Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 6.017

IJCSMC, Vol. 7, Issue. 12, December 2018, pg.341 – 350

A Comparative Study of Indexing using Oracle and MS-SQL Server for Relational Database Management Systems

Rafat Talib Hashim
Department of Computer Engineering Technologies, Imam Khadum College (IKC), Iraq
rafathashim@alkadhum-col.edu.iq

Abstract:** Relational Database Management Systems (RDBMS) maintain a collection of huge data files to provide fast and efficient methods in order to access and modify data which is necessary (Martin et al., 1992). Therefore, RDBMS have supported indexing techniques in order to access the data efficiently in static and dynamic manners. The majority of commercial (RDBMS) Relational database management systems performance is relied on I/O operations rather than other computing resources. This is because the performance cost of I/O is expensive and there are other costs such as memory allocations and CPU consumption. The most important factor to consider is whether the I/O subsystem of a given (RDBMS) will support a reliable performance as time passes. This article describes a methodology for evaluating indexing techniques for relational databases. The methodology is based on a number of experiments to test a set of indexing techniques on two different platforms (Oracle and MS-SQL Server) with different data sizes (small, medium, and very large) over the same technical environment (Multiple processors, memory, and I/O devices). This factor is necessary to satisfy the real results on different platforms: Oracle and MS-SQL Server. To run the experiments, we have taken the following indexing techniques in Oracle: B-tree, Bitmap, Reversed, and organization index. In the meanwhile, we have taken the following indexing techniques in MS-SQL Server: B-tree, Clustered index, and unique non-clustered index and Primary Key Clustered index. The methodology includes the technical environment, platforms, table schema, table sizes, and a number of indexing techniques. We have also established a number of test scenarios to achieve the real results. The methodology procedure and flowchart include an ordered set of steps that have been taken to run the experiments in this article. The aim of the methodology contribution is to measure overall performance and behavior of indexing techniques that are performed against the same set of data: As a non-clustered index on a specified set of columns, and as a clustered index on the same set of columns. Note that, we have measured the performance of SELECT operation over Oracle 10g and MS SQL Server on data sizes (5000K).

Introduction

In 1970's, Codd outlined a new approach to database construction that was based on a Relational Model of Data for Large Shared Data Banks. He described a new system for storing and working with large databases. Instead of records being stored in some sort of linked list of freeform records as in Codasyl, Codd's idea was to use a "table" of fixed-length records. A linked-list system was inefficient when storing "sparse index" databases where some of the data for any one record could be left empty. The relational model splits the data into a series of normalized tables. Nowadays, databases have matured and required new technological advancements in order to raise its overall performance, and sustain heavy loads of data**.(1)** Query processing and optimization have always been one of the most critical components of database technology. This component deals with efficient and effective processing of user queries against a database. The purpose of the query processing and optimization is to find user-defined data from large database effectively and with an acceptable accuracy (Clement et al., 1997). This sort of optimization is performed mostly by utilizing indexes to facilitate the quick access to user defined queries. The research methodology that will be followed is directly connected to our problem statement and contributions of this article. Since the article purpose and problems may vary different methods of research can be utilized. A systematic literature review has been performed to analyze all the facts about indexing with the focus on comparing experimental tests and the indexing techniques (B-tree index, Bitmap index, reverse index, and organization index) on Oracle platform and the indexing techniques (Primary key clustered, unique clustered index, and non-unique clustered index) on MS SQL Server platform with size of data (5000K) and practical results on table indexing.**(2)**

Methodology Contributions

The contribution of our test methodology is to characterize the performance and behavior of DML operations performed against the same set of table data organized:

•As a non-clustered index on a specified set of columns.

•As a clustered index on the same set of columns.

In this test, a number of questions should be answered through the experimental results as follows:

1 .Are clustered indexes necessary for all tables?

2. Are non-clustered indexes necessary for all tables?

3 . What are the performance gains or losses for row-by-row SELECT operations executed against three different sizes of tables (100K, 1000K, and 5000K) with a clustered index versus the same tables without a non-clustered index for a high-throughput workload on Oracle 10g and MS-SQL Server 2005 platforms?

4. How does a range query perform on the same tables with a clustered index versus a non-clustered index?

5. What are the effects of having the first column index be monotonically increasing? The purpose of this test is to measure performance.

