ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 1, January 2018

A Study on Big Data with Indexing Technique for Searching and Retrieval of Data Fastly

Pooja Anand[1], Dr. Sandeep Maan[2]

Research Scholar, Dept. of Computer Science and Application, Singhania University , Jhunjhunu, Rajasthan, India[1]

Associate Professor, Dept. of Computer Science and Application, Govt. College for Women, Sec.-14, Gurugram, India [2]

ABSTRACT: The rapid growth in volume, velocity, and diversity of data produced by mobile devices and cloud applications has played a major role in collection of relevance or irrelevance data or 'big data'. Available solutions for efficient data storage and management cannot fulfill the needs of such various data where the amount of data is continuously increasing day by day. For efficient retrieval and management, existing indexing techniques become inefficient with the rapidly growing index size and seek time and an optimized index scheme is required for big data. Regarding real-world applications, the indexing issue with big data in cloud computing is widespread in healthcare, enterprises, scientific experiments, and social networks. The objective of this paper is to examine and their usefulness of some indexing techniques for big data.

KEYWORDS: Indexing, big data, search key, search value.

I. INTRODUCTION

Data comes from various sources which make difficult to match, link, cleanse .At global level the amount of data stored is almost inconceivable and it is growing day by day yet only small percentage of data is analyzed actually. So how a organization can do for better use of raw information that flow at all level of organization.

Big data is that data in which the large volume of data either structured or unstructured – that help a business to fulfill their day-to-day activity. How many data is available for you but the thing is that how you can use this. But in this the amount of data is very high that important for an organization. It's depend on organizations do with the data that matters. Big data can be analyzed that by this an analyst take better decisions and business strategies to fulfill .

Volume

Organizations collect data from various sources such that business transactions, social media and information from sender or machine-to-machine like system to system . In the past, storage of data would create a big problem – but now a day's new technologies are available (such as Hadoop) to solve the burden of a business.

Velocity

Data in an unparalleled speed and must be deal with in a time. RFID tags, sensors and smart metering are driving the need to deal with torrents of data in near-real time.

Variety

All types of data formats available like text, audio, document, financial transaction etc..

Variability

As increasing the velocities and varieties of data, data flows can be highly inconsistent with periodic peaks. sometime trending in social media? Daily, seasonal and event-triggered data creates challenging to manage all data. Even unstructured data type also create issues.

DOI: 10.15680/IJIRCCE.2018. 0601008

II. HOW INDEXING IS HELPFUL ?

Indexing technique is used to retrieve data from the stored database files which is based on some attributes, entity on which the indexing has been done. Indexing in the database systems is similar to that data which we seen in books. Indexing is defined based on its indexing attributes.
Indexing is a way to optimize performance of a database by minimizing the number of disk accesses required when a query is processed.
An index or database index is a data structure which is used to quickly locate and access the data in a database table.
Here two types of index that are available to speed up full text search. These index are not must to use for this type of searching but mostly we use this where column is searched or index is desirable.

Custom indexing supports multiple field indexing based on arbitrary or user defined indices . They are usually based on indexing strategies such as B-tree, R-tree, inverted index, and hash indexing strategy. Two types of custom indexing strategies are Generalized Search Tree (GiST) and Generalized Inverted Index (GIN) . GIST or GIN are two index . There are considerable performance differences between the two index types, so it is important to understand their characteristics first .

- When you create index B-Tree is automatically inserted. All database are included B-Tree.The B stands for Balanced Tree and in this the amount of data on both sides left or right of the tree is roughly the same. So the number of traversed for search to find rows is always in the same. For equality or range queries B-Tree is efficiently worked. They can operate on all datatypes, and null values can also be retrieve.

- For equality comparison Hash Index are used, but some time you never want to use them since they are not transaction safe, after crashes need to be manually rebuilt so the advantage over using a B-Tree is rather small.

- Generalized Inverted Index(GIN) are useful when for one row many values to be map in an index. When a row has single key value B-Tree is used. GINs are better for indexing array values as well as full-text search implementing.

- Generalized Search Tree (GiST) indexes used for equality and range comparison operations. The geometric data types or full-text searchof an index is included in GiST.

1)GiST: The Generalized Search Tree or GiST indexing strategy, is an indexing strategy based on the B-tree or theR-tree. It allows for the creation of custom or arbitrary fields as indexes. The GiST has the same implementation (for indexing and retrieval) as the R-tree for those based on the Rtree and as the B-tree for those based on the B-tree. Hence, they support indexing and query on one-dimensional data, as well as multi-dimensional or spatial data. Leaf node ,root node, pointer and other characteristics are included in GiSt. Despite these similarities between the Gist and the tree based strategies, the former has an advantage of supporting ad-hoc queries over the latter.

