DOI: http://dx.doi.org/10.26483/ijarcs.v9i1.5169

Volume 9, No. 1, January-February 2018

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

EFFICIENT COMPUTING OF OLAP IN BIG DATA WAREHOUSE

Prof. Jigna Ashish Patel
Assistant Professor, CE Department
Nirma University, Ahmedabad, Gujarat, India

Dr. Priyanka Sharma
Professor and Head,CE Department
Rakshashakti University, Ahmedabad, Gujarat, India

Abstract: With the wide and fast development of tools and technology in big data era, new challenges in development of OLAP and data warehousing took place. To manage big data, distributed environment and Hadoop framework are only the solution. Exponentially increasing data create scalability issue in any model, map reduce programming model resolves that problem. Using these technologies we present series of algorithms combined in our model OOH(Olap on Hadoop) . Measures and dimensions are modelled using DET (Dimension encoding Technique) and DTT (Dimension Traversal Technique), so that Rollup and Drilldown operation can be performed well.

Keywords: Data warehouse; OLAP; Hadoop

1. INTRODUCTION

Due to recent advancement in technology, social media usage and internet awareness, lot of data is generated. Exponentially data get produced, uploaded and downloaded. It is high time to think upon the Business analytics and intelligence [1]. Plenty of tools available for Data Warehousing and data mining, but only few of them are applicable or feasible for big data. According to Gartner, Big data deals with Four V's. Volume, Variety, Veracity and Velocity. Every V has its own challenges and probable solutions. In order to make more feasible solution industries and academia both try to find out the cost effective resolutions to overcome challenges generated by big data. Online analytical processing involves operations like rollup, drill down, slice, dice and many more.[2] For judgments in businesses, analytics reports are generated and decisions based on OLAP operations and visual reports play very important role. Data analytic and data science have lot of branches under it. OLAP and data warehousing are classic field of it whichis in the research since long [3].

Computing of big data OLAP requires lot of challenges like scaling of data, speed of processing, storage of data, query performance and lot of others. In this paper we mainly focus on two challenges of big data as storage and velocity over OLAP. Data warehousing methods also known as data harmonization techniques.[4] Customary data warehouses involves only structured data but Contemporary data warehouse provides solution for varieties of data like semi structured data or unstructured data. Social media data, audio data, images, audio visual data, text data are very well known examples of modern datasets. Here every type of data generate the appealing challenges to create OLAP cube in big data era[5]

OLAP cube can be generated in many different ways two popular methods amongst them is ROLAP (Relational Online Analytical Processing) and MOLAP (Multidimensional Online Analytical Processing). Basically ROLAP is used for SQL type relational databases. Star schema and snow flake schema are well known methods to achieve ROLAP. In big data age, to have join operation on different tables produce more costlier result. It is very difficult to achieve good result when we deal with big data scalability issue.[6] Contrast with MOLAP, ROLAP requires more space to store all tables and to process all the tables again we need few join operation which become more and more expensive. As far as the MOLAP is concern we deal with multidimensional array. MOLAP provides robust performance for scalability and data storage [11]

Figure-1shows the visualization of data in multidimensional view. It can be extended for multidimensional view as it is only shown for three dimension. Our aim is to divide the OLAP cube into fixed size chunks and all chunks will be processed parallelly in order to achieve distributed work using map reduce framework. Figure 2 shows the division of cube into chunks to achieve parallelization.

