Journal of Applied Data Sciences
Vol. 1, No. 1, September 2020, pp. 11-19

Limitations of Big Data Partitions Technology

Nguyen Huyen Trang *

Management Information System, Dai Nam University, Vietnam

* corresponding author

(Received: July 15, 2020; Revised: July 21 2020; Accepted: August 29, 2020; Available online: September 1, 2020)

Abstract

Big data is defined as the amount of data that is needed by new technology and architecture so that it is possible to extract the large amount of data provided by the analysis process. Due to its enormous size it is increasingly difficult for perfect analysis using existing traditional techniques. This technology is a solution for several problems that require a distributed system for storage needs because a problem cannot be solved in one machine. Since Big Data has become the latest technology in a market that brings tremendous profits to business organizations, it becomes possible when there are specific challenges and problems and it will continue to expand. This article introduces big data technology, and explains its partition limitations.

Keywords: Big data, Partitions, technology, Data science

1. Introduction

The rapid development of the times makes various changes and needs of a technology increasingly difficult and full of challenges. High-speed data development makes it ever more difficult to handle, especially with very large amounts of data. A rapid increase in size compared to computing sources is the key challenge in managing large volumes of data.

Big Data technology produces important value from a data warehouse and that many nations already started significant projects based on this technology. We define "Big Data" as the amount of data that exceeds the ability of technology to store, manage and process it efficiently. One of the main features of big data is partitioning, which can answer this problem. But this is not a solution for all problems. Partition itself has some binding limitations so that not all data models or computational problems can be solved with big data technology.

Data partitioning is just a basic problem and was well researched in communities with databases. There are also several studies in MapReduce on data partitioning and locality optimization. In this paper, we will discuss the partitions themselves, the limitations they have, relation with the MapReduce method and the solutions to overcome them.

2. Related Works

Most people already know how important big data is, but various researchers still have different definitions of big data itself. In general, big data itself is known as a dataset that cannot be obtained, perceived, processed and managed by ordinary software / hardware tools. Because of this difference, technical practitioners, scientific and technological companies, data analysis, and research scholars have different definitions of big data.

Apache Hadoop in 2010 defined big data as "datasets which could not be captured, managed, and processed by general computers within an acceptable scope." An IDC report in 2011 defines big data as "big data technologies

describe a new generation of technologies and architectures, designed to economically extract from very large volumes of a wide variety of data, by enabling the high velocity capture, discovery, and analysis."

Big data is unique from traditionally stored data[1]. Typically speaking, data processed on the big data must be clean, registered and accurate. And the data should also match the structure of the storage which will be placed. Big data manages not only data stored in traditional storage but also data not suitable for storage. Here are some of the uses of big data in various sectors:

A. Storage in the IT Industry

The IT industry stores large amounts of data as Logs or Records to handle problems that are generally rare and require such Logs. However, data storage like this only happens a few weeks but in the end the data will be stored longer because of the value that is very important for the company. Traditional systems cannot handle data like this because of several things such as its size, raw or unprocessed, and semi-structured. Big data not only analyzes all available data to mark the point of failure but also increases the shelf life of logs.

B. Data Sensors

The extraordinary amount of data sensors also represents a major challenge for big data. Nearly all companies are now facing this vast volume of data but are using only a limited amount of data for analysis due to lack of storage capacity and analytical techniques. In addition, data sensors is categorized as moving data and silent data. Ultimately, for better business, both performance and benefit require vast volumes of data to be analyzed.

C. Risk Analysis

Modeling data for estimating risk is essential for financial institutions, so that a project can be carried out. Very large volumes of data are currently underused and need to be reintegrated into the model to more effectively assess the risk patterns.

D. Social Media

Big data is mostly used in social media. Continuing to pay attention and monitor what customers are thinking about their goods allows business companies get the consumer feedback. They can use this feedback to make decisions and achieve more value from their company.

