2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)
Amity University, Noida, India. June 4-5, 2020

Role of Bloom Filter in Analysis of Big Data

Sudhriti Sengupta
AIIT, Amity University
Noida , Sector 125
Uttar Pradesh
ssgupta@amity.edu

Ajay Rana
AIIT, Amity University Uttar Pradesh
Noida, India
ajay_rana@amity.edu

***Abstract*- Big data is a collection of large amount of data which increases in volume, velocity and variety very rapidly. As a researcher, deriving values of importance from this large repository of data is utmost important and challenging. This paper discusses the methods for using Probabilistic Data Structure in Big Data Analysis. The application is primarily focused on Bloom Filters as processing of big data is a major challenge because big data is a continuous stream of rapidly increasing data. To have maximum benefit from data, a bloom Filter can be used so that usability of big data can be achieved while decreasing space or time.**

Keyword: Big Data Analysis, Bloom Filter, Hash Function, Probabilistic Data Structure

I. INTRODUCTION

Big Data relates to a large amount of heterogeneous or homogenous data from multiple sources. The largeness of Big Data correlates with large data storage, efficient data collection and networking of devices. Presently, data has become fuel for many research, production and business for generation of services or products such that it is often considered as fourth factor in production, along with other factors such as capital, human resources and machinery [1].There are many challenging issues regarding utilization of Big Data. Researchers involving in Big data have many challenges regarding accumulation, storage and analysis of Big Data so t involved in hat the enterprise or business using Big Data can have maximize benefit using minimize resource involved in Big Data. The life cycle of Big Data [2][3][4] have the following stages,

A. Data Generation

Data is generated from various sources such as social networking sites, mobile device, networked device, sensors to name a few.

B. Acquisition

This step consists of three stages: Collection, transmission and pre-processing. After acquisition the data is filtered and ready to be stored.

C. Storage

Secure, large storage capacity having fast access is required. Some examples are flash storage, DAS attached storage and hyper-scale computing environment. Hyper-scale computing environment is used by google and facebook user.

D. Analysis

Analysis means extracting of critical piece of from large horde of data by applying variety of techniques.

During data acquisition phase data is available from many sources [2]. Illustrations can be of Data Set used by CERN is 13-15 petabytes in 2010, Social Media site twitter has 12+ TBytes of tweets growing everyday [5]. To derive some values from these enormous data researchers are focussing on the various stages of Big Data.

The conventional data structures and algorithm are not sufficient to manage larger and more complex dataset of Big Data.

In this paper, we have produced a brief overview of an advanced technology called Probabilistic Data Structure and Algorithm. We have discussed the common Probabilistic Data Structure and their implementation in Big Data Analysis. These algorithms are basically based on a variety of hashing techniques. They provide approximate results as compared to conventional data structure. Probabilistic Data Structure are used to optimize the fixed or linear memory. There execution time is constant in general. Some of the popular Probabilistic Data Structure are Bloom Filter, Hyper Log Log, q-Digest, Count-Min-Sketch, SimHash etc. Many Organisation and applications uses these techniques such as Google Big-query, Redshift by Amazon, Cassandra by Apache to name a few.

Section 2 discusses about Probabilistic Data Structure and its types. Section 3 elaborates about one of the common Probabilistic Data Structure and Algorithm called Bloom Filter and explain its role in Big Data processing and storage. This paper aims to provide a concise study of Probabilistic Data Structure, its types and implementation in Big Data Analysis and Processing.

II. PROBABILISTIC DATA STRUCTURE AND ALGORITHM

Conventional Data Structures are not suitable for analysis of the huge capacity of Big Data because the computational and time complexity is large. Probabilistic Data Structure is more efficient in its involvement of constant factor in actual run time. Thus they are suitable for Big Data processing.

One of the common task in data processing is classifying the data by using check conditions, determining the occurrence and number of unique elements, frequency of elements etc.

Hash Tables and Hash sets are most widely used for this purpose. But these data structures are not suitable when dealing with enormous velocity, volume and variety of Big Data. For dealing with these data, Probabilistic Data Structure are proposed and found to be beneficial in big data processing and analysis. These techniques are based on hash functions to represent a set of elements randomly. They provide answer approximately. However, reliable sources to estimate error is also given. They use much less memory and have constant query time. They can be paralleled and supports union and intersection operations. All these make Probabilistic Data Structure suitable for

Big Data Analysis and Processing. They are used in all the aspects of Big Data, viz., Volume to check the membership, Velocity to find frequency and rank and Variety to check similarity.

