfriendly can belong to the same batch area; (iii) the difference in the number of regions of two batch areas cannot be more than two, a heuristic used to balance the workload of each batch area [8]. Since each batch area will effectively require an independent projection of the dataset, they directly impact the size of intermediate data, and hence overall performance. Thus, it is important to construct batch areas that minimize the amount of intermediate data generated. The combined process of identifying and value-partitioning unfriendly regions followed by the partitioning of friendly regions is referred to as annotate [2]. An annotated lattice is generated and then used to perform the MR cube MapReduce. The value partitioned groups are merged to produce the final result for that group.

VI. CONCLUSION

Efficient Cube computation is important problem in data cube technology. There are so many techniques used for computing the cubessuch as Multiway array aggregation, Bottom up Computation, Star Cubing, the computation of shell fragments and parallel algorithms. Proposed approach effectively distributes data and computation workload .Using important subset of holistic measures we are doing cube materialization and identifying interesting cube groups. MR-Cube algorithm efficiently distributes the computation workload across the machines and is able to complete cubing tasks at a scale where previous algorithms fails.Extreme Data skew is most important challenge which occurs if a few cube groups are unusually large even when they belong to a cube region at the top of the lattice (i.e., those with fine granularities).This causes value partitioning to be applied to the entire cube and therefore reduces the efficiency of algorithm. We are looking into the use of compressed counting data structures such as CM-Sketch and Log-Frequency Sketch as a solution.After the extreme data skew detection and try to avoid same, we are using MR-Cube algorithm for cube materialization and identifying interesting cube groups.

REFERENCES

1. Arnab Nandi, Cong Yu, Philip Bohannon, and Raghu Ramakrishnan," Data Cube Materialization and Mining over MapReduce", IEEE transactions on knowledge and data engineering, vol. 24, no. 10, October 2012
2. Amrutha S, Viji Mohan A"Large-Scale Data Mining And Materialization UsingCm-Sketch And Mr Cube Algorithm"Iraj International Conference, 13th October 2013, Trivandrum, India. Isbn: 978-93-82702-33-7
3. A.Nandi, C. Yu, P. Bohannon, and R. Ramakrishnan, "Distributed Cube Materialization on Holistic Measures," Proc. IEEE 27th Int'l Conf. Data Eng. (ICDE), 2011.
4. Xindong Wu, Xingquan Zhu, Gong-Qing Wu, and Wei Ding, Senior Member, IEEE," Data Mining with Big Data" IEEE transactions on knowledge and data engineering, vol. 26, no. 1, January 2014
5. Dong Xin, Jiawei Han, Xiaolei Li, Zheng Shao, and Benjamin W. Wah, Fellow, IEEE, "Computing Iceberg Cubes by Top-Down and Bottom-Up Integration:TheStarCubing Approach "IEEE transactions on knowledge and data engineering, vol. 19, no. 1, January 2007
6. Abouzeid et al., "HadoopDB: An Architectural Hybrid of MapReduce and DBMS Technologies for Analytical Workloads," Proc. VLDB Endowment, vol. 2, pp. 922-933, 2009.
7. Amrutha S, Viji Mohan A"Large-Scale Data Mining And Materialization Using Cm-Sketch AndMr Cube Algorithm"Proceedings of 2nd IRAJ International Conference, 13th October 2013, Trivandrum, India. ISBN: 978-93-82702-33-7.
8. Xin et al., "Star-Cubing: Computing Iceberg Cubes by Top-Down And Bottom-Up Integration," Proc. 29th Int'l Conf. Very Large Data Bases (VLDB), 2003.
9. K. Beyer and R. Ramakrishnan, "Bottom-Up Computation of Sparse and Iceberg CUBEs," Proc. ACM SIGMOD Int'l Conf. Management of Data, 1999.
10. Talbot, "Succinct Approximate Counting of Skewed Data," Proc. 21st Int'l Joint Conf. Artificial Intelligence (IJCAI), 2009.