Bitmap Index as Effective Indexing for Low Cardinality Column in Data Warehouse

Zainab Qays Abdulhadi*
School of Information science & Engineering
Central South Univesity
Changsha, 410083, China
* Ministry of Higher Education & Scientific Research
Baghdad, Iraq

Zhang Zuping
School of Information science & Engineering
Central South Univesity
Changsha, 410083, China

Hamed Ibrahim Housien**
School of Information science & Engineering
Central South Univesity
Changsha, 410083, China
**Ministry of Higher Education & Scientific Research
Baghdad, Iraq

ABSTRACT

Speeding up query processing is a sensitive issue in the data warehouse. Using some mechanisms like summary tables and indexes can be a solution to this problem. However, though the performance when using summary tables for predefined queries is good but to save space and time during the query processing, indexing is a better solution without extra hardware. The challenge is to find a suitable index type that would enhance the query's performance. Some relational database management systems are carried out by new indexing techniques, such as bitmap indexing, to speed up processing. Bitmap indexes have a particular structure for a quick data retrieval. This paper focuses on measuring the performance of Bitmap as index in data warehouse comparing it with the B-tree index using oracle environment which uses B-Tree as default indexing technique to avoid the problem of(low cardinality column) when an attribute has few values.

Keywords

Data warehouse, Bitmap index, B-Tree index, indexing.

1. INTRODUCTION

In business environment , many organizations collect, organize and update their records of activities in large data warehouses. As years pass, a huge amount of data is gathered within the systems of the companies. As the systems grow fast, some problems in data analyzing occur like slow query executing time. So many indexing techniques have been used for speeding up query processing like B-tree indexes, B-tree cluster indexes ,Hash cluster indexes ,Global and local indexes, Reverse key indexes , bit map indexes, Function-based index [1]. Each existing indexing technique is typical for special situation.

Bitmap indexes are used for complicated and reactive queries in a data warehouse environment to prevent spending long time to access and retrieve answers for the queries. many database systems like oracle are exploiting B-Tree index technique for high cardinality column and Bitmap Indexes have principally been used for low cardinality columns. Bitmap indexes achieve important functions in answering data warehouse's queries because they have capability to perform operations at the index level before fetching data. This speeds up query processing time extremely.

1.1 Factors of indexing of data warehouse

•Distribution: The distribution of a column is the frequency of occurrence each distinct value of the column. The column distribution guides in determining which indexing technique to adopt.

•Value range: The range of values of an indexed column guides in selecting an appropriate index technique. For example, if the range of a high cardinality column is small, an indexing technique based on bitmap should be used.

•Cardinality: The cardinality of a column is the number of distinct values in the column. It is better to know that the cardinality of an indexed column is low or high since an indexing technique may work efficiently only with either low cardinality or high cardinality e.g. Bit map index only works well with low cardinality data[2] . this paper Will choose cardinality as a factor of indexing .

1.2 Features of selecting an appropriate index

There are many features that need to be taken in consideration when choosing an indexing technique

•The index which takes small space and utilizes it efficiently.

•The index should able to operate with other indexes to filtering out the records before accessing original data.

•The index should support ad-hoc and complex queries and also speed up join operations.

•The index should be easy to build (dynamically generated), implement and maintain[5].

2. LITERATURE REVIEW

There are many researchers interested in studies of indexing nature especially in Bitmap index like. Sirirut Vanichayobon, JarinManfuekphan in" Scatter Bitmap: Space-Time Efficient Bitmap Indexing for Equality and Membership Queries" in this paper Variants of Bitmap Indices have been presented to reduce storage requirements and speed up performance. They propose ,Scatter Bitmap Index, which uses less space with the same cardinality while maintaining the query processing time for equality and membership queries. Scatter Bitmap Index achieves this by representing each attribute value using only two bitmap vectors, and only the low-cost Boolean AND operation is used to answer equality queries. Because Scatter Bitmap Index has better space-time performance than the other indexing techniques, a data warehouse using the Scatter Bitmap Index can have a much greater cardinality without losing performance [3]. Rishi RakeshSinhal and his colleagues ,introduced adaptive bitmap indexes (ABIs) As a way to satisfy scientists' demands for a high performance Index with a tiny footprint. An ABI includes a locally –optimal multi-resolution bitmap index and a set of auxiliary projection indexes (PIs) that are materialized while removing false positives from current query answers, then kept in an LRU cache in memory and/or disk for use in answering subsequent queries [4]. , Kesheng Wu, Wendy Koegler and others demonstrate that compressed bitmaps can also be efficiently used to perform region growing and region tracking tasks. On uniform grids, their bitmap based approaches in the research paper (Using Bitmap Index for Interactive Exploration of Large Datasets [5].

