INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

# A REVIEW OF DATA CUBE MATERIALIZATION USING MAPREDUCE TECHNIQUES

MISS ABHILASHA A. DESHMUKH, PROF. O. A. JAISINGHANI

Department of Computer Science and Engineering, DRGIT&R, Ghatkheda, Amravati, India

**Accepted Date: 15/03/2016; Published Date: 01/05/2016**

**Abstract*:*** MapReduce is widely used and popular programming model for huge amount of data processing. Hadoop is open source implementation of MapReduce framework. Hadoop MapReduce is used for large data processing. It computes large amount of data in less time. The Performance of Hadoop depends some of the metrics like job execution time and cluster throughput. We provide extensive experimental analyses over both real and synthetic data. We demonstrate that, unlike existing techniques which cannot scale to the 100 million tuple mark for our data sets, MR-Cube successfully and efficiently computes cubes with holistic measures over billion-tuple data sets

**Keywords:** Hadoop, MapReduce, Slicing, Data Cube, Hive.

.

**Corresponding Author: MISS ABHILASHA A. DESHMUKH**

**Access Online On:**

www.ijpret.com

**How to Cite This Article:**

Abhilasha A. Deshmukh, IJPRET, 2016; Volume 4 (9): 494-502

***PAPER-QR CODE***

## 1. INTRODUCTION

In recent years the amount of data stored worldwide has exploded, increasing by a factor of nine in the last five years. Individual companies often have peta bytes or more of data and buried in this is business information that is critical to continued growth and success. However, the quantity of data is often far too large to store and analyze in traditional relational database systems, or the data are in unstructured forms unsuitable for structured schemas, or the hardware needed for conventional analysis is just too costly. And even when an RDBM is suitable for the actual analysis, the sheer volume of raw data can create issues for data preparation tasks like data integration. As the size and value of the stored data increases, the importance of reliable backups also increases and tolerance of hardware failures decreases. The potential value of insights that can be gained from a particular set of information may be very large, but these insights are effectively inaccessible if the it costs to reach them are yet greater. Hadoop evolved as a distributed software platform for managing and transforming large quantities of data, and has grown to be one of the most popular tools to meet many of the above needs in a cost-effective manner. By abstracting away many of the high availability and distributed programming issues. While Hadoop provides mechanisms to protect the data, the Name Node is a single point of failure, so in production clusters it is advisable to isolate it and take other measures to enhance its availability (this was not done for this paper since the focus was on performance).

## 2. Literature Review

The different authors are presenting the different methods which are previously used for anonymization. We discuss some advantages and limitations of these systems. Privacy preserving data analysis and collaborative data publishing has received considerable attention in current years as promising approaches for sharing data while preserving individual privacy.

### 2.1 Challenges in Big Data

Big Data has different characteristics such as it is large volume, heterogeneous, autonomous source with distributed and centralized control, seek to explore complex and evolving relationship among data. These different characteristics of Big Data make it challenge for discovering useful information or knowledge from it. After analyzing and research challenge form a three tier structure framework to mention different challenges at different tier.

### 2.2 Method Overview

#### 2.2.1 Data Cube

Data cube provide multi-dimensional views in data warehousing. If n dimensions given in relation then there are 2^n cuboids and this cuboids need to computed in the cube materialization using algorithm which is able to facilitate feature in MapReduce for efficient cube computation. In data cube Dimension and attributes are the set of attributes that user want to analyze. Cube lattice is formed representing all possible groupings of this attributes, based on those attributes. After that by grouping attribute into hierarchies and eliminating invalid cube regions from lattice we get more compact hierarchical cube lattice. Finally cube computation task is to compute given measure for all valid cube groups. There are different techniques of cube computations like multi- dimensional aggregate computation, BUC (Bottom-Up Computation), star cubing for efficient cube computation.

### 2.2.2 Cube Materialization

Cube materialization task comes under the MR-Cube approach. Materializing the cube means computing measures for all cube groups satisfying the pruning condition. After materializing cube we can identify the interesting cube groups for cube mining algorithm. The main MR-Cube and Map-Reduce task is perform using annotated lattice. The combine process of identifying and value partitioning unfriendly regions followed by partitioning of regions is referred as annotate.

