A comparative Big Data Structure Research on r-Train and R-Tree

RongzhenWei
Department of computer science
George Mason University
Fairfax VA United States
rwei2@gmu.edu

Abstract- ***Big Data has been regarded as an emerging trend and the need for analysis of Big Data is arising in all science and engineering domains. The data structure that can properly manage big data is needed. In this paper, we conducted a research on big data, and looked for a better methodology for organizing bid data. Specifically, a comprehensive literature review regarding to big data is first presented for a deeper understanding. Next, r-Train and R-Tree, two data structure designed for big data, are studied. Also, a comparison between these two kinds of structure is provided.***

Key words: big data, data structure, r-Train, B-tree

1. INTRODUCTION

There are numerous well-developed data structures applied for various purposes with different philosophy. People usually divide the data structures into two types, namely, linear data structure and nonlinear data structure. The array is a very frequent-used linear data structure, and graphs and trees are two of the pervasively nonlinear data structures in computer science.

However, those traditional data structures seem not enough to efficiently manage the big data [1]. The topic – looking for a good data structure for big data has caught a great attention in both academic and industrial circles. Many researchers [1] [2] [3] have proposed useful ideas, and in the article, we will specifically introduce two data structures for big data – r-Train and R-tree.

Both r-Train and R-tree are developed from traditional data structures [4] [5] and based on data structure basis. r-Train is considered as a linear data structure that encapsulates the merits of the arrays and of the linked lists [6], and at the same time inherits a reduced amount of their characteristic demerits. R-tree [3], as a nonlinear data structure, takes advantage of R-tree with reduced index size reduced and diminished cost of data interrogation and handling [7].

In this paper, a review of big data will be presented, and then r-Train and R-tree will be introduced. A comparison and discussion of these two data structures will be provided. Finally, some suggestions based on the study of this paper will be given.

2. LITERATURE REVIEW

Big Data has been regarded as an emerging trend and the need for analysis of Big Data is arising in all science and engineering domains [4]. Obviously, the phrase - big data can be understood as the bigger and more complex data sets comparing to traditional data [1]. Their sizes are usually described by terabyte, petabyte, even exabyte [8]. Because of their massive size, the way of data management has been totally different [5].

For instance, if we have a tensor made by millions of columns and millions of rows, building index for the data will possibly take days, and what if sorting is needed? We have to admit that for this type of data, traditional methods do not look enough to be able to handle [6]. In the other word, the method of processing big data requires much more efficiency in both space and time [9].

A research [10] concludes that Big data is characterized by five Vs:

- Volume - This applies to the huge voluminous
- Variety - Sizably voluminous data could be in a number of different forms such as structured or unstructured, text, images, videos etc.
- Velocity - data that emanate at high speed.
- Variability - This refers to the data of cognition in real time,
- Value - The data should be valuable to the society.

Big data is also a wrapper for different types of granular data [11]. Below, five key sources of high volume data are listed as public data, private data, data exhaust, community data, and self-quantification data [12]. Many applications have been developed according to those data by companies and scientific institutions [11]. For storing and processing the huge volume data, new techniques like cloud computing have been used [12]. A popular computer tool that provides software framework to solve problems involving big data is Apache Hadoop [6]. It uses the MapReduce programming model to distribute storage and process data [12].

The results of big data analysis are commonly used to predict the future trends and conduct the decision making [4]. An interesting fact in big data is that those data are captured really fast, so the predictive results are expected to be presented as fast as possible. The less time analysis process takes, the more valuable the results will be [8]. Google has been trying to characterize the behavior of users by gathering data, like the restaurants users frequently go to, the movies users like and so on [10]. In this case, the data of most recent five years (2013 - 2018) have more significance than that from further five years as 2008 - 2013.

For those requirements, the data structure for big data should integrate multiple methods and techniques, especially mathematical model and statistical tool [6]. For example, the big data can be represented within multi-dimensional vector space, and topological data analysis method will be applied to resolve. Also, a lot of attempts have been put on enhancing and improving the existed data structure to fulfil the analytical needs. R-train and R-tree are both good examples of advanced data structure for big data [3] [12].

3. A COMPARISON OF R-TRAIN AND R-TREE

a. r-Train

r-Train [4], initially proposed by R. Biswas, is a dynamic data structure based on both array and linked list. The term “train” here derived from the usual railways transportation systems for there are a lot of similarities in the object r-train with the actual train of railways [4]. Analogously, the terms coach, passenger, availability of seats, etc. will be used in the way of discussion for r-train [4].

A r-train is basically a linked list of tagged coaches, so a linked list can be considered as a r-Train with r=1. The linked list is called the ‘pilot’ of the r-train [4]. The number of coaches in a r-train denotes the ‘length’ of the r-train which may increase or decrease depending on the data. A r-train T of length l (>0) will be written as the following notation [4]

$$T = < (C1, sC1), (C2, sC2), (C3, sC3), \ldots\ldots, (Cl, sCl) >,$$

where the coach Ci is a l array of length r. the element ei is the address of the next coach Ci+1 (or an invalid address in case Ci is the last coach). sCi is the status of the coach Ci.