Md. GolamRabilulAlam in ((A New Approach of Dynamic Encoded Bitmap Indexing Technique based on Query History)) conclude a method towards dynamic n-items pattern selection for the remapping of lookup table of the specific attribute they have tried to reduce scan time complexity of query history file using normal approach. Though scan time raises significantly it may be noted that still the query processing time remains reasonable for most of the ad hoc queries[6]., ShawanaJamil Conclude that Bitmap Indexes play a key role in answering data warehouse's queries because they have an ability to perform operations on index level before retrieving base table e.g. count based query. The results from this analysis show that Bitmap Indexes can be compressed to reduce storage requirement and speed up performance and also a Bitmap Indexes perform well for statistical query [2]. ,Bin He Conclude that bitmap index has three attractive advantages: Saving disk access by avoiding tuple-scan on a table with a lot of attributes, Saving computation time by conducting bitwise operations, and Leveraging the anti monotone property of iceberg queries to develop aggressive pruning strategies. When he presents an efficient algorithm for iceberg query processing using compressed bitmap indices[7]. ,Weahason Weahama prove that Bitmap indexing techniques have proven to be time-efficient for answering data warehouse queries by performing fast binary operations on the index level, before retrieving base data [8].

3. B-TREE INDEX

Before proceed with the comparison between B -Tree and Bitmap index , this paper briefly review B-tree since it considers as a fast data indexing type. For each B-tree operation, the number of disk accesses raise with the height of the B-tree, which is kept low by the B-tree operations[9] .

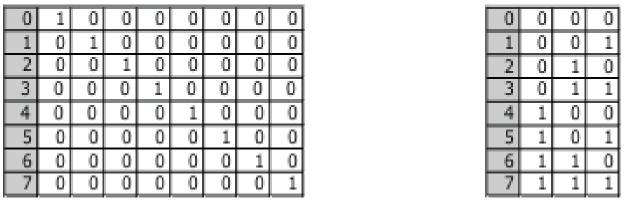
The top most level of the index is called the root. The lowest level is called the leaf node. All other levels in between are called branches , B-tree is widely used in a relational database environment but it cannot handle efficiently on a large amount of data which causes memory overhead in complex and interactive queries [10].

4. BITMAP-INDEX

A bitmap index is a special kind of structure used by most highend database management systems to to gain optimal search and retrieval of low variability data.

O'Neil, Spiegler and Maayan,first present Bitmap index in a form of table such that a processor with aw-bit architecture can process the data within w-rows in parallel. Certain types of queries may benefit from this factorization, thus reducing query response times. In a bitmap index those attributes has only two possible attribute values :present \1" or absent \0" . The size of this representation is larger than a representation where each distinct attribute numbered value with a unique if binary sequential integer starting at zero.

Let n be the number of rows and L be the number of distinct attribute values in a given column, the total of n, L bits in the index. In contrast, if abinary value assign to each distinct attribute value, we obtain a smaller representation of the data with size nlogL. Figure1 shows the size comparison table.



0	1	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0
3	0	0	0	1	0	0	0	0
4	0	0	0	0	1	0	0	0
5	0	0	0	0	0	1	0	0
6	0	0	0	0	0	0	1	0
7	0	0	0	0	0	0	0	1

(a) nL bits in size

0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

(b) $n \log_2 L$ bits in size

Fig 1: Comparison between binary representation and bitmap representations[11]

The first column in each table represents an attribute value, the remaining cells in each row are the bits necessary to represent each attribute value between these two representations, and it can be noted that n log 2L < n L is always true. [11].