Then what is Partitioning? imagine we have an encyclopedia, that must be a very big book. over time and the history of the book will certainly become even greater and even more difficult to use. so we have to separate it into several volumes. Data partitioning is a technique for separating data during processing from master data [2]. with this technique we are able to divide a large dataset into several sections with each size that can be processed or accessed. This will help us to improve the data quality, management and performance. partitioning is a type of data management requiring the data handler to manage the partitioning (or division, segregation, separation) of variables and data in a given data set under those variables. Who is involved in partitioning? Data handler itself is most often the experimenter, the one who is required to expose the subject to set a treatment and continuously measure and identify their response against the treatment.

2.1. Limitations of Big Data in Partitions

Big data is a solution for several problems that require a distributed system for storage or computing needs. because a problem can no longer be solved in one single node / host machine and the machine's ability can no longer be improved. Partitioning is one of the main frameworks of big data that can solve this problem. However, Partition

itself is not the solution to all problems in big data. Even the partition itself has some binding limitations so that not all data models or computing problems can be solved with this technology.

2.1.1 Inflexible Data Structure

The main problem with Partitions is the inflexible data structure. The data structure that we generally operate on big data technology is very representative of the effectiveness and efficiency of the processing that we will do later. To simplify the explanation of the problem we try to simplify using analogies and simple examples, Let's assume that we have transaction data from an e-commerce user with the following data structure stored in storage.

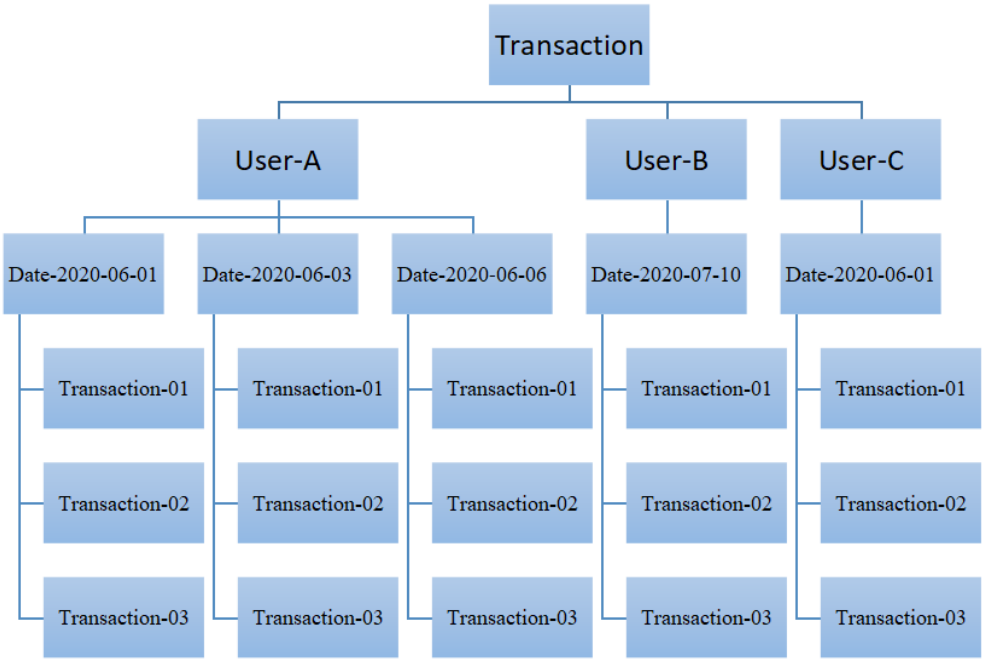

Fig. 3. Structure 01.

From this Example we can search for something easily when the partition is known, for example:

1. How many purchases of user B on 10-07-2020?

   We can do the aggregation of the sums to the partition / Transaction / User-B / Date-2020-07-10.

2. How many times does user A shop at the store?

   We can look at the partition / Transaction / user-A and count the number of transactions that have been made. Of course this requires a lot of resources or data if user A has conducted transactions for a long time and a lot so that the data in his partition is quite large.

