Journal of Engineering and Applied Sciences 14 (20): 7731-7735, 2019
ISSN: 1816-949X

Big Data Harmonization-Data Loading and Data Storage

[1]Jigna Patel, [2]Priyanka Sharma and [1]Jitali Patel
[1]Department of Computer Science and Engineering (CSE), Institute of Technology,
Nirma University, Ahmedabad, India
[2]Department of Information Technology (IT), Rakshashakti University, Ahmedabad, Gujarat
jignas.patel@nirmauni.ac.in

Abstract: With the wide and fast development of tools and technology in big data era, new challenges in development of OLAP and data harmonization become the essential. Data harmonization provide the common level of granularity from the heterogeneous data sources and with the variety of data formats. 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) to show the data loading and data storage process.

Key words: Data warehouse, OLAP, Hadoop, heterogeneous data, programming model, data storage process

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 (Song *et al*., 2015). 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 (Patel and Sharma, 2014). For judgements 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 which is in the research since long (Blanco *et al*., 2015).

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 study we mainly focus on two challenges of big data as storage and velocity over OLAP. Data warehousing methods also known as data harmonization techniques (Shin and Choi, 2015). 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 (Patel and Sharma, 2015).

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 (Katal *et al*., 2013). 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 (Anonymous, 2013).

Figure 1 shows 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

Corresponding Author: Jitali Patel, Department of Computer Science and Engineering (CSE), Institute of Technology, Nirma University, Ahmedabad, India, jignas.patel@nirmauni.ac.in

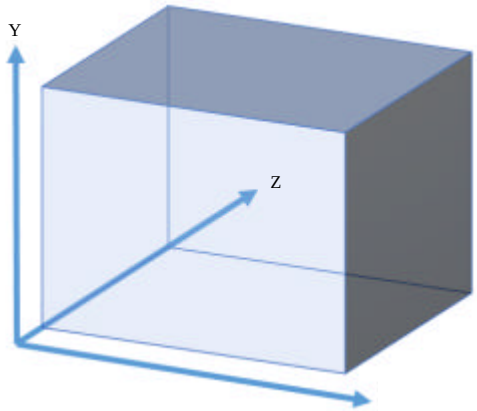

Fig. 1: Multidimensional data cube

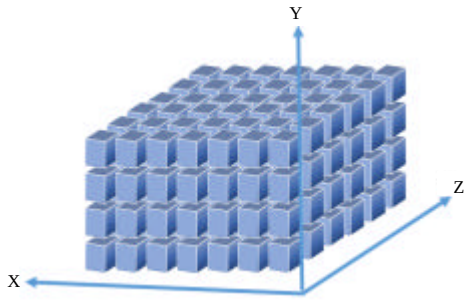

Fig. 2: Visualization of partitioned cube

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.

As we dealt with multidimensional data model or MOLAP Fig. 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 (Cuzzocrea *et al*., 2013). 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 (Mansmann *et al*., 2014).

The study 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.

MATERIALS AND METHODS

Problems in existing environment: By looking at the literature survey and related work, at present the deficiency in operational provision 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 (Li *et al*., 2014). 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 (Cheng *et al*., 2013). 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. In OOH, we used DET (Dimension Encoding Technique) and DET will solve sparsity problem in multidimensional array as we have used integer encoding technique. Map-reduce based data loading will help in time reduction to load data into warehouse. Data storage mechanism apply indexing to reduce efforts in storage and process.

Algorithms: DLT (Data Loading Technique), DST (Data Storage Technique), DET (Dimension Encoding Technique).

DST (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.

