

The Impact of Partitioned Fact Tables and Bitmap Index on Data Warehouse Performance

Emany Sidi

Laboratory modeling and information theory Abdelmalek Essaadi University, Tétouan, Morocco

Mohamed El Merouani

Laboratory modeling and information theory Abdelmalek Essaadi University, Tétouan, Morocco

El Amin A. Abdelouarit

Laboratory modeling and information theory Abdelmalek Essaadi University, Tétouan, Morocco

ABSTRACT

The design process is the most considered task of a data warehouse designer, this because of its performance criticality while going to production.

Even if technology has developed in term of memory volume and speed, storage... the fact to minimize the execution time and the storage space still being a preoccupation for data warehousing specialists.

This papers shows that using bitmap index and partitioned fact tables in data warehouse based on a star schema is more advantageous based on compression.

General Terms

Business Intelligence, Data Warehouse.

Keywords

Data Warehouse, DBMS, Indexes, Business Intelligence

1. INTRODUCTION

The data warehouse physical design is considered as the most important task, including how to improve access to this data.

This is the main reason that makes data warehouse performance as the main concern for designers. This performance includes memory performance

It's proved before that using a star schema with partitioned tables and Bitmap Index [1], the focus now is on the performance optimization in term of memory and space disk usage.

2. ROLAP SYSTEMS

ROLAP Systems are based on relations of a data cube, it's constituted from fact and dimension tables. The fact table contains in their attribute values of activity results and a foreign key to each dimension. The advantage of ROLAP is the use of existing databases which reduces it implementation cost [8] [10].

This study is to prove how the memory and disk space usage has improved when using bitmap and partitioned table on data warehouse based on a star schema.

3. STAR SCHEMA

In this model, each group of dimensions are placed in a dimension table, the facts are placed in fact table. The result is a star schema where the fact table are in center rounded by the dimensions tables [11], the dimension tables contains qualitative data represented in a big number of attributes. These qualitative data supports many analyze processes. In other side, the fact table has an important number on instances, each tuple in fact table has two kind of attributes:

- Foreign keys referencing to the dimension table
- A set of measures that can be aggregated to perform treatments.

The fact table is generally normalized, but the dimension ones are not; the queries that are used in these schema are called "Star join query".

The figure 1 shows an example of a star schema.

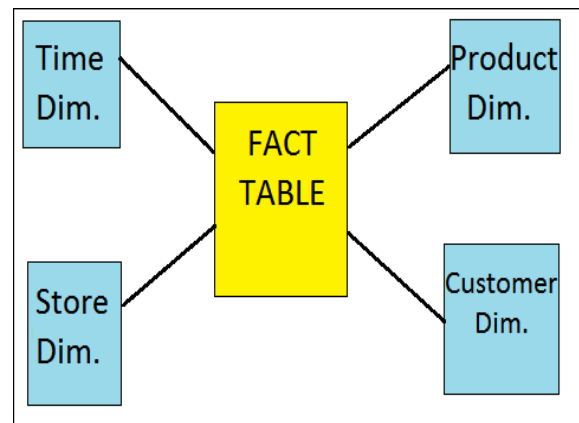


Fig 1: Example of Star Schema

4. BITMAP INDEX

4.1 Definition

It's a data structure defined in a DBMS used to optimize access to data in databases. It is a type of indexing is particularly interesting and effective in the context of selection queries. The index bitmap attribute is encoded in bits, where its low cost in terms of space occupied. [8] All possible attribute values are considered, the value is present or not in the table. Each of these values is an array of bits, called bitmap, which contains as many bits as n-tuples present in the table. Thus, this type of index is very effective when the attributes have a low number of distinct values. Each bit represents the value of an attribute for a given tuple. For each bit, there is an encoding presence / absence (1/0), which indicates that a tuple or not the present value characterized in bitmap.

Table 1: Basic Bitmap adopted by [10]

ROWID	C	B0	B1	B2	B3
0	2	0	0	1	0
1	1	0	1	0	0
2	3	0	0	0	1

3	0	1	0	0	0
4	3	0	0	0	1
5	1	0	1	0	0
6	0	1	0	0	0
7	0	1	0	0	0
8	2	0	0	1	0

To illustrate how a bitmap index works, we take an example EE-PP-O'Neil and O'Neil [7]. Table 1 illustrates a basic bitmap index into a table containing 9 records, where the index is created in the C column with integers ranging from 0 to 3, we say that the cardinality of the column C is 4, by what there are 4 distinct values [0, 1, 2, 3], where the index bitmap C Contains 4 bitmaps shown as B0, B1, B2 and B3 corresponding value represents. In this example, the first line where RowID = 0, column C is worth 2, consequently, B2 column bit value "1", while the other bitmaps are set to "0". Same for the next line, where C = 1 corresponds to the bitmap B1 is set to 1 and the rest to "0". This process is repeated for the remaining lines.

4.2 Properties

Bitmap indexes have a very interesting property of responding to certain types of requests without returning the data themselves, thus optimizing the response time, disk storage. This is possible by counting operations (COUNT) and logical operators (AND, OR, etc.) that act "bit by bit" on bitmaps.

5. HYPOTHESIS

The bitmap index is useful for faster access to data [2][3][4][5][7], but when using a star schema, this index is not efficient when the fact table contains a large data [6]: Less number of foreign keys and hence shorter query execution time (faster), to make it real, we look forward the use of partitioned tables, this will reduce the data volume in table: data will be distributed in many tables, so it conduces to less foreign key.