But for some cases the data structure above is no longer relevant:

1. On 2020-06-01 how many users did a transaction?
2. What is the total number of transactions above two million?

The two cases above require us to track all partitions, which means we have to do a Full Scan Table, because we don't know the main data. This is not relevant because the initial mindset of using big data technology, our data is very large, hundreds of terabytes, petabytes, and even more than that. So doing a full scan table doesn't make sense to do. because it is very possible if it takes weeks or even months just to do simple operations like a full scan table. That is one example of the limitations of big data technology related to the features and principles of big data, namely the structure of data is very closely affecting the effectiveness and efficiency of an operation such as search, storage, and computing, depending on the technology we use.

2.1.3 Storage Consumption Swelling

The available capacity is not sufficient to store large and large quantities of data commonly generated by nearly everything, for example social media are the biggest contributors followed by sensor devices etc. Because of Big Data 's stringent network demands, storage and server outsourcing data into the cloud may seem like a solution. The problem isn't solved by storing massive quantities of data in this cloud. Since insights through big data requires all the collected data, and then unifies it by extracting essential information. The solution to the problem with the data structure described above is to use secondary or similar data structures as appropriate.

For example, let's assume the data structure that has been exemplified above is an example of a structure that has fulfilled the needs regarding transactions by customers (user transactions), we will create a new structure to meet the needs related to the time span (time based transaction) to answer question number 1 as follows.

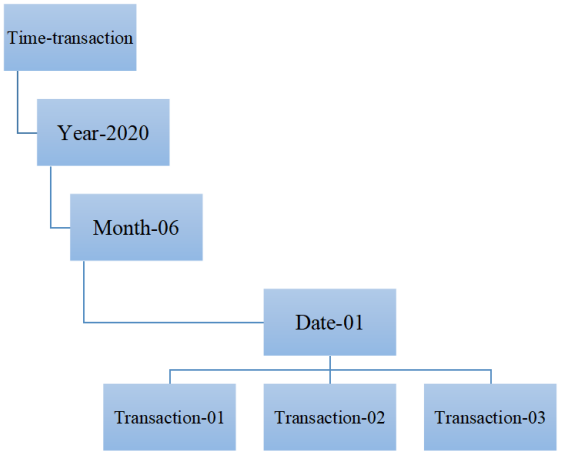

Fig. 4. Structure 02.

Then for question number 2, the following data can be made.

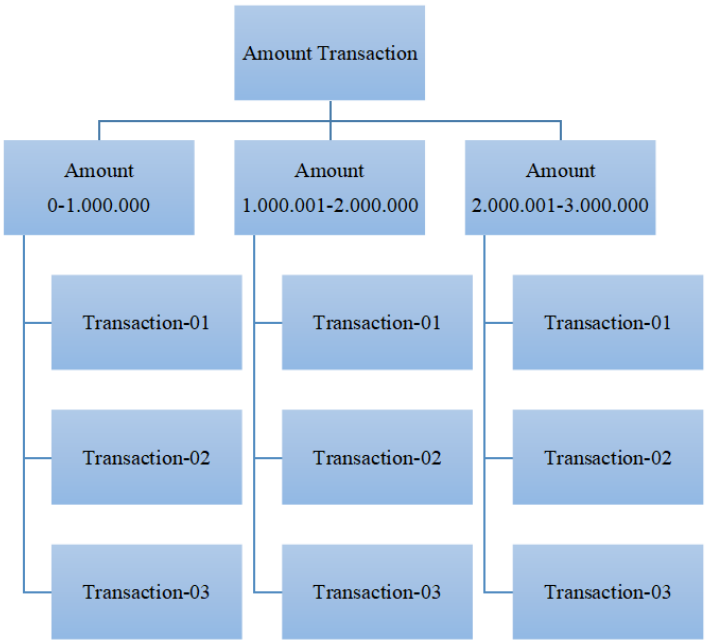

Fig. 5. Structure 03.