Bitmap indexes efficient for ad hoc range of queries Because it perform fast Boolean operations . Bitmap indexes are dynamic and effective tools to get optimal performance of large database This index data structure is mostly used for On-Line Analytical Processing (OLAP) and data warehouse applications [12].

5. THE PROBLEM

The number of values of an attribute in a dataset is known as the attribute cardinality [13]. This kind of repetition in value in the column can be described as data having a low cardinality, that is when the information being displayed can only have a very small number of outcomes; therefore, the same value would be repeated multiple times. One example of this is when values are defined as male and female. There are only two different ways of describing gender so each column in the table would have low cardinality and be perfect to display with a bitmap index [14] .

Selecting appropriate physical structures that improve system performance is the role of data warehouse administrators. However, given the wide development of data warehouses, as well as their structural and operational complexity, minimizing the administration function is a crucial issue [15].

6. ANALYSIS OF PERFORMANCE OF TWO INDEXING TECHNIQUE

The experiments have tested the two indexing techniques on the basis of the time taken for execution . by make three experiments, in each one the query testing process for three conditions, two conditions and one condition

The table below Fig.6 illustrates the performance of bitmap index and B-Tree index By Execution time. These queries are examples of user queries in the data warehouse and the ability to answer these queries efficiently is a critical issue in the data warehouse environment. The data set which has been used to carry out this analysis is acquired from UCAS organization[16] that responsible for managing applications to higher education course .

Records were divided into four subset (10000000 record,800 000 records, 600 000 records,300000 records).The records of applicants students which have column of ID, COLLAGE , REGION , AGE, GENDER .

due to Bitmap indexes consider as the best choice for queries that include multiple conditions in the WHERE clause…. In all experiments where clause statement have been used

1-SELECT ID,AGE,GENDER,REGION,COLLEGE,

FROM TEST_PU_BIT
WHERE SEX = 'FEMALE'
AND AGE = '20'
AND REGION = ' ENGLAND'

2-SELECT ID,AGE,GENDER,REGION,COLLEGE,
FROM TEST_PU_BIT
WHERE AGE = '20'
AND REGION = ' SCOTLAND’

3- SELECT ID,AGE,GENDER,REGION,COLLEGE,
FROM TEST_PU_BIT

WHERE AGE = '21'

The first chart describes the execution of where clauses with three conditions, Second chart described the execution of where clause with two conditions, third chart described the execution of where clause with one conditions , experiments Executed in oracle environment and the summary of results of Expirment1 are figured in the fig(2),fig(3),fig(4)