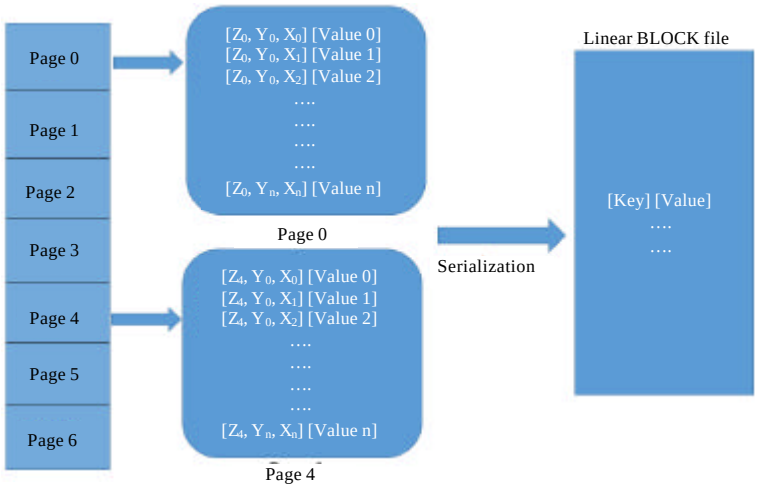

Fig. 3: Serialization

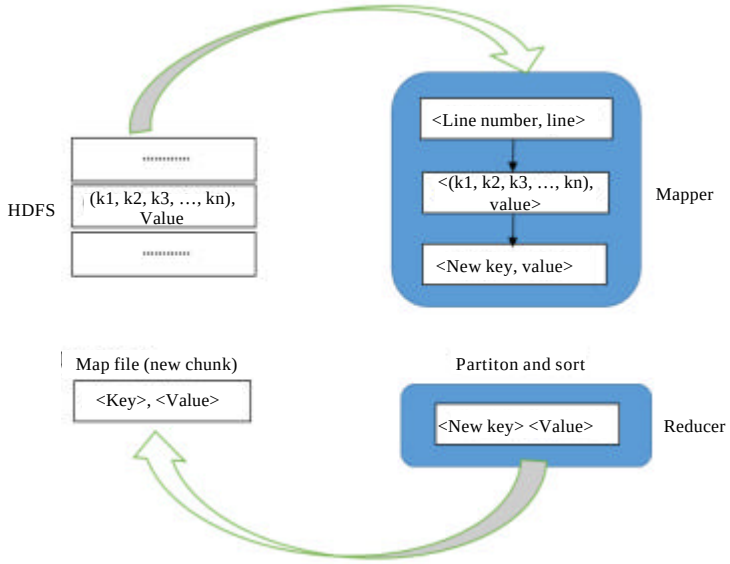

Fig. 4: Data loading technique

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:

$$Index(X) = [D1+[D2*P1]+[D3*P2]+, ...,+[Dn*Pn-1]] \quad (1)$$

As shown in Fig. 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

DLT (Data Loading Tecnique): Data loading implementation involves two phases. First, phase will load data to HDFS (Hadoop Distributed File System) and second phase is responsible for generating chunk files through map reduce process as shown in Fig. 4.

The data loaded to Hadoop distributed file system is in the XML format, Parser will run to acquire dimensions,

measures and level information. Every line of the original file contains a sentence line number and the value. Here, sentence line number is dimension and measures. Input formatter, mapper, reducer and output formatter. Input formatter takes the raw data and apply the parsing to separate measures of dimensions, levels of dimensions and meta data.

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) = Π i = 1all_level|dimi||
Cji belongs to |size(dim)|
Cji = j
End for
End for

RESULTS AND DISCUSSION

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 10 GB 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. Figure 5 and 6 show number of levels for each dimension. Time = {<Year>, <Season>, <Month>, <Day>, <Slot>}; Area A = {<1°>, <1/2°>, <1/4°>, <1/8°>, <1/16°>, <1/32°>, <1/64°>}; Depth D = {<100 m>, <50 m>, <10 m>}.

