ISSN:0975-9646

An Efficient and Compact Indexing Scheme for Large-Scale Data Store

S.Sageengrana[#1] , P.Ponnaruvi [#2], G.Tamilmani [#3,]K.Rajathi[#4]

Department of Computer Science and Engineering
[#1,#2,#3]Assistant professor,
[#1,#2]Vel Tech High Tech Dr.RangarajanDr.Sakunthala Engineering College
[#3, #4]Vel Tech Technical University
No.60 Avadi-VelTech Road,Chennai-600 062, TamilNadu,India

Abstract-E-Learning aims at replacing old-fashioned pre-determined learning with an on demand process of learning. The learning objects for the E-Learning process were used as a reusable component, which is interoperable to make the learning environment easy, and make the education at content available in any format that user needs. To have an effective courseware presentation, the (Multilevel Bitmap index with Bloom filter) Multilevel-BIBF was used to quickly provide learners with the correct learning content. Bloom filter (BF) was used to Multilevel Bitmap indices into access efficient. The main objective of this work is focused towards providing an efficient E-learning environment by assembling the Learning Objects (LO) dynamically for a given user query with minimum access time supported by Multilevel Bitmap index with Bloom filter. Hence, we propose a novel Multilevel-BIBF-HT (Multilevel Bitmap index with Bloom Filter and Hash table) scheme, which can achieve precise query results at the cost of slightly more space and time overheads. Our experimental results confirm the effectiveness, efficiency and scalability of the indexing scheme.

1. INTRODUCTION

E-Learning provides effective, task relevant and just-in-time learning for the new, dynamically changing, distributed business world. The use of network technologies for E-learning contents facilitates learning anytime and anywhere, and it also acts as a force for people and organizations to have a competitive edge in the rapidly changing global economy. The learning content is the heart of the education; however, it is locked into certain tools and platforms, which make it impossible to reuse them new platforms. In the last decade, learning objects, which are smaller chunks of learning content, have gained a lot of interest as the basis of a new type of computer-based instruction, in which the instructional content is created from individual components.

The learning objects are a collection of content items, practice items, and assessment items, that are combined based on a single learning objective, and which that enable and facilitate the use of educational content online. The concept of the learning object has evolved from the need to reuse digital learning materials. There are plenty of famous E-Learning standards launched to support the E-Learning environment, such as SCORM (Sharable Content Object Reference Model), IEEE LTSC LOM (Learning Object Metadata), and IMS Common Cartridge. SCROM standard used to create learning object of the above standard.The learning objects are created according to the SCORM standards. SCORM refers to a Sharable Content Object Reference Model. The SCORM is a collection and harmonization of specifications and standards that defines the interrelationship of content objects, data models and protocols such that objects are sharable across systems, thus promoting reusability and interoperability of learning content across Learning Management Systems. Efficiently handle the huge amount of E-learning resources during storage and retrieval time by using Bitmap index.

A bitmap index is a special kind of indices that stores the bulk of its E-Content as bit array format and answers the queries by performing bitwise logical operations on these bitmaps. In a Bitmap Index, a Bitmap for each key value is used instead of a list of rowdies. In Bitmap structures, a two dimensional array is created with one column for each row in the table being indexed. Each column represents a distinct value within the Bitmapped index. This two dimensional array represents each value of the index, multiplied by the number of rows in the table. If increase the size of an index tables its leads performances delay in access time. So Bloom filter is used to shrink the index table size.

Bloom filter was first proposed by B.Bloom in 1970 and were widely used in many applications in databases and networking, which can yield an extremely compact data structure that supports membership queries to a set. BF and the hash table are used to verification values of items, respectively. .

Encoding methodology used to reduce the size of Bitmap index table but it's very difficult to compress and decompress the huge content. To avoid this problem we used Bloom filter with multiple hash tables. This technique is effective for queries that ask for a small and huge volume of data with query criteria based on row identifiers. For example, consider an E-Content where the data is physically ordered by date. A query that asks for the total Learner (visitor) of E-Learning resources in every Monday for the last three months would effectively select only twelve rows by using Multilevel Bitmap indexing schemes. In other bitmap indices, all the rows of data would have to be scanned for potential answers to the query. In Multilevel Bitmap index, only the rows that involve the query constraints are processed by the Bloom filter, so the query performance can be extremely fast.