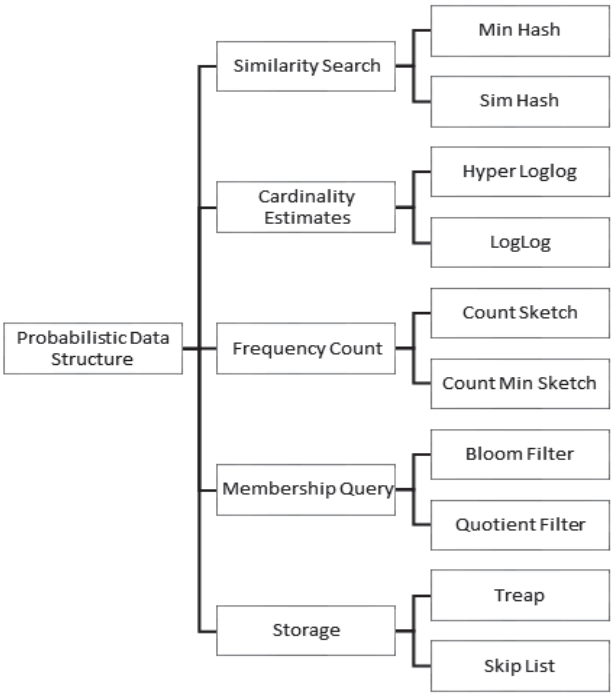

Fig. 1. Some Common Probabilistic Data Structure and their applications

In the figure 1 we see that Bloom filter and Quotient Filters are used to check membership, i.e. if any elements belong to a given set. Count Min Sketch and Count Sketch are used to calculate frequency of data elements occurring in some datasets. To find the similarity or form clusters, SimHash and Minhash are used. Random Sampling, q-digest and t-digest are used to rank the elements in appropriate order. Other popular algorithms are Linear Counting, LogLog and HyperLogLog which are used for counting purpose. There are many probabilistics Data Structure and Algorithm which are suitable for its application in Big Data Analysis. In the subsequent section we have discussed about the working and impact of Bloom Filter in analysis of Big Data.

III. BLOOM FILTER

A Bloom Filter is a probabilistic data structure to quickly test whether data sets belong to a larger set by using multiple has functions [10] [11]. Bloom Filtering Technique is used to test whether an element is a member of a set. It returns two types of result that can be defined in false positive or false negative. It is used to check membership of an element e in a set of S elements. The output of Bloom Filter is either element e might be present in S or it is not at all present in a particular data set. The fundamental concept of Bloom Filter is to use multiple hash function to determine whether an element belongs to a particular data set. Bloom filter essentially consists of a data set, generally consisting of 0 or 1and some hash functions. Hash functions have two applications in Bloom Filter: adding new Positives and to check membership of data. The number of hash function to be used depends on the required accuracy of bloom filter. Other factors impacting the accuracy of bloom filter are size of the data set and number of elements added to the set.

A basic Bloom filter works in two steps:

1. Adding positive to the data set.
2. Testing for member ship in set

The data structure used in Bloom filter consists of an array of n bits where every bit is initialized zero. Membership of any element into the bloom filter is done by using hash functions. If the result of hash function is positive, then set corresponding bit to 1 else no change is done. In this way adding positives to the Bloom filter is done. The number of hash functions to be used depends on the required accuracy of the Bloom filter. In Bloom filtering the inputs given to the hash functions to add positives should be relevant.

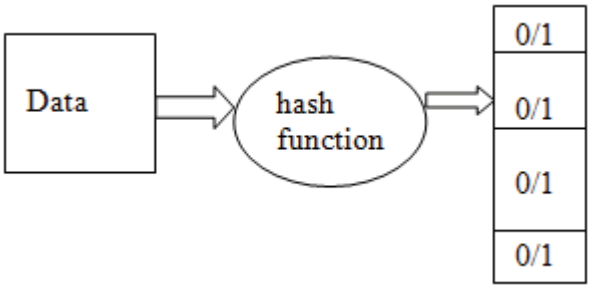

Fig. 2. Adding element in the Bloom Filter.

The data might be taken after initial some amount analysis, that means to build the Bloom filter, some initial level of analysis is required. By using the result of this analysis and some hash functions the bit values in the filter will be set to positive or negative.