### 2.2.3 MapReduce

MapReduce is a programming model designed for processing large volumes of data in parallel by dividing the work into a set of independent tasks. The nature of this programming model and how it can be used to write programs which run in the Hadoop environment is explain by this model. Hadoop is an open source implementation for this environment

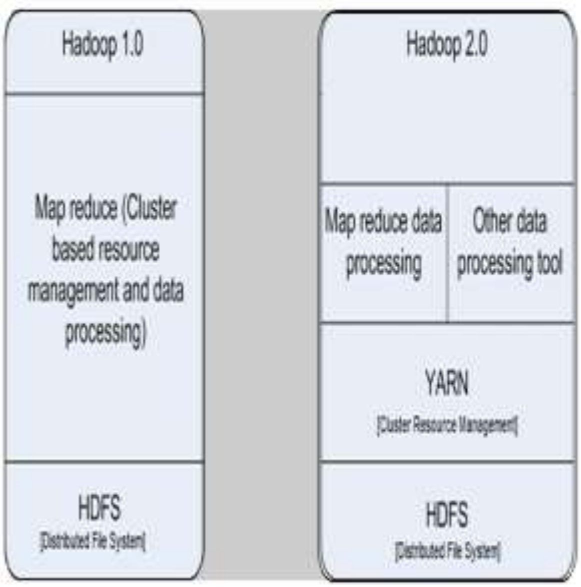
Hadoop 1.0
Map reduce (Cluster based resource management and data processing)
HDFS
[Distributed File System]
Hadoop 2.0
Map reduce data processing
Other data processing tool
YARN
[Cluster Resource Management]
HDFS
[Distributed File System]

**Fig 2.1: Architecture of Hadoop1.0 and 2.0**

MapReduce is used in Hadoop1.0 but due to some resource management issues like inflexible slot configuration, scalability. After Hadoop version 0.2, MapReduce changed significantly. Now it known as MapReduce 2.0 or YARN. MapReduce 2.0 has two major functionalities of job tracker which are spit into resource management. In Hadoop1.0 Job Tracker has a responsibility for managing the resources and scheduling jobs across the cluster. But in Hadoop2.0 the architecture of YARN allows the new Resource Manager to manage the usage of resources across all applications. And Application Masters takes the responsibility of managing the job execution.

This new approach improves the ability to scale up the Hadoop clusters to a much larger configuration than it was previously possible. In addition to this, YARN permits parallel execution of a range of programming models. This includes graph processing, iterative processing, machine learning, and general cluster computing.

### 2.2.4 MR-cube Approach

MR-Cube is a MapReduce based algorithm introduces for efficient cube computation and for identifying cube sets/groups on holistic measures. MR-Cube algorithm is used for cube materialization and identifying interesting cube groups. Complexity of the cubing task is depending upon two aspects: size of data and size of cube lattice. Size of data impacts size of large group and intermediate size of data, where as the cube lattice size impacts on intermediate data size and it is controlled by the number/depth of dimension.

#### 2.2.4.3 Batch Areas

Given the annotated cube lattice, process each cube group independently with the added safeguard that partitions the groups that belong to a reducer-unfriendly region. This partially alleviates the problem of large intermediate data size. Furthermore, another significant drawback of the naive approach is its incompatibility with pruning for monotonic measures, i.e., each cube group is processed independent of its parent group, we can no longer prune a group's children based on the having conditions such to address those problems, we propose to combine regions into batch areas. Each batch area represents a collection of regions that share a common ancestor region.A batch area typically contains multiple regions with parent/child relationships.

#### 2.2.4.4 Cube Mining

As discussed previously the highlighting of cube groups that may be interesting to the user is quite desirable. Materializing the cube (i.e., computing measures for all cube groups satisfying the pruning conditions) is often only the first step in the process of identifying interesting cube groups. Such tasks are trivial when the size of the full cube is tenable and when the interestingness can be defined as a simple value predicate. However, analysts often require more complex measures of interestingness. For example, the query “which city had the highest reach for each product category?” requires a comparison of groups in the city; category along the city dimension.

### 2.3 A Data Anonymous Method based on Overlapping Slicing

The idea of fuzzy clustering the attributes are mainly based on the idea of fuzzy clustering and present a linear algorithm of processing data with group to generate multiple data tables, and make them satisfy l-diversity. The system also conduct several experiments to confirm that overlapping slicing technology ensures data security and improves the effectiveness of anonymous data at the same time. The overlapping slicing processes high-dimensional data effectively. System assume there is no intersection between quasi-identifier attribute and sensitive attribute, namely, sensitive attributes of these records do not appear in the other data set, which has been released.