The rest of the paper is organized as follows. In the next section II, we conduct a literature review of some related works. In Section III, we give an overview of Multilevel-

BIBF and discuss our design. We evaluate the performance in Section IV and conclude this paper in Section V

1.1 *Related work*

Junhua Fang developed the index is very important to improve the efficiency of flash-based DBMS. In µ-Tree is a higher efficiency index. The µ-Tree is improvement of B+-tree for flash memory. One of great strengths of µ-Tree is the ability to remove the phenomenon of "wandering trees" that bring by traditional B+ Tree in flash memory. There is a scent of trouble to deal with the update of index and the maintenance of the tree's structure for intensive update or small record update operations. µ-Tree structure to reduce cost in update index and the maintenance of tree's structure .The update-area and self-adaptive mechanism is utilized in the improved µ-Tree structure.

Harbin Lu proposed Dynamic Router table techniques. During insert and delete a records Prefix in B-Tree 30 present less than MRT (Multi way Range Tree). Prefix in B-Tree (PIBT) structure split into two tables for construct prefix in B-Tree. One is longest matching prefix table; another one is Highest-priority Range-table. The Dynamic Router table is faster for update operations.

Bin He, Hui-I Hsiao principles of the Dual Bitmap Index that uses less space than existing bitmap indices while maintaining the same improvements in query processing speed. The Dual Bitmap Index represents each attribute value using only two bitmap vectors, with each bitmap vector representing many attribute values. Compare all five bitmap indexing techniques, and shows that the Dual Bitmap Index is more efficient than the existing techniques.

Bin He, Hui-I Hsiao produced Bitmap index, which builds one bitmap vector for each attribute value, is gaining popularity in both column-oriented and row-oriented databases in recent years. It occupies less space than the raw data and gives opportunities for more efficient query processing. Iceberg query is special type of aggregation query that computes aggregate values above a user provided threshold. Exploited the property of bitmap index and developed a very effective bitmap pruning strategy for processing iceberg queries. The index pruning-based approach eliminates the need of scanning and processing the entire data set (table) and thus speeds up the iceberg query processing significantly.Deke Guo,[5] A Bloom filter is a simple randomized data structure that answers membership query with no false negative and a small false positive probability. It is an elegant data compression technique for membership information and has broad applications.

Attribute A	coarse level			fine level			fine level		
Values	Bin no	equality		Bin no	equality		Bin no	equality	
		=0	=1		=0	=1		=0	=1
A	0	1	0	0	1	0	---	0	0
E	0	1	0	0	1	0	---	0	0
Z	1	0	1	---	0	0	1	0	1
B	0	1	0	0	1	0	---	0	0
O	1	0	1	---	0	0	0	1	0
C	0	1	0	0	1	0	---	0	0
J	1	0	1	1	0	1	---	0	0
R	1	0	1	---	0	0	0	1	0