After adding positives to the data set, the Bloom filter is ready to check the relevance of the input data by using the same process but in a reverse way. The hash functions are applied to the input data and if the corresponding bit is 1 than the input is a suitable for further processing else it is discarded

To build Bloom filter for filtering e first we must find out a reliable hash function h(x). By using h(x) we set positives to the bit array used in data structure. We might use more than one has function depending on the desired level of analysis. After building the data set in the Bloom filter, all the data are filtered by checking whether the bit is position or negative in the bit array.

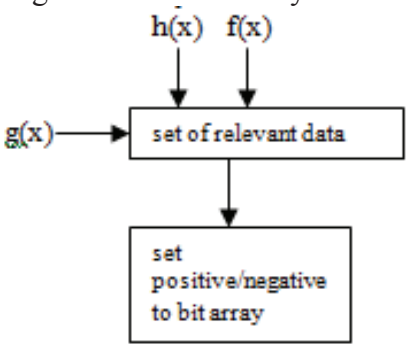

Fig. 3. Querying in Bloom filter.

In Figure 3 we have used three hash function h(x),f(x) and g(x) which can be used in filtering in Big Data. It can be illustrated by figure 4.

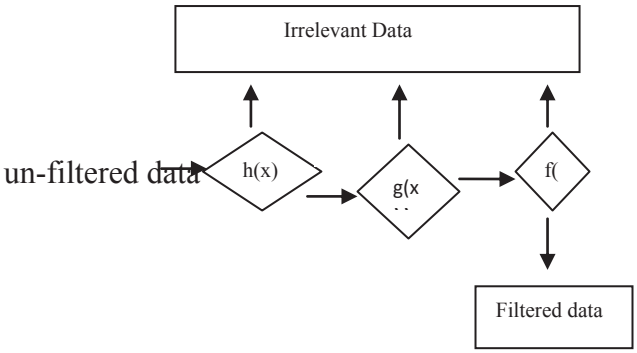

Fig. 4. Filtering in Big Data

Increase in the number of hash function, lowers false positive rate but it will increase in programming overload in insertion and checking. Increase in size of bit array will also lower false positive rate but it will increase the space needed.

There are four main categories of Bloom Filter.

- Static Bloom Filter uses fixed size array.
- Counting Bloom Filter uses counter, in lieu of array.
- Incremental Bloom Filter uses dynamic array which keeps on changing depending upon the number of elements.
- Ageing Bloom Filter uses a cache system based on Least Recently Used Algorithm.

Some of the applications of Bloom Filter in Big Data analysis is discussed as follows.

Bloom Filters was first used by Burton H. Bloom in 1970 to search in dictionaries and databases [13]. In 2000, Fat et al. used counting Bloom Filters to support frequency query in web caches[14].Cohen and Matias used Special Bloom Filter to filter minimum count of an element in multisets[15]. In 2003, Kumar et al. used Bloom Filter to measure traffic flow in Blind streaming [16]. Deng et al. used bloom filter to detect redundancy in stream data[17].Almeida et al. used bloom filter to apply membership query in streaming of Big Data[18]. In 2010, Goel and Gupta used layered Bloom Filter for supporting Big Data frequency query [19]. Hua and Li used Bloom Filter to approximately test membership in big data with disruptions [20].

IV. Discussion

Big Data is heterogeneous, voluminous and sometimes incomplete and inconsistent. For efficient and effective use of data, its processing must be reliable and automated. One of the main issues is storing, indexing and categorizing massive data. This paper discusses Probabilistic Data Structure and one of its type called Bloom Filter, which helps to achieve constant space and time complexity in view of massive streaming of data. Bloom Filter is simple and adaptive in nature so it is used in a large number of applications. Bloom Filter supports different operations like hashing, membership query etc. Time and Space complexity of Bloom Filter is low. Also computational cost of Bloom Filter is minimum.

V. Conclusion

The volume, variety, velocity and veracity makes analysis and processing of Big Data by conventional Data Structure difficult as it involves large space, time and computational complexities. To overcome, this issue, Probabilistic Data Structure and Algorithms are used in Big Data analysis and processing. These give approximate results but the error margin can also be calculated. Many types of Probabilistic Data Structure and Algorithms are present which are involved in different steps of Big Data Processing like membership query, search, count, storage etc. Membership query is used for identifying elements belonging to certain categories. Bloom filter, Quotient filters etc. are common Probabilistic Data Structure used in this field. In this paper, we have discussed the role, implementation techniques and some applications of Bloom Filters in Big Data Analysis and Processing. This will help the research to have a basis overview of Probabilistic Data Structure with emphasis of Bloom Filter in Big Data analysis.