### 2.4 Privacy Preserving Research for Re-publication Multiple Sensitive Attributes in Data

They have developed a new generalization principle that effectively limits the risk of Multiple Sensitive Attributes privacy disclosure in re-publication. The results show that algorithm has higher degree of privacy protection and lower hiding rate. This paper presents an analytical study that various inference channels of publishing of dynamic multiple sensitive attribute dataset and discuss how to avoid such inferences.

### 2.5 Safe Realization of the Generalization Privacy Mechanism

Focuses on the organization of the collection and anonymization phases at the data source i.e., at each SPT.Given system focused precisely addresses this issue and proposes to adapt the traditional Generalization privacy mechanism to an environment composed of a large set of tamper-resistant smart portable tokens seldom connected to a highly available but untrusted infrastructure. This conjunction of hypothesis makes the problem fundamentally different from any previously studied privacy-preserving data publishing problem we are aware of. This work paves the way for a new family of privacy preserving distributed protocols exploiting the emergence of more and more powerful smart tokens. Future work will mainly consist in generalizing the approach to a wider variety of privacy mechanisms. The results presented in this paper are a strong incentive to go in this direction.

## 3. Proposed Plan Of Work

### 3.1 System Architecture

We first formally describe our problem setting. Then, we present our data-privacy definition with respect to a privacy constraint to prevent inference attacks by data- adversary, followed by properties of this new privacy notion. Let T = {t1, t2, . .} be a set of records with the same attributes gathered from n data providers P = {P1, P2, . . . , Pn}, such that Ti are records provided by Pi. Let AS be a sensitive attribute with a domain DS. If the records contain multiple sensitive attributes then, we treat each of them as the sole sensitive attribute, while remaining ones we include to the quasi-identifier. However, for our scenarios we use an approach, which preserves more utility without sacrificing privacy. The goal is to publish an anonymized T* while preventing any data-adversary from inferring AS for any single record. A data-adversary is a coalition of data users with n data providers cooperating to breach privacy of anonymized records.

When data are gathered and combined from different data providers, mainly two things are done, for anonymization process. To protect data from external recipients with certain background knowledge BK, I assume a given privacy requirement C is defined as a conjunction of privacy constraints: C1∧C2∧...∧Cw. If a group of anonymized records T* satisfies C, we say C(T*) = true. By definition C(Ø) is true and Ø is private. Any of the existing privacy principles can be used as a component constraint Ci. We now formally define a notion of data-privacy with respect to a privacy constraint C, to protect the anonymized data against data-adversaries. The notion explicitly models the inherent data knowledge of a data-adversary, the data records they jointly contribute, and requires that each QI group, excluding any of those records owned by a data-adversary, still satisfies C.

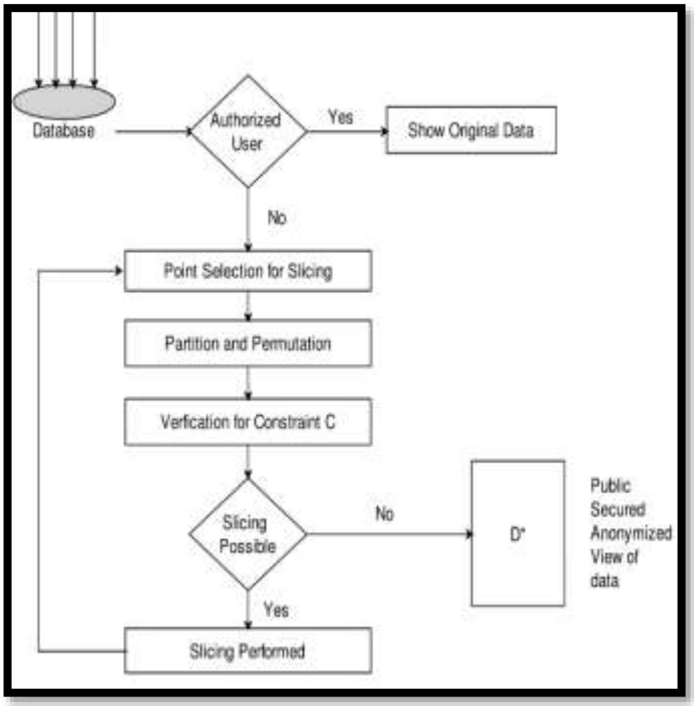
Database
Authorized User
Yes
Show Original Data
No
Point Selection for Slicing
Partition and Permutation
Verfication for Constraint C
Slicing Possible
No
D*
Public Secured Anonymized View of data
Yes
Slicing Performed