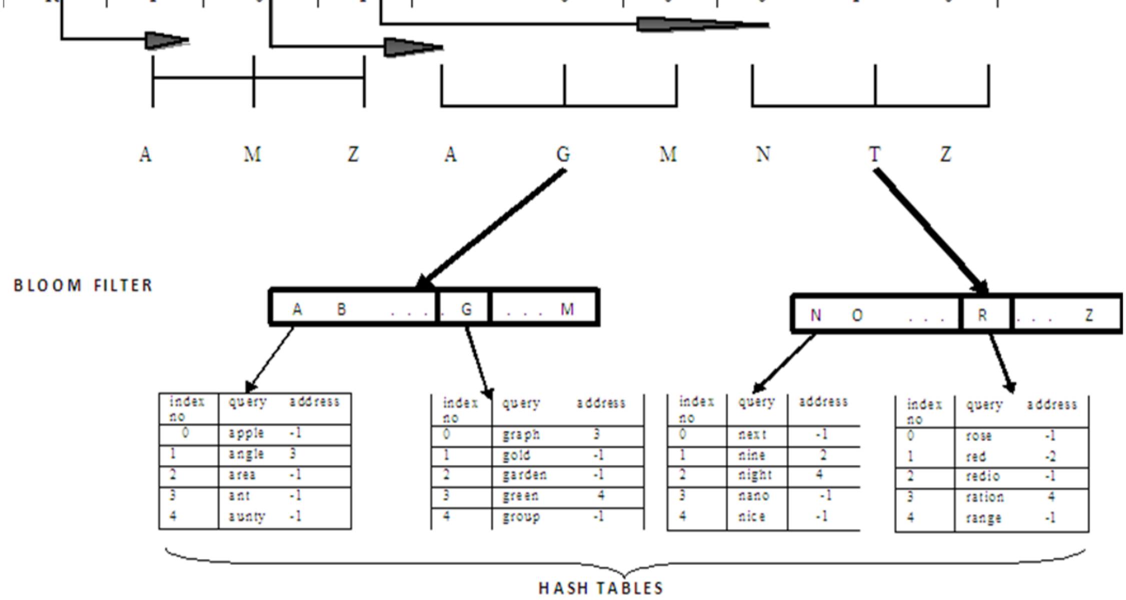


Fig 1: Multi level BIBF index

2. A NOVEL STRUCTURE

2.1. *Multilevel-BIBF*

To index an attribute, Encoded Bitmap Index uses $\lceil \log_2 C \rceil$ bitmap vectors while Simple Bitmap Index uses C bitmap vectors, Interval Bitmap Index uses [C /2] bitmap vectors, Scatter Bitmap Index uses $[2\sqrt{C}]$ bitmap vectors, and Dual Bitmap Index uses $\sqrt{2c + 0.25} + 0.5$ bitmap vectors. The main disadvantage of these Bitmap Indexes, especially Simple Bitmap Index, is that space usage performance degrades as the cardinality in an indexed attribute grows. Although the Encoded Bitmap Index can utilize space requirement efficiently, the complexity of an equality query retrieval functions is undesirable. we proposed structure is composed of Multilevel Bitmap index with Bloom filters and a hash table (HT) that stores the huge volume of items. One novel feature of this hash table is that it uses an improved method of generating verification values. Multilevel Bitmap index with Bloom Filter (M-BIBF), to improve an equality query performance of traditional Bitmap Index. The method makes use of the benefit of multiple hash tables.

2.2 *Proposed Structure*

Figure 1 shows the proposed structure, which is composed of three parts: Multilevel BI ,BF and a hash table. BF and the hash table are used to store multiple attributes and verification values of items, respectively. The idea of a Multi-level indexing is basically to partition the data into bins. To generate a two level indexing, we build the fine-level index first, and then build the coarse-level from the fine-level. This is a practical approach because the total size of fine-level bitmaps is relatively small. In addition, having the fine-level bitmaps also enables us to make more accurate decisions on how to generate the coarse-level bitmaps. Each coarse level bitmap is the result of a bitwise logical OR on a number of fine-level bitmaps. To decide which fine-level bitmaps to group together is equivalent to placing the values corresponding to the fine-level bitmaps; hence , the term coarse-level . Fine bin identify the index location of the Bloom Filter.

Bloom filters were applied to Multilevel Bitmap indices way of provides fast and integrated querying over compressed bitmaps, with direct access and no need to decompress the Bitmap. The bloom filter each block is to contain the index of particular alphabets. Each index is point to the separate hash table. The main drawback of hashing technology is collision. To avoid the collision is very important to handle efficient hash tables. So in our approach solve this problem by using chaining with replacement of hash tables. The actual load of Bitmap can be determined quite accurately by the number of set bits. In the special case of the M-BIBF balancing scheme, the Bloom Filter is divided into d separate sub tables T1, . . . , Td, with a single uniform hash function mapping to each one of these sub tables.

Upon an element's arrival, it is placed in the first sub table in which the corresponds mapped Bloom filter index has a load lower than a pre-defined normal Bitmap index. M-BIBF a lookup operation follows the same steps as an insertion operation: The Bloom Filters of the mapped index are queried one by one, until either the element is found.

