**International Journal of Management and Commerce Innovations** ISSN 2348-7585 (Online)
Vol. 4, Issue 1, pp: (606-608), Month: April 2016 - September 2016, Available at: www.researchpublish.com

# Comparative Study of Bitmaps Indexing and B- Tree

Sudhanshu Deshmukh[1], Pravin S. Metkewar[2]

[1]MBA-IT 1st Year Student, SICSR, Affiliated to Symbiosis International University (SIU), Pune, Maharashtra, India
[2]Assoc. Professor, SICSR, Affiliated to Symbiosis International University (SIU), Pune, Maharashtra, India

***Abstract:*** **In spite of the fact that tracking records on the database are for the most part viewed as a typical issue, it assumes a key part in the enquiry execution, especially on the account of enormous databases such as information warehouse. Should a fitting file structure be chosen, the time required for enquiry reaction will diminish widely. Most specialists go for the bitmap list as favoured indexing method for situations where cardinality is low. once the list size is large, the cardinality of ordered sections expands ,this brings the reaction time to rise. The B- tree record is thought to be a solution for this problem. We concentrate on various kind of bitmap indexing systems (straightforward bitmap and encoded bitmap) and perform collected operation on inquiry with the help of both basic and encoded bitmap indexing and examinations the outcome. This paper tries to show the comparative study of bitmaps indexing and B- trees on the parameters such as response time, throughput time.**

***Keywords:*** **Data warehousing, Bitmap Index, BTREE Index, Query Processing, Response Time.**

## 1. INTRODUCTION

The data in the data warehouse is stored in the form of records. Every record has a key field associated with it, that helps it to be recognized uniquely.

Indexing is a data structure technique to efficiently and quickly retrieve records from the database files based on some attributes on which the indexing has been done.The simple example of indexing is the index of a book which helps us to identify the exact page number of the chapters shown on the index page.

### 1.1 Simple bitmap INDEX:

The fundamental thought process of a straightforward bitmap index is showing to the Value of quality with some series of bits (1 or 0). For instance, in a quality GENDER, there are two qualities, FEMALE,MALE. Presently for these qualities we can show them by setting bits to every quality I.e for "M" we set "1" when GENDER=M. Now for "F" we can set "0" when GENDER=F. At the point when the estimation of quality is extensive i.e. |A| = LARGE then setting bits to the relating esteem require immense space. ENCODED BITMAP is the solution for this problem.

### 1.2 B- Tree Index:

The index leaf nodes are stored in an arbitrary order—the position on the disk does not correspond to the logical position according to the index order. It is like a telephone directory with shuffled pages. If you search for "MR.X" but first open the directory at "Mr.Y", it is by no means granted that X follows Y. A database needs the second structure to find the entry among the shuffled pages quickly and efficiently the B-tree solves this problem.

### 1.3 High Cardinality Values:

High-cardinality values refer to columns with values which are very much unique and not common in the table.Each row in the table will have a unique identity.The simple example which can be taken is a Mobile number,each individual's data stored in the table will have his/her's unique phone number which won't be common .So that is the high cardinality entity.

### 1.4 Low Cardinality Values:

Low cardinality values refer to very few unique values in the column of the corresponding table.An example of this is shown in the table1 where the low cardinality column is the city column ,this column has very few unique values i.e there are multiple rows which have the same rows and hence the cardinality is low.

## 2. METHODOLOGY

***Experiment Setup:***

A data set of 1000 entries was taken for the experiment. A database with name data warehousing was created and it had a table named data,it had in total 5 columns.There were few columns with high cardinality and few with low cardinality.The below table shows the table structure of the database table and the type of values used in the table.

**TABLE 1**

| First_name | Last_name | City | Gender | Email_id |
|---|---|---|---|---|
| **Charles** | Vinson | Pune | Male | charles@live.com |
| **Adams** | Smith | Pune | Male | adams@live.com |
| **Lauren** | Foster | NY | Female | Lauren@xyz.com |
| **Miles** | Nelson | Harare | Male | Miles@xyz.com |

The Low cardinality columns in the table were city,gender whereas the high cardinality column was the email_id column.These were the columns on which database queries were applied and also indexing was applied.