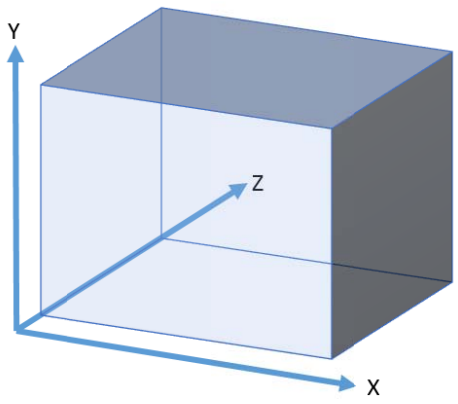


Fig.1 multidimensional data cube

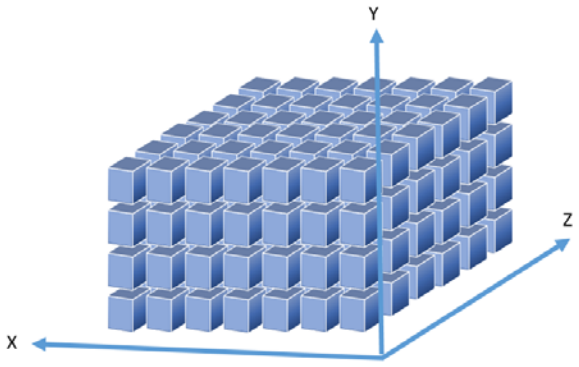


Fig. 2 visualization of partitioned cube

As we dealt with multidimensional data model or MOLAP figure-1 shows the multidimensional data cube, chunking is not a new approach in data warehousing. We used chunking method in order to utilize the distributed data environment model. [8] To process the large volume of data parallel processing and distributed working environment is only the solution. We applied horizontal scaling and Hadoop is the best solution to achieve same.[10]

The paper is organized in mainly four sections, first we introduced problems in existing environment then series of algorithms and then the implementation and results are discussed. Dimension encoding algorithm is responsible for encoding and decoding of actual data which will resultant into lesser storage space. Dimension traversal algorithm is responsible for Roll-up and drill down operation where ups and downs for levels of each dimension.

2. PROBLEMS IN EXISTING ENVIRONMENT

By looking at the literature survey and related work, at present the deficiency inoperationalprovision for multidimensional data storing model and OLAP analysis. It definitely needs to be resolved straightaway in big data era. At the same time Hadoop is most widely used to resolve scalability issue.[7] To address scalability issue and performance challenges for MOLAP of big data, map reduce programming resolve it using distributed data model. As far as the performance of the OLAP is concerned we use the chunking/ partitioning method to store the big data. To retrieve the particular queried data from the distributed data among number of nodes we process it using indexing method.

Concept of Indexing is used in order to reduce processing unwanted data [9]. Seeking to a particular data field extract wanted data,even though it is in a different chunk and return the pointer to the initial stage.

Series of algorithms are applied in order to escape more storage cost, in OOH we adopted basic and simple data model and advanced algorithms. OOH we adopted basic and simple data model and advanced algorithms. In OOH, we used DET (Dimension encoding Technique) and DTT (Dimension Traversal Technique), DET will solve sparsity problem in multidimensional array as we have used integer encoding technique. Dimension traversal algorithm is used for Roll up and Drill down operation. In following section we mentioned all the algorithms we used for our approach.

3. ALGORITHMS

DET (Dimension encoding Technique)
DTT (Dimension Traversal Technique)
DST (Data Storage Technique)

DET (DIMENSION ENCODING TECHNIQUE)

Principally two dimension coding techniques existing for encoding purpose. Binary encoding and integer encoding both have their own pluses and minuses. To avoid sparsity in multidimensional array we can use integer encoding and to gain level wise information directly we can use binary encoding. We used integer encoding technique to avoid sparsity problem.

Let dim_level be a dimension level of dimension dim
Input: dimension dim (“Targeted dimension”)
Process:

```
For i=0 to |all_level(dim)|
        For j=0 to |size(dim)=□i=1all_level|dimi||
                Cji belongs to |size(dim)|
                Cji=j
        End for
End for
```

DATA STORAGE TECHNIQUE

In order to reduce the storage cost required by OLAP in big data, it is highly essential to serialize the data. In MOLAP storing OLAP requires more space as we need to store multidimensional array and in big data which becomes larger. So in OOH we decided to calculate multidimensional array but we don’t store it. We directly take the data from database server and store in serialized fashion, which will take key and value instead of storing n dimensional data and its value.

In OOH the chunk file and the cells of block are serialized for resolution and deserialize for request-query from user.Chunk file is nothing but the map file given to our mapper stored in Mapfile. Sensibly, cube cells and chunk files are connected with the values of multidimensional array only but actually they are the mapfile of HDFS.