References

[1] A. Schmidt, M. Atzmueller and M. Hollender," Data Preparation for Big Data Analytics", Enterprise Big Data Engineering, Analytics, and Management,vol. 10,pp-225-239,2016.

[2] I. Taleb,R Dssouli and A.Mohamed , "Big Data Pre-processing: A Quality Framework", IEEE International Congress on Big Data, Vol. 10,pp. 191 – 198, 2015

[3] M. Chen, S. Mao, and Y. Liu,"Big Data: A Survey", Mobile Network Application,Vol.19,pp-171-209, 2014.

[4] H. Hu, Y. Wen, T.-S. Chua, and X. Li, "Toward Scalable Systems for Big Data Analytics: A Technology Tutorial", IEEE Access,vol. 2, pp. 652-687, 2014.

[5] S. Kaisler, F. Armour, J. Espinos and W. Money, "Big Data: Issues and Challenges Moving Forward", Hawaii International Conference on System Sciences .Vol.1, pp. 995-1004,2013.

[6] Tsai, Chun-Wei.Lai, Chin-FengChao, Han-Chieh Vasilakos, Athanasios V, "Big data analytics: a survey", Journal of Big Data vol. 6,2015.

[7] S. K. Bansal, "Towards a Semantic Extract-Transform- Load (ETL) Framework for Big Data Integration", IEEE International Congress on Big Data (BigData Congress),vol.1 pp. 522-529. 2014.

[8] X. L. Dong and D. Srivastava, "Big data Integration", International Conference on Data Engineering (ICDE),vol.29,pp. 1245-1248, 2013,

[9] G.-Z. Yang, J. Andreu-Perez, X. Hu, and S. Thiemjarus, Multi-sensor Fusion, Body Sensor Networks, Information Fusion, vol.35,pp-68-80, 2014.

[10] Chen, Z H E, Bayesian Filtering: From Kalman Filters to Particle Filters, and Beyond, Statistics: A Journal of Theoretical and Applied Statistics ,vol.1,pp.182-191,2003.

[11] Allenby, Greg M.Bradlow, Eric T.George, Edward I.Liechty, John McCulloch, Robert E, Perspectives on Bayesian Methods and Big Data, Customer Needs and Solutions,2014.

[12] S.Tarkoma,C.Rothenberg,E. Lagerspetz, Theory and Practice of Bloom Filters for Distributed Systems, IEEE Communications Surveys & Tutorials, vol. 14, no. 1, pp. 131-155.

[13] Bloom B.H.Space/time trade-offs in hash coding with allowable errors Commun. ACM, vol. 13,issue 7, pp. 422-426,1970.

[14] LFan, Cao P., Almeida J., Broder "A.Z.Summary Cache: A scalable wide-area web Cache sharing protocol", *EEE/ACM Transaction. Netw*ork,vol. 8 ,issue 3, pp. 281-293,2000.

[15] S Cohen and Y Matias,"Spectral bloom filters",Proceedings of the ACM SIGMOD International Conference on Management of Data), vol. 3, pp. 241-252,2003

[16] A. Kumar, J. Xu and J. Wang, "Space-Code Bloom Filter for Efficient Per-Flow Traffic Measurement," in *IEEE Journal on Selected Areas in*

Communications, vol. 24, no. 12, pp. 2327-2339, Dec. 2006. doi: 10.1109/JSAC.2006.884032

[17] F Deng and D Rafiei," Approximately detecting duplicates for streaming data using stable bloom filters",Proceedings of the ACM SIGMOD International Conference on Management of Data, ,vol.1, pp. 25-36,2006

[18] P Almeida, C Baquero ,N Preguiça and D Hutchison,".Scalable bloom filters"Information. Process Letter, vol. 101 ,issue 6, pp. 255-261,2007

[19] A Goel and P Gupta,".Small subset queries and bloom filters using ternary associative memories, with applications" SIGMETRICS Performance Evaluation Review, vol., 38,issue 1, pp. 143-154,2010.

[20] Y. Hua, B. Xiao, B. Veeravalli and D. Feng, "Locality-Sensitive Bloom Filter for Approximate Membership Query," in *IEEE Transactions on Computers*, vol. 61, no. 6, pp. 817-830, June 2012