These two additional data structures make the data that needs to be stored will multiply at least 2-3 times from the start. In many cases swelling of data storage consumption can no longer be avoided, especially for big data technology that offers various storage solutions such as HDFS, HBase, Cassandra. Although this can be done with a variety of others, without the need to duplicate raw data for example by pre-computing, of course, with different advantages and disadvantages.

2.1.4 Inefficient Computational Process

This often happens especially for big data technology that offers distributed computation such as MapReduce and Apache Spark. Let's assume we do a process with a different approach to the same amount of data as the partition structure that we can exchange, for example the above is initially / user / date / to / date / user.

- The more partitions means the more processes that are waiting in line, this is at risk of causing Bottlenecks, if the number of nodes we set is too small, this will be an advantage if we have a large number of nodes.
- Conversely, if there are fewer partitions, which means that the amount of data will be more and more in one partition, then the number of queues will be less, but in one process will take up more and more resources, this is very suitable if the nodes we specify are smaller or computing capacity (RAM) in one node is high.

Several researchers do the work on handling large data set. The team of Tim Oates and David Jensen[3][4] proved that increasing the size of the training data did not greatly increase classifier accuracy. It was identified that as the number of training instances increases, the classifier's complexity also increases without a considerable increase in classification performance.

Hall et al[5] are presented by combining parallel study of the decision trees. The proposed algorithm constructs a decision tree with separate data subsets of a parallel collection of complete data, constructs the rule set and then

combines it into a single set of rules. Tests on the two data sets indicate that the quantity of rules created by the decision tree is increasing. Data partitions are used to divide data files, the reason for this is that the file is too large for a single disk, or because a single disk can not support the file access rate. Some of the available horizontal data partitioning techniques are round robin partitions, Range partitioning, and Hash partitioning[6]. Round robin is a simple partition strategy which uses round robin manner to divide instances in a data partition.

The hash partitioning technique selects one or more attributes that are applied to them from the data set as partition attributes and hash functions [7]. The function specifies the location of instances of data on a given partition. Hash function has a spectrum from zero to n-1. If the hash function returns i it places the data instance in the partition i^{th}. Applications requiring only sequential and associative access to information are applications that are appropriate for hash partitions.

Range clubs that partition data instances with similar data values together. The example is City = Tokyo, income > 25K. Range partitioning suffers from data skew problem. The skew problems are less vulnerable to hashing and round robin. The proposed method of round robin partitioning is the most suitable method algorithm, since it does not suffer from data skew.

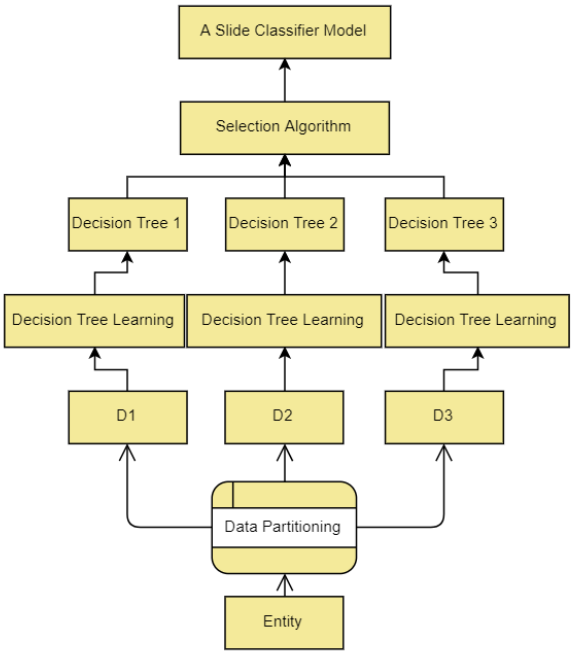

Fig. 1. Data mining algorithm proposed for three partitions of the data.