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

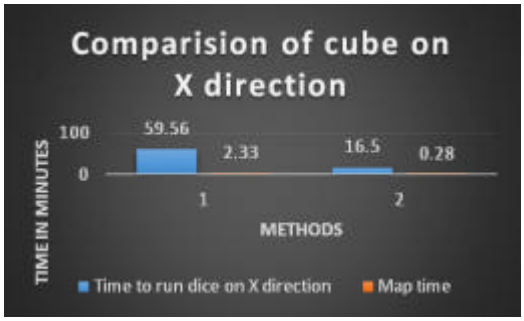

Fig. 5: Comparison of cube on X direction

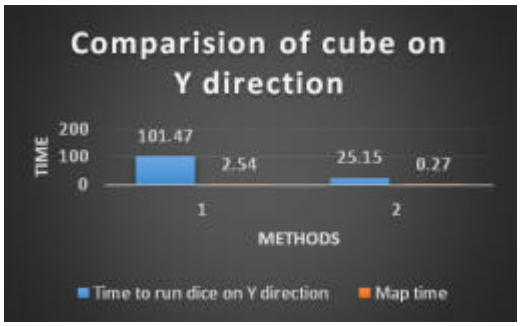

Fig. 6: Comparison of cube on Y direction

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 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.

CONCLUSION

In this study, we exemplify the model to execute OLAP operations in Hadoop environment successfully. We elaborated the data storage 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

Anonymous, 2013. Big data survey research brief. SAS Institute Inc., Cary, North Carolina, USA. https://webcache.googleusercontent.com/search?q=cache:nimF4XYSLUwJ:https://dsimg.ubm-us.net/envelope/145343/297972/1384815138_BigDataResearchBrief.pdf+&cd=1&hl=en&ct=clnk&gl=pk

Blanco, C., I.G.R. de Guzman, E. Fernandez-Medina and J. Trujillo, 2015. An architecture for automatically developing secure OLAP applications from models. Inf. Software Technol., 59: 1-16.

Cheng, X., C. Hu, Y. Li, W. Lin and H. Zuo, 2013. Data evolution analysis of virtual dataspace for managing the big data lifecycle. Proceedings of the 2013 IEEE International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum, May 20-24, 2013, IEEE, Cambridge, Massachusetts, pp: 2054-2063.

Cuzzocrea, A., L. Bellatreche and I.Y. Song, 2013. Data warehousing and OLAP over big data: Current challenges and future research directions. Proceedings of the 6th International Workshop on Data warehousing and OLAP Vol. 13, October 28-28, 2013, San Francisco, California, USA., ISBN:978-1-4503-2412-0, pp: 67-70.

Katal, A., M. Wazid and R.H. Goudar, 2013. Big data: Issues, challenges, tools and good practices. Proceedings of the 6th International Conference on Contemporary Computing (IC3) 2013, August 8-10, 2013, IEEE, Dehradun, India, ISBN:978-1-4799-0191-3, pp: 404-409.

Li, J., L. Meng, F.Z. Wang, W. Zhang and Y. Cai, 2014. A map-reduce-enabled SOLAP cube for large-scale remotely sensed data aggregation. Comput. Geosci., 70: 110-119.

Mansmann, S., N.U. Rehman, A. Weiler and M.H. Scholl, 2014. Discovering OLAP dimensions in semi-structured data. Inf. Syst., 44: 120-133.

Patel, J. and P. Sharma, 2015. Decision support system in diabetes disease with providing health care services. Intl. J. Adv. Eng. Technol., 1: 17-24.

Patel, J.A. and P. Sharma, 2014. Big data for better health planning. Proceedings of the 2014 International Conference on Advances in Engineering & Technology Research (ICAETR-2014), August 1-2, 2014, IEEE, Unnao, India, ISBN:978-1-4799-6392-8, pp: 1-5.

Shin, D.H. and M.J. Choi, 2015. Ecological views of big data: Perspectives and issues. Telematics Inf., 32: 311-320.

Song, J., C. Guo, Z. Wang, Y. Zhang and G. Yu *et al*., 2015. HaoLap: A hadoop based OLAP system for big data. J. Syst. Software, 102: 167-181.