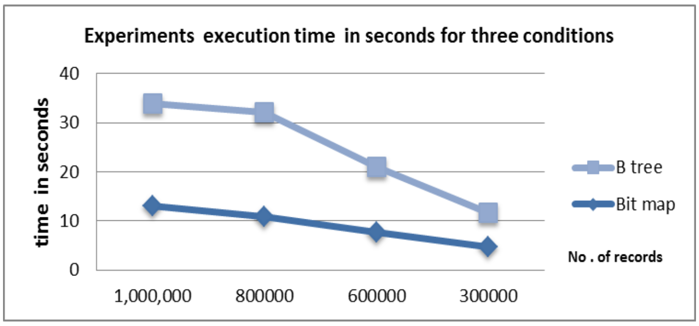


Fig.2 Experiment1execution time in seconds for three conditions

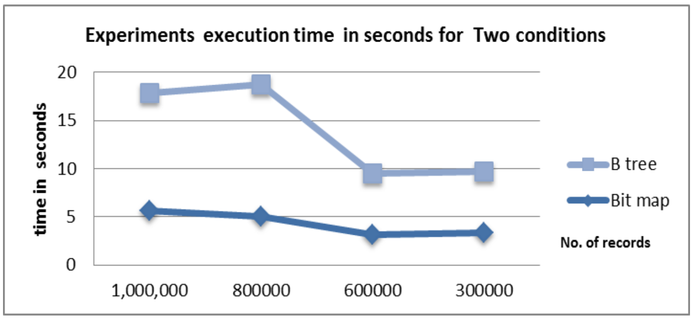


Fig.3 Experiment1 execution time in seconds for two conditions

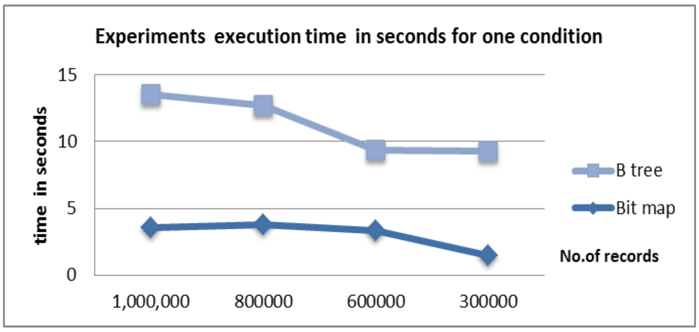


Fig.4 Experiments1 execution time in seconds for one condition

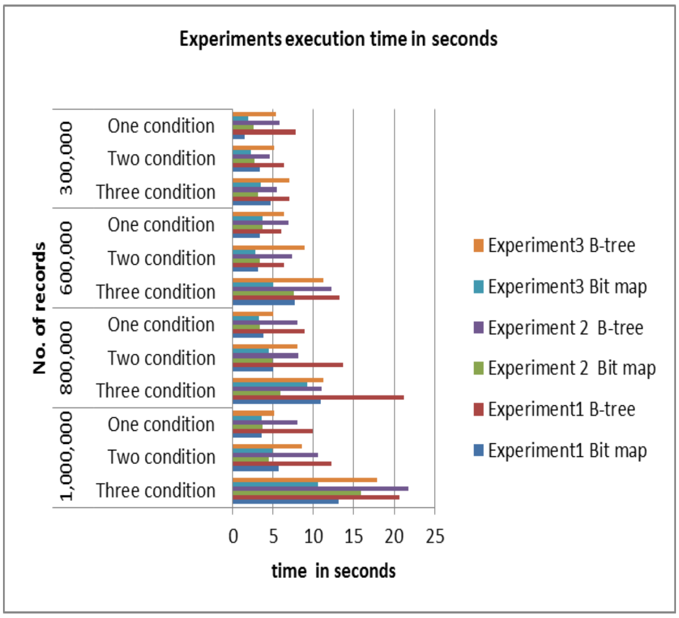
The result of experiments 2 and experiment 3 illustrated in the table below in fig(6) and fig(7)

No. records	No . of condition	Experiment1		Experiment 2		Experiment3	
		Bit map	B-tree	Bit map	B-tree	Bit map	B-tree
1000,000	Three condition	13.15	20.67	15.89	21.81	10.60	17.88
	Two condition	5.63	12.24	4.43	10.52	4.87	8.54
	One condition	3.58	9.94	3.68	8.06	3.57	5.17
800,000	Three condition	10.86	21.17	5.94	11.04	9.28	11.26
	Two condition	5.02	13.68	4.87	8.13	4.48	7.97
	One condition	3.81	8.88	3.37	7.98	3.22	5.01
600,000	Three condition	7.72	13.28	7.55	12.24	5.04	11.24
	Two condition	3.15	6.32	3.32	7.33	2.80	8.93
	One condition	3.34	6.02	3.74	6.93	3.74	6.39
300,000	Three condition	4.74	7.02	3.12	5.51	3.47	7.00
	Two condition	3.38	6.33	2.68	4.56	2.19	5.18
	One condition	1.48	7.79	2.57	5.76	1.95	5.34

Fig.6 Table of the results of experiments 1, experiment2 and experiment 3 .