The length l of the pilot could be any natural number, but the l arrays of the TCs are each of fixed length r which store data elements (including elements) of common datatype where r must be a natural number.

Taking matrix representation as an example, if we write a n*n matrix A by row vectors as

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} A_1 \\ \vdots \\ A_n \end{bmatrix}$$

It can be represented by a n-Train data structure X as

$$X =< X_0, X_1, \dots, X_{n-1} >$$
$$with\ X_0 =< x_{00}, x_{01}, \dots x_{0n-1}, e_1 >$$
$$X_1 =< x_{10}, x_{11}, \dots x_{1n-1}, e_2 >$$
$$\vdots$$
$$X_{n-1} =< x_{n-10}, x_{n-11}, \dots x_{n-1n-1}, e_n >$$

Where $X_0, X_1, \dots, X_{n-1}$ are coaches representing the row vectors, and $x_0,\ x_{01},\ \dots x_{n-1n-1}$ are the passengers representing the matrix elements, and $e_1,\ e_2,\ \dots e_n$ are storing the addresses of their coaches.

b. R-tree

R-tree [6], was originally proposed by Guttman in 1984, aimed at handling geometrical data, such as points, line segments, surfaces, volumes, and hypervolumes in high-dimensional spaces [6]. This data structure, apparently, is based on B-tree and B+-tree [6]. Generally Speaking, R-tree data structure is designed to be as an additional access method to handle multi-dimensional data.

An R-tree of order (m,M) has the following characteristics [5]:

(1). Each leaf node (unless it is the root) can host up to M entries, whereas the minimum allowed number of entries is m ≤ M/2. Each entry is of the form (mbr, oid), such that mbr is the minimum bounding d-dimensional rectangles that spatially contains the object and oid is the object's identifier.

(2). The number of entries that each internal node can store is again between m ≤ M/2 and M. Each entry is of the form (mbr, p), where p is a pointer to a child of the node and mbr is the minimum bounding d-dimensional rectangles contained in this child.

(3). The minimum allowed number of entries in the root node is 2, unless it is a leaf (in this case, it may contain zero or a single entry).

(4). All leaves of the R-tree are at the same level.

From the definition of the R-tree, it is

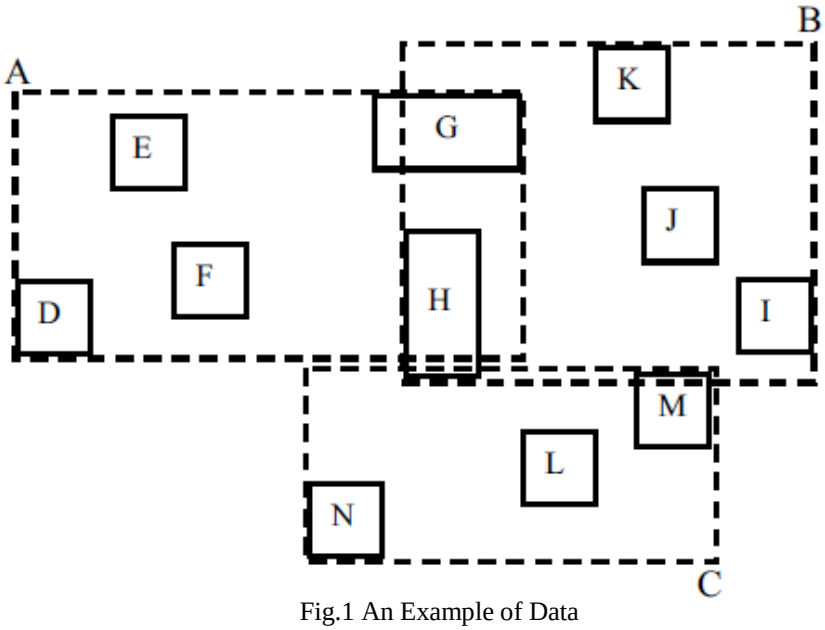

Fig.1 An Example of Data

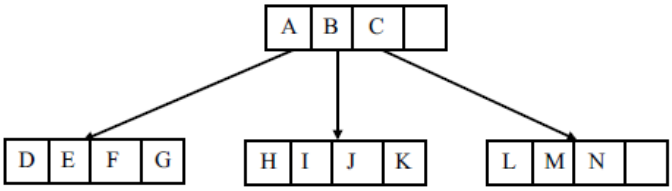

Fig.2 The corresponding R-tree

guaranteed that R-train is a height-balanced tree [6].