Education Organizations Supporting E-Learning

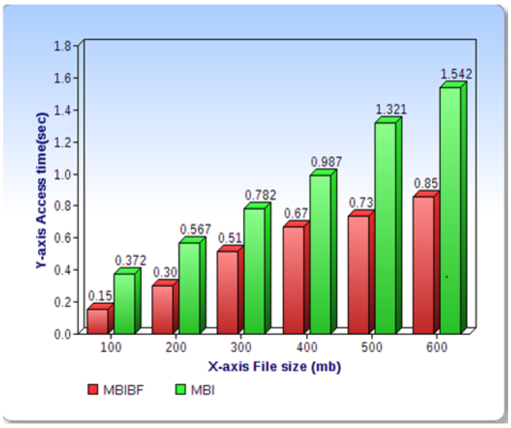
Organization	Characteristics
MedEdPortal, Association of American Medical Colleges (AAMC) (http://www.aamc.org/meded/mededportal/)	Repository All digital content types Material linked to educational competencies Peer reviewed "Virtual patients" bank
End of Life/Palliative Education Resource Center (EPERC) (http://www.eperc.mcw.edu/)	Repository Digital content in end-of-life issues Peer reviewed Links to other online resources
The Health Education Assets Library (HEAL) (http://www.healcentral.org)	Repository Large number of learning assets Growing number of learning objects Peer reviewed
Multimedia Educational Resource for Learning and Online Teaching (MERLOT) (http://www.merlot.org)	Repository for higher education Links to other online resources with peer-review comments Growing science and technology section
International Virtual Medical School (IVIMEDS) (http://www.ivimeds.org/)	A consortium of medical schools Setting standards in medical education Repository for member schools Partnerships Blended learning

3.PERFORMANCE EVALUATION DATA SET

We will analyze the E-Content size of different repositories. We define size as the number of objects present in the repository. We compare the number of objects between repositories of the same domain. In general, individual learning object repositories seems to vary from hundreds to millions of objects. Their average size depends on the type of repository. LORPs can be considered to have few thousand of objects. LORFs are in the order of the tens of thousands. We take more number of e-content data set some of them mentioned below tables it's a free repository and other repositories may require a membership or other fees to cover the ongoing expenses of Web-site maintenance.

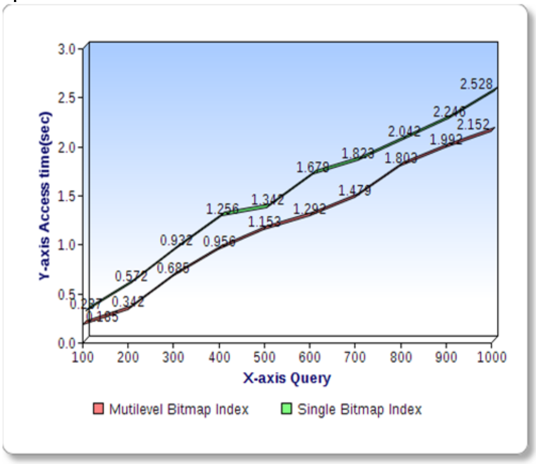
3.1 E-Content Size vs. Query Processing Time

This experiment used the above data set to compare the performance of a simple bitmap index and M-BIBF. Figure shows the average of the processing time required by simple bitmap index query time is proportional to the E-Content size. As the E-Content size grows, the query processing time grows. However, it can be seen that the M-BIBF the retrieval performs better than simple bitmap indexed retrieval even when the multiple hash table base E-Content size is large, as indicated by the slope of the lines. Query processing time grows faster with the increase in E-Content size in case of simple-indexed retrieval as compared to the increase in query processing time for M-BIBF.



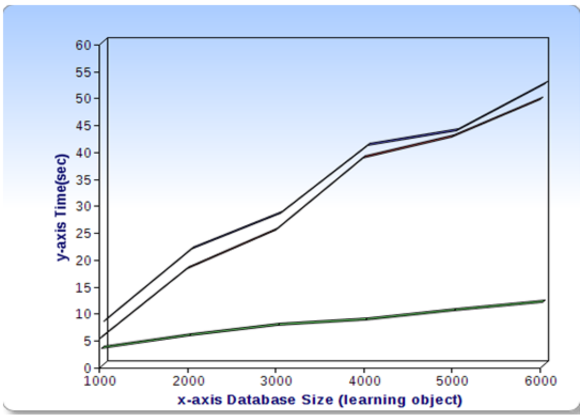
3.2 E-Content Size Vs. Index Creation Time

The index creation time variation with the increase in the E-Content size was studied to find out any erratic behavior as the E-Content size grows very large. Figure shows how index creation time varied with the increase in E-Content size with respect M-BIBF and single Bitmap index. It can be seen that the creation of M-BIBF is faster than other bitmap index.