3. Partition Method

While sampling and size reduction methods used in single-machine partitioning algorithms represented improve the scalability and speed of the algorithms [8], data size growth is nowadays much faster than memory and processor progress. Therefore, a single-processor and memory machine can not manage terabytes and petabytes of data and demonstrates the need for algorithms that can be performed on multiple machines. This technique allows the massive amount of data to be split into smaller pieces that can be loaded on multiple machines, and then uses the processing power of these machines to solve the enormous problem.

3.1. MapReduce Multi Machine Partition Algorithms

MapReduce is a framework that is shown below, The open source version of this is initially represented by Google and Hadoop[9]. This section evaluates the algorithms implemented on the basis of this framework and discusses their improvements in terms of three features:

- **Speed up:** Means runtime ratio while the dataset is constant and the number of machines in the system is increased.
- **Scale up:** Indicator when x time larger system with the same runtime will do x time larger job
- **Size up:** Maintaining unchanged machine numbers, runtime increases linearly with data size

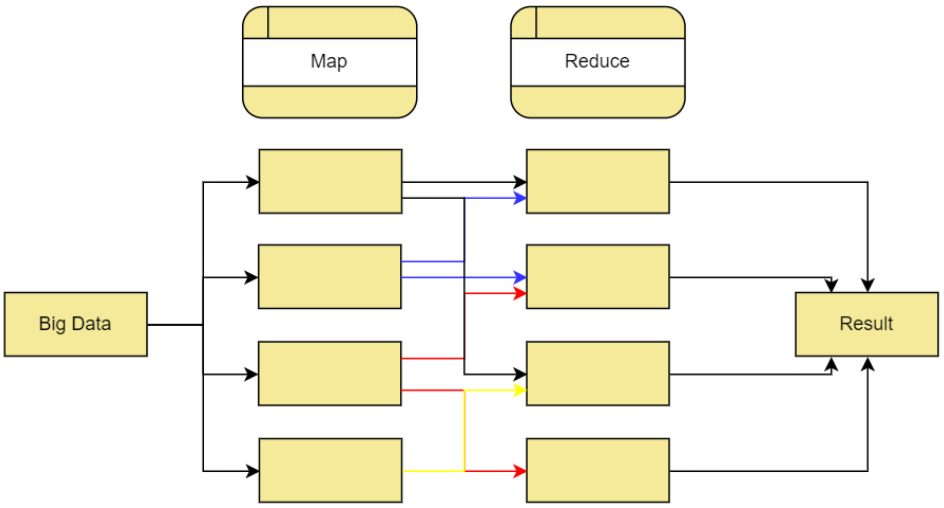

Fig. 2. MapReduce Framework

K-means based at MapReduce (PK-means)

PK-Means[10] is distributed version of the well known K-means partition algorithm[11][12]. The aim of the k-means algorithm is to partition desire data set into k clusters in such a way that instances in one cluster display more similarities than most other cluster instances. K-means partition randomly pick k instance of data set in the initial stage and repeatedly execute two steps: First, it assign each instance to the closest cluster and updates the centers for each cluster with the mean of the instance after completing the assignment for all instances in the second step.

PK-Means uses MapReduce framework to spread computation between several machines to speed up and scale the operation. Person clustering in the mapper that comprises the first step and then general clustering in the reducer. PK-Means has almost linear speed and linear scale up, too. It's got a good size up too. This represented a scale of 0.75 for 4 computers. PK-Means, on the other hand, is an exact algorithm , meaning it gives the same level of clustering as its serial counterpart k-means.

MR-DBSCAN

A very recent suggested algorithm is MR-DBSCAN[13] which is a scalable DBSCAN algorithm built on MapReduce. In parallel DBSCAN algorithms which MR-DBSCAN fulfills, there are three major drawbacks: First, they are unsuccessful in balancing the load between parallel nodes, secondly, these algorithms are limited in scalability because not all critical sub-procedures are parallel and finally their architecture and design limits them to less portability to emerging parallel processing paradigms.