From the three experiments it is observed that bitmap indexes are faster and take less execution time rather than B – tree especially for columns having Boolean type like column value like having ('Y' or "N', 'Male' or 'Female') .

So bitmap indexes are only convenient for static tables which are updated at the close, so if the tables are not read-only during query time, B-Tree index is an effective choice than bitmap index. Experiments show that Bitmap index is useful for Decision Support System, OLAP ,data warehouse and B-tree Index is very useful for the OLTP environment. So an study was needed to find wich an indexing techniquewill be used in data warehouse depending on the time needed to implement specific query type.



Fig(7) chart of response time for all experiments

7. CONCLUSION

The experiments show that bitmap indexes are more beneficial than B-tree indexes particularly when the table has millions of rows like data warehouse and the columns have low cardinality. Moreover, bitmap indexes provide a better performance compared to b-tree indexes whene queries use a combinations of multiple conditions with OR/AND operators. Bitmap index is also a convenient index for the table which is read-only, or when there is a low updating activity on the key columns finally , bitmap indexes offer important function for saving space and time as well as very good query performance in large data warehouse.

8. REFERENCES

[1] Oracle® Database Administrator's Guide, “Managing Indexes”, available at : http://docs.oracle.com/cd/B19306_01/server.102/b14231/indexes.htm.

[2] ShawanaJamil , RashdaIbrahim , “Performance Analysis of Indexing Techniques in Data Warehousing” , IEEE International Conference on Emerging Technologies，2009.

[3] SirirutVanichayobon, JarinManfuekphan , Le Gruenwald “Scatter Bitmap: Space-Time Efficient Bitmap Indexing for Equality and Membership Queries” ，Conference IEEE，2006. .

[4] Sinha, R.R.; Winslett, M.; Kesheng Wu; Stockinger, K.; Shoshani, A.， “ Adaptive Bitmap Indexes for Space-Constrained Systems” Data Engineering, 2008. ICDE . IEEE 24th International Conference on 2008

[5] Kesheng Wu; Koegler, W.; Chen, J.; Shoshani, A. “ Using Bitmap Index for Interactive Exploration of Large Datasets”,2003.

[6] Md. GolamRabilulAlam, Mohammed Yasir Arafat, Mohammed Kamal UddinIftekhar，“A New Approach of Dynamic Encoded Bitmap IndexingTechnique based on Query History“IEEE, 5th International Conference on Electrical and Computer Engineering，ICECE 2008

[7] He, Bin; Hsiao, Hui-I; Liu, Ziyang; Huang, Yu; Chen, Yi ,” Efficient Iceberg Query Evaluation UsingCompressed Bitmap Index” IEEE JOURNALS & MAGAZINES,2012.

[8] WeahasonWeahama, SirirutVanichayobon and JarinManfuekphan ,“ Using Data Clustering to Optimize Scatter Bitmap Index “,,2009.

[9] Cormen T. H., Leiserson C. E., Rivest R. L. “Introduction to algorithms, McGraw-Hill ,2000

[10] sirirut Vanichayobon, Le Gruenwald, "Indexing Techniques for Data Warehouses Queries", The University of Oklahoma ,1999.

[11] Eduardo Gutarra Velez,”Multi-column Bitmap Indexing”,Master Thesis of Computer Science (MCS)of Computer Science, EAFIT, 2009.

[12] Kurt Stockinger,Kesheng Wu,Arie Shoshani,”Strategies for Processing ad hoc Queries on Large DataWarehouses, .

[13] Kesheng Wu, Kurt Stockinger and ArieShoshani ,”Breaking the Curse of Cardinality on Bitmap Indexes” 2008.

[14] Wisegeek “What Is a Bitmap "Index”, available at : http://www.wisegeek.com/what-is-a-bitmap-index.htm5

[15] StéphaneAzefack, KamelAouiche,” Dynamic index selection in data warehouses “IEEE Conference ,2007.

[16] USAS, “annual datasets”, available at : http://www.ucas.ac.uk/about_us/stat_services/stats_online/annual_datasets_to_download/.