3.3 E-Content Size vs. Space Requirements

This experiment was conducted to know how the index space varies with the increase in the size of the e-Content. So the Learning object Repository were generated and the size of e-Content was varied. Figure shows how the space required for the index varied with the increase in e-Content size. It can be seen that the space required for the index varied linearly with the increase in the size of the e-Content



4. CONCLUSION

Multilevel-BIBF its reduce search time and quick response because Multilevel-BI based on the BF.BF has multiple hash table it so multiple search perform in a particular time without load (or) delay. In this Multilevel-BIBF reduce space and time complexity. Insert, delete and update operation only change in hash table level not in BF and BI level. So dynamically redefine this indexing scheme is very easy and Multilevel-BIBF indexing scheme is storage efficient and easy to maintain which make it more scalable.

REFERENCES

[1] Wang yan, Lu yipping; zhao thingting, cha jaianzhong "Research of Data Model of SCORM Run-time Environment IEEE 14th International Conference on High Performance Computing and Communications 2010

[2] Yen, Neil Y. Shih, T.K. ; Chao, L.R. ; Qun Jin "Ranking Metrics and Search Guidance for Learning Object Repository" IEEE Transactions On Knowledge And Data Engineering, Vol. 9, No. 11, 2010

[3] Bin He, Hui-I Hsiao, "Efficient Iceberg Query Evaluation Using Compressed Bitmap Index Bin" IEEE Transactions On Knowledge And Data Engineering, Vol. 24, No. 9, September 2012

[4] Can Wang, Zhi-Guang Qin, Lei Yang, Juan Wang, "A Fast Duplicate Chunk Identifying Method Based on Hierarchical Indexing Structure" International Conference on Industrial Control and Electronics Engineering 2012

[5] Deke Guo, Wu, Honghui Chen, and Xueshan Luo,"The DynamicBloom Filter," IEEE Transactions on knowledge and data Engineering, Vol. 22, No. 1, January 2010.

[6] Haibin Lu, Member, IEEE, and Sartaj, Fellow, IEEE " A B-Tree Dynamic Router –Table Design", IEEE Transaction on computer, VOL 54, NO 7, july 2008

[7] Hong Lu, Beng Chin Ooi, Heng Tao Shen, and Xiangyang Xue, "Hierarchical Indexing Structure for Efficient Similarity Search in Video Retrieval" IEEE Transactions On Knowledge And Data Engineering, Vol. 18, No. 11, November 2006

[8] Garrido.A, Maralies.l "E-Learning and Intelligent Planning: Improving Content Personalization" IEEE Transactions On computer vol 54, NO 7 2014

[9] Rafael P. Laufer, Pedro B. Velloso, Otto Carlos, M. B. Duarte,"A Generalized Bloom Filter to Secure Distributed Network Applications," Computer Networks, 55, pp 1804-1819, 2011.

[10] Kala Karun. A, Chitharanjan. K, "A Review on Hadoop – HDFS Infrastructure Extensions" IEEE Conference on Information and Communication Technologies (ICT), 2013

[11] Niwan Wattanakitrungroj and Sirirut Vanichayobon," Dual Bitmap Index Space-Time Efficient Bitmap Index for Equality and Membership Queries" Artificial Intelligence Research Lab, Department of Computer Science, Prince of Songkla University, Thailand 2013

[12] Saber Heni, Ridha Ejbali, Mourad Zaied And Chokri Ben Amar, "A Neural Principal Component Analysis for text based documents keywords extraction" 3rd International Conference on Next Generation Networks and Services, 2011

[13] Sebasti´an Pe na Saldarriaga, Mohamed Cheriet, "Indexing On-Line Handwritten Texts Using Word Confusion Networks" International Conference on Document Analysis and Recognition 2011

[14] WANG Jing-bin and PENG Zhi-xing, "An optimization algorithm for spatial index structure based on spatial clustering" International Conference on Business Computing and Global Informatization, 2011

[15] Xiao Yu and Bo Hong, "Bi-Hadoop: Extending Hadoop To Improve Support For Binary-Input Applications" 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing 2013