Map-Reduce for Cube Computation

Prof. Pramod Patil1, Prini Kotian2, Aishwarya Gaonkar3, Sachin Wani4, Pramod Gaikwad5

Department of Computer Science, Dr. D.Y. Patil Institute of Engineering and Technology Pimpri, Pune-411018

Abstract
Analyzing of large data sets is a major concern. Big data contains large amount of unstructured data having heterogeneous patterns. It is quite difficult for existing techniques to give better performance while processing such large set of data. On the other hand tremendously changing size of the data and design parameters for the same becomes an inimitable and interesting challenge. This paper deals with the real world challenges of cube computation and materialization over interesting measures. Cube designing is efficient for handling of unstructured data. There are various techniques for cube computation such as annotation, aggregation, materialization and mining. So Map Reduce approach is provided for efficient computation of cube. Hadoop is an open source software framework for storing and processing big data in distributed form. Hive is an infrastructure on the top of Hadoop for storing query and analysis of large data sets. MR-Cube is a framework of Map Reduce for computation of online analytical processing. Thus MR-cube successfully handles cube computation with dynamics measures over large data sets.

Keywords: Data cube, Cube Materialization, Cube Mining, Map-reduced, MR-Cube, Dynamic Measures.

I. INTRODUCTION

In the past few decades, Organization have tried different approaches to solve the problem of handling Big Data that requires lot of storage and large computation that demands a great deal of processing power. Thus Hadoop was adopted as a platform to provide distributed storage and computational capabilities. Merits of using Hadoop are scalability and availability along with the distributed environment which are provided by Hadoop Distributed File System (HDFS) and computational capabilities provided by Map Reduce. HDFS helps in replication of files when software and hardware failure occurs, and automatically re-replicates data blocks by providing security to Big Data.

Map-Reduce was introduced by Google in 2004. It is based on Divide and Conquer principles. Map-Reduce is the main processing engine of Hadoop. The Map Reduce model simplifies parallel processing by abstracting away the complexities involved in working with distributed systems, such as parallelization, distribution of work while dealing with software and hardware failure. With this abstraction, Map Reduce allows the programmer to focus on addressing business needs.

Data cube is a way of organizing data in N-dimensions so as to perform analysis over some measure of interest. Data-Cube is an easy way to look at complex data in simple format. Challenges of Data-cube computation are Size, Complexity, Design, and Quality.

The paper is organized as follows: Firstly explanation of different approaches of Cube Designing. Next is the Limitations of these techniques over handling of Big Data sets. Then the implementation steps of Map Reduce based Approach used for data cube materialization and mining. And the last part of paper is Conclusion.

II. DIFFERENT TECHNIQUES FOR CUBE COMPUTATION

Here we generate various related methods that are used for the computation of the cube and its performance scope and its merits and demerits.

In paper [1], introduces analyzing and optimization technique for cube computations such as effectively distribute the data. They also focus that no single machine is overwhelmed with small number of nodes. The CUBE needs computing group bys on all possible combinations of list of attributes and is equivalent to the union of standard group by operations. The proposed algorithm is only works for algebraic measure i.e. such as SUM, COUNT, and AVERAGE etc.

In paper [2], they introduce top-down approach for cube computation called multi way array aggregation. The computation begins with the grouping of queries as larger group-byss and proceeds towards the smallest group-byss. Here the planes should be computed and interesting groups are sorted. Limitation of this method is computing takes well only for a small number of dimension.

In paper [3], Bottom-up Cubing Algorithm (BUC) method is used in which First, BUC if a group has value partitioned then algorithm executed on a single reducer is self-contained. Let us consider partitions dataset on dimension A, producing partitions a1, a2, a3, a4. Then, it recourses on partition a1, the partition a1 is aggregated and BUC produces <a1,*,*,*>. Computing cube form cuboid to base cuboid.
In paper [11][12][4], Parallel Algorithms are introduced for cube computation over clusters. In these Algorithm data, dimensions and measures are given as input. Parallelized aggregation of data subsets whose results are then post-processed to derive the final result.

BPP (Breadth-first Partitioned Parallel Cube), a parallel algorithm designed for cube materialization over flat dimension hierarchies. Another Parallel algorithm PT (Partitioned Tree) works with tasks that are created by a recursive binary division in each lattice on a single machine into two sub trees having an equal number of nodes. In PT, there is a parameter parallelism (number of reducers) that controls when binary division stops.

In paper [14] two more algorithms is described. RP (Replication Parallel BUC) and ASL (Affinity SkipList). Algorithm Rp is dominated by PT. In Algorithm ASDL each cube region in parallel is used to maintain intermediate results during the process.

In paper[16], For fast online multi-dimensional analysis of stream data, three important methods are proposed for efficient and effective computation of stream cubes. Based on this design methodology, real life cubing can be constructed.

Introduce MR-Cube [7], a Map-Reduce based framework for efficient cube computation and identify interesting cube groups on holistic measure. Cube region is grouping of attribute while a cube group is values of those attribute. Cube lattice is formed by representing all possible groupings of the attributes. Challenging issue is that effectively distribute the computation in terms of efficiency and scalability.