**Figure 3.1. System Architecture**

Fig.3.1 shows our proposed system in which input data is given from different providers. Select point for slicing. Check that input data against privacy constraint C for data privacy. Check further is slicing is possible or not. If slicing possible then do it and if not then display the output data. Our final output T* are anonymized data which will seen only by authenticate user. Any adversary cannot breach privacy of data. In this system we are using horizontal as well as vertical partitioning over database. Slicing algorithm provide better column partitioning. To understand this properly let's consider hospital management system for experiment. Let different departments are the providers who provide data from different sources. We consider disease as a AS (sensitive attribute) and age and zipcode are QI (quasi identifier).

### 3.2 Related Terminologies

In the proposed research work we implement the big data application on hadoop architecture, in that whole implementation we covered the below research methodologies

**a: Cube Query-** First in the first phase of application we generate the cube query it will directly execute on database

**b: Map reduce-** Map reduce is the related how to query will execute on searching time schema. Cube query execute with cluster base execution. It first classify all records in different maps and collect the results in different lists.

**c: Materialize view**- In this phase data will collect from different list as materialize view. it may be a local copy of data located remotely, or may be a subset of the rows and/or columns of a table or join result, or may be a summary based on aggregations of a table's data.

**d: Slicing / Secrecy View-** In the final phase we display all the data on browser we will display it with secrecy view like slicing it will provide additional security to data.

### 4.Conclusion

Above system can be used in many applications like hospital management system, industrial areas where we like to protect a sensitive data e.g. salary information of the employee. Pharmaceutical company where sensitive data may be a combination of ingredients of medicines, in banking sector where sensitive data is account number of customer, balance etc. This proposed system help to improve the data privacy and security when data is gathered from different sources and output should be in collaborative fashion.

### 5. REFERENCES

*1.* Arnab Nandi, Cong Yu, Philip Bohannon, and Raghu Ramakrishnan, Fellow, IEEE, “Data Cube Materialization and Mining overMapReduce” *TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 24, NO. 10, OCTOMBER 2012.*

*2.* Xindong Wu, Fellow, IEEE, Xingquan Zhu, Gong-Qing Wu, and Wei Ding” Data Mining with Big Data” *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 26, NO. 1, JANUARY 2014.*

3. J. Gray, S. Chaudhuri, A. Bosworth, A. Layman, D. Reichart, M. Venkatrao, F.Pellow, and H. Pirahesh, Data Cube: A Relational Operator Generalizing Group- By, Cross-Tab and Sub-Totals, Proc.12th Intl Conf. Data Eng. (ICDE), 1996.

4. 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.*

5. J. Dean and S. Ghemawat,"Mapreduce: Simplified Data Processing on Large Clusters," Comm. ACM, vol. 51, pp. 107-113, Jan. 2008.

6. J Jery Hanson, "An introduction to the Hadoop Distributed File System", IBM DeveloperWorks, 2011.

7. A. Abouzeide t al.,"HadoopDB: An Architectural Hybrid of MapReduce and DBMS Technologies for Analytical Workloads," Proc. VLDB Endowment, vol. 2, pp. 922- 933, 2009.

8. L. Buzzi, M. Bardellini, D. Siracusa, G. Maier, F. Paolucci, F. Cugini,L. Valcarenghi, and P. Castoldi, "Hierarchical border gateway protocol(HBGP) for PCE-based multi-domain traffic engineering," in Proc. 2010ICC.

9. Jing Yang and Ziyun Liu , Yangyue , Jianpei Zhang, "A Data Anonymous Method based on Overlapping Slicing", in Proceedings of the 2014 IEEE 18th International Conference on Computer Supported Cooperative Work in Design.

10. Xiaolin Zhang, Lifeng Zhang, "Privacy Preserving Research for Re-publication Multiple Sensitive Attributes in Data", in 978-1-4244-8728-8/11/$26.00 ©2011 IEEE.

11. Tristan Allard, Benjamin Nguyen, Philippe Pucheral proposed, " Safe Realization of the Generalization Privacy Mechanism", 2011 Ninth Annual International Conference on Privacy, Security and Trust.