MR-DBSCAN proposes a novel data partitioning method based on the emission of computation costs and a scalable DBSCAN algorithm in which all critical sub-procedures are fully parallel. Big dataset tests affirm the MR-DBSCAN scalability and performance.

4. Solution

Based on a paper made by Rajeev Agrawal & Christopher Nyamful [14] there are a number of suggested solutions to reduce the risk of problems with partial and big data processing.

- Preparing for Large Storage Media

  In storing a data it is advisable to have a device that has the ability to measure access, access time, transfer speed and effectiveness. So, we are advised to use Hard disk drive (HDD) or Solid state drive (SSD).

- Backup Strategy

  Recovery is the main goal of the backup strategy. Which is where if something unexpected happens, the system can be restored again. Full Backup guarantees full recovery speed since creating a backup data set takes a lot of time due to its large size. Data deduplication technology continually decreases the volume of data blocks stored for each backup, allowing users to make backups and recover data in a relatively short time. Backups are generally done from a replication system in an efficient storage system, rather than directly from a production system. Replication will store copies directly and in real time of the production data.

5. Conclusion

Partitioning is a basic and major feature of big data technology related to the physical location where data processing and location are stored. This is a solution of several scalability problems so that it can provide a scale out solution. However, the partition itself has several limitations and attachments that need to be considered before use, because the data structure that represents the partition is very influential on the effectiveness and efficiency of an operation on big data technology both storage, computing, and various other processing. It is likely that the expenditure we have invested in big data technology has no effect if the design of our data structure is incorrect.

References

[1] Katal, Avita, Mohammad Wazid, and Rayan H. Goudar. "Big data: issues, challenges, tools and good practices." *2013 Sixth international conference on contemporary computing (IC3)*. IEEE, 2013.

[2] Marz, Nathan, and James Warren. *Big Data: Principles and best practices of scalable real-time data systems*. New York; Manning Publications Co., 2015.

[3] Oates, Tim, and David Jensen. "The e! ects of training set size on decision tree complexity." *Proc. 14th Int. Conf. on Machine Learning*. 1997.

[4] Oates, Tim, and David D. Jensen. "Large Datasets Lead to Overly Complex Models: An Explanation and a Solution." *KDD*. 1998.

[5] Hall, Lawrence O., Nitesh Chawla, and Kevin W. Bowyer. "Combining decision trees learned in parallel." *Working Notes of the KDD-97 Workshop on Distributed Data Mining*. 1998.

[6] Ray, Chhanda. *Distributed database systems*. Pearson Education India, 2008.

[7] Patil, Dipak V., and R. S. Bichkar. "An optimistic data mining approach for handling large data set using data partitioning technique." *International Journal of Computer Applications* 24.3 (2011): 29-33.

[8] Shirkhorshidi, Ali Seyed, et al. "Big data clustering: a review." *International conference on computational science and its applications*. Springer, Cham, 2014.

[9] Anchalia, Prajesh P., Anjan K. Koundinya, and N. K. Srinath. "MapReduce design of K-means clustering algorithm." *2013 International Conference on Information Science and Applications (ICISA)*. IEEE, 2013.

[10] Zhao, Weizhong, Huifang Ma, and Qing He. "Parallel k-means clustering based on mapreduce." *IEEE international conference on cloud computing*. Springer, Berlin, Heidelberg, 2009.

[11] Jaiwei, Han, and Micheline Kamber. "Data mining: concepts and techniques." *ed: Morgan Kaufmann San Francisco* (2006).

[12] Mirkin, Boris. *Clustering: a data recovery approach*. CRC Press, 2012.

[13] He, Yaobin, et al. "MR-DBSCAN: a scalable MapReduce-based DBSCAN algorithm for heavily skewed data." *Frontiers of Computer Science* 8.1 (2014): 83-99.

[14] Agrawal, Rajeev, and Christopher Nyamful. "Challenges of big data storage and management." *Global Journal of Information Technology: Emerging Technologies* 6.1 (2016): 1-10.