III. LIMITATION OF EXISTING TECHNIQUES

Limitations in the existing techniques:
1. The existing techniques are designed to handle clusters of small number of nodes or for single machine processing. Thus it is difficult to manage processing of large amount of data.
2. The previous techniques deal with algebraic measure and the data is growing large day by day which needs holistic measures but distribution using holistic m challenge.

There are several more challenges arising when dealing with data cubes over large amount of data.
1. Size: The size of data over large data sets and also the size of intermediate data generated after mapping phase become a great challenge as it can lead to disk running out of space when naive algorithm is used.
2. Complexity: Complexity of cube building becomes a challenge when dealing with large amount of unstructured and real time data where measures an dimensions change all the time.

3. Design: Designing methods of data cubes have been becoming interesting and challenging. The parameters to be considered are construction time, cube updation techniques, maintenance plan and the design techniques to be adopted.

4. Quality: Quality becomes a complex factor when data is huge and as data cube is formed the quality of cube tends to be affected during aggregation phase. Thus it is important to control the quality of final cube.

IV. MAP-REDUCE BASED APPROACH FOR CUBE COMPUTATION OVER BIG DATA

Map-Reduce is a programming model and an associated implementation for popular parallel execution frameworks. A proposed methodology is used also to handle two major issues such as data distribution and computation distribution by illustrating a framework to partition high multidimensional lattice into region areas and distribution of data analysis and mining under parallel computing infrastructure. The research contribution is as follows: Partitioning high multidimensional lattice into region areas, Three phase high multidimensional data computation algorithm to handle billions of data streams, Fusion of stream mining model with multidimensional data streams.

Fig 1 - Flow Chart for cube computation by using map-reduce approach
The given Map-reduced based system is designed with flow diagram as shown in figure 1. It consists of following steps:

(i) Data Sample i.e. data set, which pre-process the data, dimension hierarchies and measures and convert into search query logs. According to that annotated cube lattice is constructed using sample data.

(ii) The Annotated cube lattice is constructed using Value partitions which are of reducer unfriendly regions and batch areas techniques are used.

(iii) In cube materialization using Map-Reduction technique tuples are mapped to each batch areas. Reducer evaluates the measure for each batch area. Then cube is loaded into DB for future exploration.

After that according to user queries selecting and executing appropriate cubes in database is take place.

As shown in following table, data sets are maintained as a set of tuples. Each tuples has a set of attributes, such as ip and query. For many analyses, it is more desirable to map some raw attributes into a fixed number of derived attributes through a mapping function. For example, ip is mapped to country, city, state. Similar query is then mapped to topic, category subcategory.

A. Lattice Construction:

For example, Fig. 4 illustrates a cube lattice where the dimension attributes include the six attributes derived from ip and query.
For effectively parallelism we use Partitioning technique called Batch Area. As shown in fig 4. Each batch area represents a collection of regions that share a common parent. The combined process of identification and value partitioning Unfriendly regions and partitioning of regions into batches is referred to as annotate so lattice formed is annotated lattice.

![Annotated cube lattice](image)

Fig 4- Annotated cube lattice. Each color in the lattice indicates a batch area b1 to b5.

The cube region term is used to denote a node in the lattice and the term cube group is used to denote an actual value belonging to the cube region. Then two techniques required for efficiently distribute the data and computation task. As shown in figure 3 Value Partitioning is used partitioning groups that are reducer unfriendly and dynamically adjust the partition factor. The reducer unfriendliness of each cube region is estimated by sampling approach.

**B. Cube materialization using map-reduced**

In map reduced based approach, mappers are allocated to each batch area and it emits key: value pairs for each batch area. In required, keys based on value partitioning are used, then in shuffle phase sorted by using key. The BUC Algorithm is run on each reducer, and the cube aggregates are generated. All value partitioned groups need to be aggregated to compute the final measures.

**C. Data Aggregation Map-Reduce:**

Data aggregation is most important challenge which causes it to be from separate Map-Reduce that can be integrated with aggregation phase post –materialization. It is feasible to perform both large-scale cube materialization and mining in same distributed framework of similar interesting cube groups.

**V. CONCLUSION**

In this paper, we study annotation, aggregation, materialization and mining techniques for efficient cube computation. Proposed approach deals with cube groups instead of cube region to overcome workload of cube group computation. Thus MR-Cube successfully handles cube computation with dynamic measure over large datasets.

**REFERENCES**


[2]. Y. Zhao, P. M. Deshpande, and J. F. Naughton. “An array-based algorithm for simultaneous multidimensional aggregates”. In SIGMOD’97.


[5]. D. Xin, J. Han, X. Li, and B. W. Wah. Starcubing: Computing iceberg cubes by top-down and bottom-up integration. In VLDB’03


[8]. Fangbo Tao , Kin Hou Lei, “EventCube: Multi-Dimentional search and mining of structured and Text data” , ACM 978-1-4503-2174-7, 2013


[10]. Yixin Chen, Jiawei Han,“Stream Cube: An Architecture for Multi-Dimensional Analysis of Data Streams” Springer Science + Business Media, Inc. Manufactured in The Netherlands, Distributed and Parallel Databases , 2005.


[13]. Arnab Nandi, Cong Yu, Philip Bohannon, and Raghu Ramakrishnan “Data Cube Materialization and Mining over MapReduce“ IEEE transaction on Knowledge and Data Engineering, vol. 24, no. 10, Oct 2012.