Once the data was finalized and the data cleansing was performed i.e removal of blank spaces and redundant data from the 'data' table,indexing was applied on the table.The indexes used or created in this table were Bitmap index and BTREE index.A common query was used for bitmap indexing on HIGH cardinality values and for BTREE indexing for HIGH cardinality values.Similarly, another query was used for bitmap indexing on LOW cardinality values and for BTREE indexing for LOW cardinality.After Creating the bitmap and BTREE indexes on the table of the databases,The Queries applied were as follows :

**TABLE 2**

| |
|---|
| HIGH CARDINALITY ON BITMAP AND BTREE INDEXES |
| SELECT * FROM DATA WHERE EMAIL_ID="ALIJA.MUR889@LIVE.COM "; |
| HIGH CARDINALITY ON BITMAP AND BTREE INDEXES |
| SELECT * FROM DATA WHERE CITY="PUNE" |

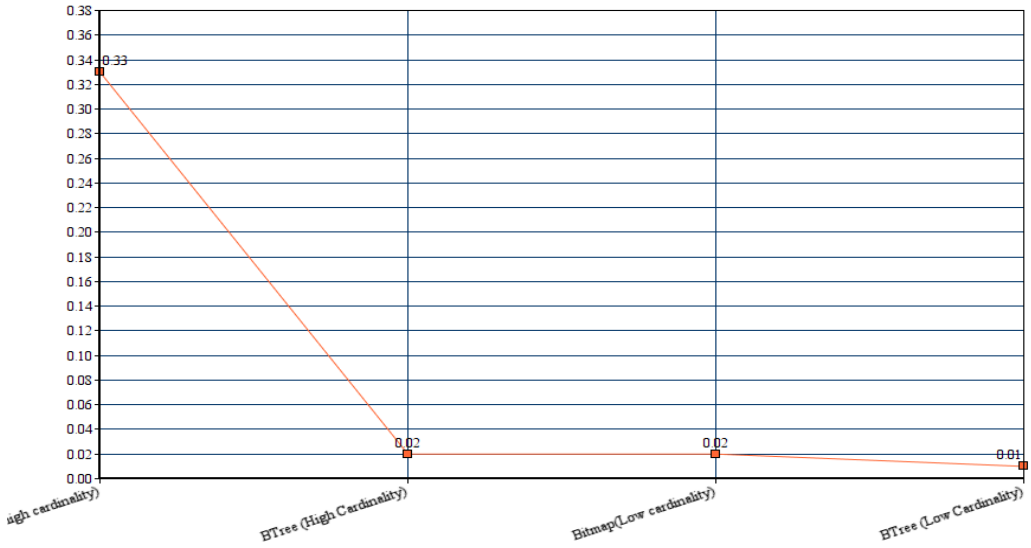
## 3. RESULTS AND DISCUSSIONS

After performing the experiment on the data it was observed that BTREE indexes have less response time on high cardinality values as compared to Bitmap indexing.The Variation of Query response time observed was very high after applying the queries.While on the other hand there was not many queries response time variation when the similar query was applied on low cardinality values for both Bitmap indexes and BTREE indexes.

**TABLE 3**

| BITMAP INDEX | BTREE INDEX |
|---|---|
| HIGH CARDINALITY : 0.33ms | HIGH CARDINALITY : 0.01ms |
| LOW CARDINALITY : 0.02ms | LOW CARDINALITY : 0.01ms |

In the results for high cardinality values, there was only one output row displayed whereas in the case of low cardinality values there were 33 rows displayed which had similar entries of the city name in the 'data' table.




## 4. CONCLUSIONS

It is commonly accepted that Bitmap index is more efficient for low cardinality attributes but in our experiment there was not many query response time variation in between the bitmap and BTREE indexes.The response time was almost the same for low cardinality values.But when it came to high cardinality values there was a large query response time variation between BTREE and Bitmap index.We can come to a conclusion that BTREE indexing can be used for High cardinality values whereas Bitmap indexing and BTREE indexing will give similar efficiency on Low cardinality values.

## REFERENCES

[1] P. O'Neil, Model 204 Architecture and Performance. In Proceedings of the 2nd international Workshop on High-Performance Transaction Systems, Lecture Notes In Computer Science, vol. 359. Springer-Verlag, London, (September 28 - 30, 1987),pp.40-59

[2] R. Kimball, L. Reeves, M. Ross, The Data Warehouse Toolkit. John Wiley Sons, NEW YORK, 2nd edition, 2002

[3] W. Inmon, Building the Data Warehouse., John Wiley Sons, fourth edition, 2005

[4] C. DELLAQUILA and E. LEFONS and F. TANGORRA, Design and Implementation of a National Data Warehouse. Proceedings of the 5th WSEAS Int. Conf. on Artificial Intelligence, Knowledge Engineering and Data Bases, Madrid, Spain, February 15-17, 2006 pp. 342-347