To do that, we have to divide the fact table in many partitions based on a dimension, and would be easier to take the time dimension to partition it (in months, years, decade...)

So this experimentation is to demonstrate that by using bitmap index and needs to partition fact tables in a star schema to get best memory and disk space optimization.

Note that we take two similar environment with SQL Server as RDBMS (Relational Data Base Management System).

6. ANALYSIS AND RESULTS

To satisfy this study we created the following Data warehouse sample Schemas:

- Star schema 1 : 1 fact table 3 dimension with bitmap index and fact table partitioned based on age dimension
- Star schema 2 : 1 fact table 3 dimension with bitmap index only
- 6 million rows inserted in both fact tables with same data (by respecting it dimensions)

To get a deep analyze it's better to run same query on both schemas (Table2)

6.1 Technical specification of lab environment

This study has been executed in similar technical environment that we present in the following table: (Table 1)

Table 2. Technical specification of lab environment

Physical Memory	Cores	Storage Disk	Operatio n System	RDBMS
8 Gb	8 Cores	1 TB	Win 2008 Server	MS SQL SERVER 2008 R2

The huge resources that we offer has for objective to make difference between old decision systems with lower resources and the new decision systems with high technology, high resources.

Table 3. Star join query used

```
SELECT d1.att2, d3.att2, d1.att1,
SUM(f.measure1)
from Fact1 f inner join dim1 d1 on
d1.dimkey1 = f.dimkey1
GROUP BY d1.att1
```

6.2 Used query

The objective is to stress these 2 data warehouses schemas before and after applying the bitmap index and partitioning table to star schemas, this application will be called in this work as optimization tasks.

The comparison will be based on memory occupation and used disk space.

6.3 Query execution results

6.3.1 Query execution results on schema without partitioned fact table

The test is done first on the Data warehouse schema where there is no fact table partitioned, starting the test on this table has as objective to show the improvement when applying the partitioned table approach on data warehouse fact tables.

Occupied cores	% Memory Occupied	Real time Query execution in seconds	Fact Table Size in Tb (compression)
4 cores (50%)	6718 Mb 82%	14s	0,701 Tb

6.3.2 Query execution results on schema with partitioned fact table

After executing same query on a partitioned fact table in star schema per time dimension d1 (per year), the results are as the following:

Occupied cores	% Memory Occupied	Real time Query execution in seconds	Fact Table Size in Tb (compression)
7 cores (87,5%)	3360 Mb 41%	8s	0,519 Tb

6.4 Results analysis

Based on this results, the use of partitioned fact tables and bitmap index are optimizing the data warehouse resource

using, note that the use of huge physical memory and less cores in first query is qualified as a bad use of physical resources.

The partitioned fact tables let the RDBM optimize the use of physical resource by;

- Using sufficient core by a distributed query execution; partitioned fact table means many fact table by age dimension.
- Optimum use of physical memory, distributed results in same time.
- Less rows in partitioned fact table means high performance of bitmap index.

7. CONCLUSION AND FUTURE WORKS

Using partitioned tables has highly optimized the performance by using the right physical memory resources, this independently of wish schema type: star or snowflake.

The most important is that the partitioned fact tables converted the data warehouse in many small data mart that distributed the charges while query execution.

In other side, the bitmap was useful with the partitioned tables that permits a better compression, this were highly noted in results of used disk space.

The message here is clear, bitmap index must be used with a suitable data ware house configurations, and among these configurations, the use of partitioned fact tables.

These results opened for future work to discuss how useful will be this configuration for the decisional dashboard and business reporting, as per it importance for business intelligence purpose.

8. REFERENCES

- [1] Emany Sidi, Mohamed El Merouani and El Amin A Abdelouarit. Article: Star Schema Advantages on Data Warehouse: Using Bitmap Index and Partitioned Fact Tables. *International Journal of Computer Applications* 134(13):11-13, January 2016.
- [2] E. Abdelouarit, M. El Merouani, A. Medouri, The bitmap index advantages on the data warehouses

American Academic & Scholarly Research Journal Vol. 6, No. 4, July 2014

- [3] E. Abdelouarit, M. El Merouani, A. Medouri, Data Warehouse Tuning: The Supremacy of Bitmap Index *International Journal of Computer Applications (0975 – 8887)* Volume 79 – No7, October 2013.
- [4] E. Abdelouarit, M. El Merouani, A. Medouri, The impact of indexes on data warehouse performance *IJCSI International Journal of Computer Science Issues*, Vol. 10, Issue 5, No 2, September 2013.
- [5] E. Abdelouarit, M. El Merouani, A. Medouri, OPTIMISATION DES PERFORMANCES DES ENTREPÔTS *Rev. Ivoir. Sci. Technol.*, 20 (2012) 35 - 67 ISSN 1813-3290, <http://www.revist.ci>
- [6] S. Chaudhuri, U. Dayal, An Overview of Data Warehousing and OLAP Technology., *ACM SIGMOD RECORD*. 1997
- [7] E. E-O'Neil and P. P-O'Neil, Bitmap index design choices and their performance implications, *Database Engineering and Applications Symposium. IDEAS 2007. 11th International*, pp. 72-84.
- [8] R. Kimball, L. Reeves, M. Ross, *The Data Warehouse Toolkit*. John Wiley Sons, NEW YORK, 2nd edition, 2002.
- [9] W. Inmon, *Building the Data Warehouse.*, John Wiley Sons, fourth edition, 2005.
- [10] 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.
- [11] R. Kimball and K. Strehlo. Why decision support fails and how to fix it. *SIGMOD Record*, 24(3) :92-97, September 1995.