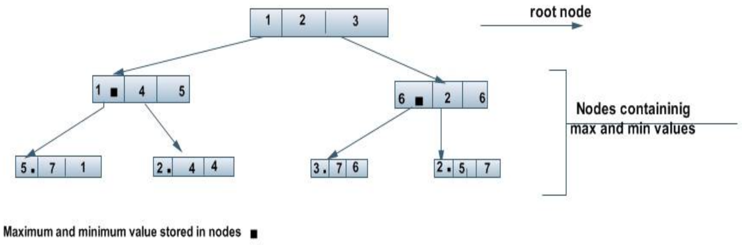
1 2 3
root node
1 ■ 4 5
6 ■ 2 6
Nodes containinig max and min values
5. 7 1
2. 4 4
3. 7 6
2. 5 7
Maximum and minimum value stored in nodes ■

Fig 1:gist

DOI: 10.15680/IJIRCCE.2018. 0601008

2) GIN: In GiST, the Generalized Inverted Indexor GIN indexing strategy (or access method) uses custom fields as index. It is mainly designed for fulfill specific users requirements. Though the GIN is implemented like the B-tree and has properties of the inverted index but GIN has differs from the B-tree which are based on predefined comparison-based operations. In this like a B-tree index which comprises of Entries Tree or list tree and Posting Tree or Posting List . In the ET, each entry represents an element ofthe searched key or indexed value, for example, arrays. The PL is a pointer to a list of items, or a pointer to a B-tree (for leaf nodes), in which case it is called a PT.While the B-tree is good for single-match indexes or range queries, the GIN works best for indexes having many duplicates. This is because the GIN queries data only by point or equality matching. This is often viewed as a limitation.

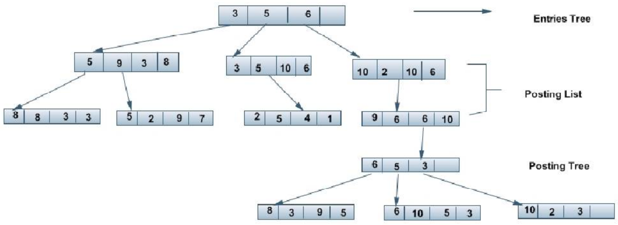

Fig 2:gin

To find which index to use gist or gin, follow these performance differences:

- Gin index is three time faster in searching than gist.
- Gin is three times longer to build than gist.
- Gin is larger than gist.

GIN indexes are best for static data because searching is faster. Gist index are faster to update for dynamic data. GiST indexes are good for dynamic data and fast if the number of unique words is under 10,00,000, while GIN indexes will handle 10,00,000+ words better but updating is slowly.

III. CONCLUSION

This paper include popular data indexing approaches for Big Data processing and management. The objective is to find how indexing techniques is suitable with big data. In how much time we retrieve data from various data blocks and how they are utilized for solving Big data management issues. The paper concludes that these techniques are helpful in full text searching, but in case where a column is searched on a regular basis. They have predefined operations also and how much data is used for searching or retrieve data.

REFERENCES

1.Gärtner M, Rauber A, Berger H (2013) Bridging structured and unstructured data via hybrid semantic search and interactive ontology-enhanced query formulation.
2. Demirkan H, Delen D (2013) Leveraging the capabilities of service-oriented decision support systems: putting analytics and big data in cloud.
3. Amer-Yahia S, Doan A, Kleinberg J, Koudas N, Franklin M (2010) Crowds, clouds, and algorithms: exploring the human side of "big data" applications. Paper presented at the proceedings of the 2010 ACM SIGMOD international conference on management of data, Indianapolis, Indiana, USA
4. Dixon Z, Moxley J (2013) Everything is illuminated: what big data can tell us about teacher commentary.
5. Liu W, Peng S, DuW,WangW, Zeng GS (2014) Security-aware intermediate data placement strategy in scientific cloud workflows. 6. Dopazo J (2013) Genomics and transcriptomics in drug discovery. Drug Discov Today 19(2):126–132.
7. Wang J, Wu S, Gao H, Li J, Ooi BC (2010) Indexing multi-dimensional data in a cloud system.
8. Fiore S,D'AncaA, Palazzo C, Foster I,Williams DN,loisioG(2013) Ophidia: toward big data analytics for science.

9. Chen J, Chen Y, Du X, Li C, Lu J, Zhao S, Zhou X (2013) Big data challenge: a data management perspective.
10. Wang M, Holub V, Murphy J, O'Sullivan P (2013) High volumes of event stream indexing and efficient multi-keyword searching for cloud monitoring. Future Gener Comput Syst 29(8):1943–1962
11. Rodríguez-García MÁ, Valencia-García R, García-Sánchez F, Samper-Zapater JJ (2013) Creating a semantically-enhanced cloud services environment through ontology evolution.
12. Cambazoglu BB, Kayaaslan E, Jonassen S, Aykanat C (2013) A term-based inverted index partitioning model for efficient distributed query processing.
13. Bast H, CelikikM(2013) Efficient fuzzy search in large text collections.
14. Paul A, Chen B-W, Bharanitharan K, Wang J-F (2013) Video search and indexing with reinforcement agent for interactive multimedia services.