Let X be the multidimensional array with n dimensions as {D1, D2, D3… Dn} Co-ordinates of array values are denoted as {P1, P2, P3… Pn}, Serialization

Index(X) = [D1 + [D2 * P1] + [D3 * P2] + ……. + [Dn*Pn-1]]… … … … eq-1

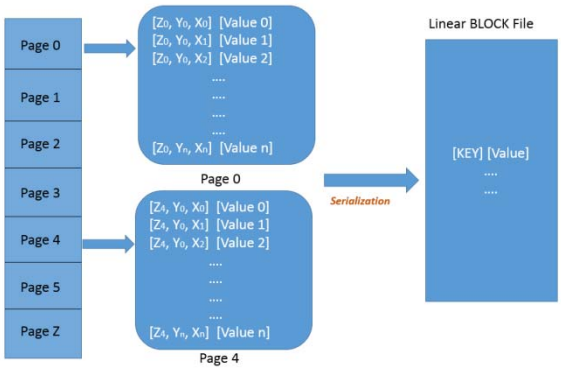


Fig 3: Serialization

As shown in figure-3 the paging concept is clearly visible instead of storing the value using multidimensional array we can serialize it in a key.

Similarly deserialize concept is applicable when we process the query.in order to find out the coordinates we can reversely apply the same concept.

Deserialization

T1 = index

P1 = T1 % D1 T2 = T1 / P1

P2 = T2 % D2 T3 = T2 / P2

.

..

…

Pn = Tn % Dn

Dimension traversal Technique

In OOH, to perform roll up and drill down, we devised a technique called DTT. The hierarchy tree is generated from all dimension values. In hierarchy tree the dimension level is treated as node of a tree.

Drill down and roll up are the operations performed with the logic if value of C i+1 is given which will go till C I called as drill down operation. Similarly, roll up operation if Ci to C i+1. Following operations are to be applied in order to achieve.

Value (Ci) = [(order (Ci) + |Ln| X (order (Ci-1) + |Ln-1| X . . . . X (order (C1) + 0] ----- eq (2)

From above equation we can derive the relation of Value (Ci) and (order (Ci)

finali = Value (Ci)

(order (Ci) = finali % |Li|

finali-1 = finali / |Li|

(order (C1) = final1 % |Li| ----------eq(3)

Based on equation (3) and (4) Value (Ci-1) to Value (C1) can be calculated. if Value (Ci) is given, then the order of whole Path can be found. Eg. At day level Value (23)= 32.

<19901 , 22 , 23> is found. value(23)

4. IMPLEMENTATIONS AND RESULTS

As there is no impact of domain or an application on our model, we can choose any database to test our model. We downloaded the oceanography data of around 10GB to validate our model. Mainly the oceanography database includes three dimension, Time, Area and Depth. For all these dimensions we have number of levels too. Following figure shows number of levels for each dimension.

(i)Time = {<Year>, <Season>, <Month>, <Day>, <Slot>};

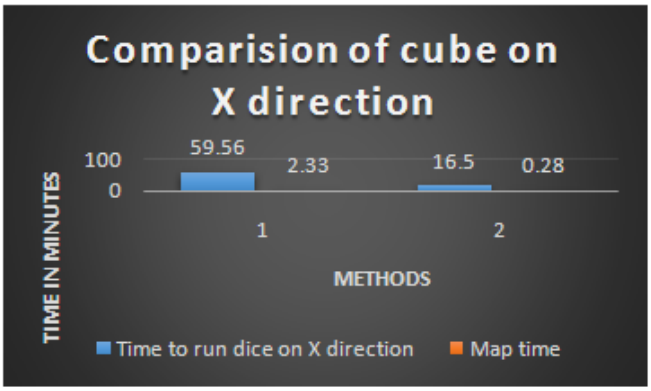
(ii)Area A = {<1°>, <1/2°>, <1/4°>, <1/8°>, <1/16°>, <1/32°>, <1/64°>};

(iii)Depth D= {<100m>, <50m>, <10m>}.