Figure 1 is about a set of the MBRs of some data geometric objects (not shown). These MBRs are shown by vertices D, E, F, G, H, I, J, K, L, M, and N, which will be stored at the leaves of the R-tree. Also, it demonstrates A, B, and C that organize the aforementioned rectangles into an internal node of the R-tree. If we Assume that M = 4 and m = 2, Figure 2 indicates the corresponding MBR. It is evident that several R-trees can represent the same set of data rectangles. Each time, the resulting R-tree is determined by the insertion (and/or deletion) order of its entries.

c. Comparison

In the beginning of this paper, two types of data structure are mentioned – linear and nonlinear. From this perspective, r-Train is a linear data structure, which arranges data into a sequence and follows a certain order, whereas R-tree is a nonlinear data structure, which cares more about how the data is organized, accessed, associated and processed.

Hence, the applications of the data structures depend on the data and the usage of data. For instance, if we are simulating and trying to forecast the weather change in a large region, using r-tree will be a better choice as data are supposed to be sorted by time and locations. A multidimensional matrix will be suitable. However, if we attempt to analyze the weather statistically, R-tree will be the priority option.

A compare table of these two data structures are given blew,

Table.1 Comparison between R-tree and r-Train

Name	r-Train	R-tree
Type	Linear	Nonlinear
Basis	Combination and improvement of array and linked list	Improvement of B-tree and B+-tree
Big-O (Search)	O(n)	O(log n)
Big-O (insert/delete)	O(n)	O(log n)
Logic	Matrix	Set

d. Suggestions on big data structure approach

Big Data are high-volume, high-velocity and high-variety information assets that require new forms of processing to enable enhanced decision making, insight discovery and process optimization. This new form for big data should integrate multiple methods and techniques, especially mathematical model and statistical tool.

We have seen that the combination of array and linked list can make r-tree. What if we go further, and make more combinations? For example, linear data structure and nonlinear can be integrated.

Imagine that we have a matrix A as shown; of course, it is a two-dimensional array,

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix}$$

Where each element $a_{11}, a_{12}, \dots, a_{nn}$ are the root of a tree or a heap.

4. CONCLUSION

The importance of big data analysis has been considerably increasing, such that finding and using a good data structure have become necessary and important for big data. People have spent a lot of time and efforts on it, and we have had some progress.

In this research, we reviewed the basis of big data and introduced two data structures, which are r-train and R-tree. Such kind of overwhelming data could not be handled by traditional way; thus, many new methods are designed, and new models are set up. Big data actually is pushing research forward.

In conclusion, integration of multi-techniques and inspiration from math will be necessarily needed in the big data structure research, and we are looking forward to having proper resolution methods for big data.

REFERENCES

[1] Gerard George, Martine R. Haas, Alex Pentland, "BIG DATA AND MANAGEMENT," *Academy of Management Journal,* vol. 57, no. 321-326, 2014.

[2] S. Manghani, "The Art of Paolo Cirio: Exposing New Myths of Big Data Structures," *Theory, Culture & Society,* vol. 34, pp. 7-8, 2017.

[3] Sutikno T, Stiawan D, Subroto, "Fortifying Big Data infrastructures to Face Security and Privacy," *TELKOMNIKA,* vol. 12, pp. 751-752, 2014.

[4] B. Alam, "Matrix Multiplication using r-Train Data Structure," *2013 AASRI Conference on Parallel and Distributed Computing Systems,* pp. 189-193, 2013.

[5] Inbal Yahav, Galit Shmueli, "A TREE-BASED APPROACH FOR ADDRESSING SELFS-SELECTION IN IMPACT STUDIES WITH BIG DATA," *MIS Quarterly,* vol. 40, pp. 19-848, 2016.

[6] Lakhmi Jain, Xindong Wu, R-tree : Theory and Applications, Springer Science+Business Media, 2006.

[7] Yingsheng Chen, Dianne Hall, "Reliability Assessment Model for Big Data Structure of internet things," *TELKOMNIKA,* vol. 14, pp. 363-368, 2016.

[8] Hari Singh , Seema Bawa, "A MapReduce-based scalable discovery and indexing of structured big data," *Future Generation Computer Systems,* vol. 73, pp. 32-43, 2017.

[9] Marcello Trovati , Richard Hill, Ashiq Anjum, Shao Ying Zhu, Lu Liu, Big-Data Analytics and cloud computing, Switzerland: Springer International, 2015.

[10] B. Marr, Big data : using smart big data, analytics and metrics to make better decisions and improve performance, JohnWiley & Sons Ltd, 2015.

[11] Shanyun Liu, Rui She, Pingyi Fan, Senior Member, "Differential Message Importance Measure: A New Approach to the Required Sampling Number in Big Data Structure Characterization," *IEEE,* 2018.

[12] Riya Ojha, Rakshit Singh, Aditya Singh, "Big data analysis: structuring of data," *2017 International Conference on Technical Advancements in Computers and Communications,* vol. 45, 2017.

[13] M. A. Qader, "I/O Optimization in Big Data Storage Systems," *UC Riverside Electronic Theses*

and Dissertations.