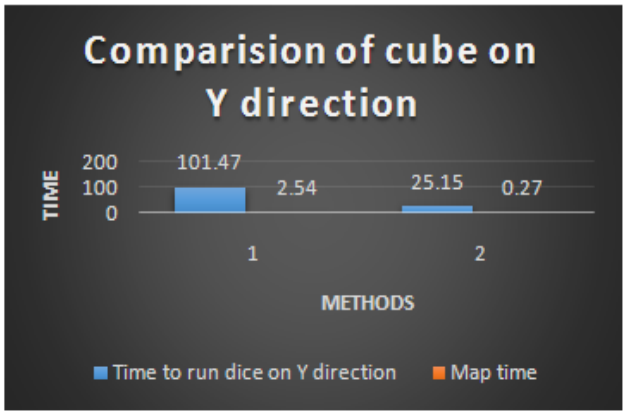
We implemented our Algorithms on Hadoop. For the implementation in OOH we used map reduce framework. Input-formatter, mapper, reducer and output formatter are the key four parts in which the map reduce job executes. The query quadruple is submitted by the client and verified by the job node to avoid process failures.

As and when query quadruple is submitted by client and verified for deterministic failures. Input formatter takes data from chunk selection algorithm as chunk list. Every coordinates of cell are deserialize and checked by query conditions. If coordinates match the query condition, serialized coordinate of the cell are pass to mapper. Firstly the chunk selection file is detected by job node. It will scan all the cells and chunks, parallelly coordinates of the cell are deserialize and checked by the query conditions. If the coordinates find match with the serialized coordinate of the cell then it is pass to mapper. Mapper work with the < key, value > pairs.

In existing models , Input formatter deserialize every key and value and check by condition, mapper used to process a single element and wait till next element, which is simple a brute force technique. OOH, input formatter runs DST and DET algorithm which reads the block of data so it process the array of element and mapper will process the bunch of data and meanwhile input formatter is ready with the next block of data and hence, it is much faster than existing models.



Plot 1: comparison of cube on X direction



Plot 2: comparison of cube on Y direction

5. CONCLUSION AND FUTURE ENHANCEMENTS

In this paper we exemplify the model to execute OLAP operations in Hadoop environment successfully. We elaborated the scalability and data loading issue perfectly with the solution. Evaluation and comparison of OOH is shown with existing models (HaOLAP), existing approach for Hadoop based OLAP operations. Future enhancement at this stage is to involve more operations like pivot or rotate with OLAP and to abide with the current model.

REFERENCES

[1] J. Song, C. Guo, Z. Wang, Y. Zhang, G. Yu, and J.-M. Pierson, "HaoLap: A Hadoop based OLAP system for big data," Journal of Systems and Software, vol. 102, pp. 167–181, Apr. 2015.

[2] J. A. Patel and P. Sharma, "Big data for better health planning," in Advances in Engineering and Technology Research (ICAETR), 2014 International Conference on, 2014, pp. 1–5.

[3] Blanco, I. García-Rodríguez de Guzmán, E. Fernández-Medina, and J. Trujillo, "An architecture for automatically developing secure OLAP applications from models," Information and Software Technology, vol. 59, pp. 1–16, Mar. 2015

[4] D.-H. Shin and M. J. Choi, "Ecological views of big data: Perspectives and issues," Telematics and Informatics, vol. 32, no. 2, pp. 311–320, May 2015.

[5] J. Patel and P. Sharma, "Decision Support System in Diabetes Disease with Providing Health Care Services," 2015.

[6] Avita katal,Mohammad wazid,R H Goudar," Big data:, Issues,Challenges,Tools and Good practices" IEEE 2013.

[7] J. Li, L. Meng, F. Z. Wang, W. Zhang, and Y. Cai, "A Map-Reduce-enabled SOLAP cube for large-scale remotely sensed data aggregation," Computers & Geosciences, vol. 70, pp. 110–119, Sep. 2014.

[8] A. Cuzzocrea, L. Bellatreche, and I.-Y. Song, "Data warehousing and OLAP over big data: current challenges and future research directions," in Proceedings of the sixteenth international workshop on Data warehousing and OLAP, 2013, pp. 67–70.

[9] Xin Cheng,Chungjin Hu,Yang Li, Wei Lin, Haolei Zuo., "Data Evolution of virtual dataspace for managing the big data lifecycle ",IEEE 2013

[10] S. Mansmann, N. Ur Rehman, A. Weiler, and M. H. Scholl, "Discovering OLAP dimensions in semi-structured data," Information Systems, vol. 44, pp. 120–133, Aug. 2014.

[11] Big data survey research brief ,2013, www.sas.com/resources/whitepaper/wp_58466.pdf