6. What are the CPU utilization characteristics when rows are selected from a table with a clustered index and from non-clustered index?

Criteria for Comparison

Each type of index is related to query evaluation algorithms that access the requested information, and update algorithms that maintain it. There are many criteria by which indexing techniques can be compared. We need to consider the overall speed, space requirements, CPU time, memory requirements, measurements of disk traffic such as numbers of seeks and volumes of data transferred, and ease of index construction. All of these considerations will be in the context of assumptions made about the properties of the data and queries (Zobel et al., 1996).

Methodology Assumptions

To make a contribution to the study of indexing, it is not sufficient to simply describe the current indexing techniques**.(3)** It is also necessary to provide a demonstration of the value of the method, and place it in the context of other established methods. This demonstration will be based on several constraints and assumptions: the class of data (such as textual and multimedia data), the class of queries (such as SELECT queries), characteristics of the application, for example-and characteristics of the supporting hardware for both MS SQL Server and Oracle. Database Administrators (DBA) will judge the success of the used technique according to its performance on the basis of the stated. Assumptions should not only be claimed to be reasonable, they should be argued for, and, where possible, demonstrated as being reasonable. Similarly, assumptions about hardware should associate with the current technology or likely future improvements. The performance of the hardware should be related to common benchmarks, to allow comparison with familiar systems and to convey the impression that the technique will be of value on probable hardware-rather than a machine with limited memory but massive arrays of parallel disks**.(4)**

Methodology Description

The main aim of this section is to conduct the tests described in the previous section against throughput workload that represented real-world scenarios. Another aim is to keep the test setup (server configuration, database settings, table schema, computer configuration and Operating System) approximately constant across the tests so that we compare the overall performance between different SELECT operations using different indexes**.(5)** After some testing and analysis using real-world measurements, we have noticed that the testing results could not represent the real performance measurements. This is because the experiments are conducted on standalone

machine. Even though, we have obtained results in a way that would make these results meaningful and applicable to a wide variety of other workloads. Based on our findings, we have drawn a number of recommendations for DBA's and researchers. This methodology is used for the tests explained in Experiments Test and Results section. Our intent is that these individual tests might help the DBA's and researchers to estimate the overall impact of the index choices for a particular application on both platforms: Oracle and MS- SQL Server. Further intent is that the obtained results could ease the selection of the optimal indexing techniques for a certain application and platform**.(6)**

Technical Test

We have conducted all tests on computer hardware that was configured with adequate storage. We have used a Dell with Core ™ i7 CPU processors: 720 @ 1.60GHz and 4-GB memory. The tests are performed on Windows 7 Home Premium for both platforms: MS SQL Server 2005, and Oracle 10g.

Test H/W & S/W (Test Environment)

We have obtained results for the following test scenarios as shown:

SELECT performance

1- Measure the time taken to select 5000K rows of data from the table with the primary clustered index, unique clustered index and unique non-clustered index by using individual (row-by-row) select statements in MS SQL Server Platform.

2- Measure the time taken to select 100K, 1000K and 5000K rows of data from the table with the Bitmap index, B-tree index, Reverse index and Organization index by using individual (row-by row) select statements in Oracle Platform.

Test Procedure

As shown in Figure 2.1 and Figure 2.2, our methodology procedure was performed on two platforms: Oracle 10g, and MS-SQL Server, respectively. The following steps are used to execute the tests over Oracle platform as described in Test scenarios .

1 The table of size 100k is created and initialized on Oracle platform.

2 The bitmap index is created on the table.

3 The particular test is executed.

4 The table is dropped.

5 The table of size 100k is created and initialized on Oracle platform.

6 The above steps (1-5) have been repeated for other indexing techniques (unique index (Btree), reverse index and organization index).

7 Note that the procedure that contains steps from 1 to 7 has also repeated for data sizes (5000 K) On the MS-SQL Server platform, we have performed the above procedure (steps 1- 7) taking into considerations the following indexing technique

1 Primary key clustered index

2 Unique clustered index

3 Unique non-clustered index

Figure 1 illustrates the flowchart of the test procedure. Note that this process of flowchart shows the steps from 1-7 on different platforms with different data sizes. Figure 2 illustrates Flowchart of the index evaluation on MS-SQL Server platform.

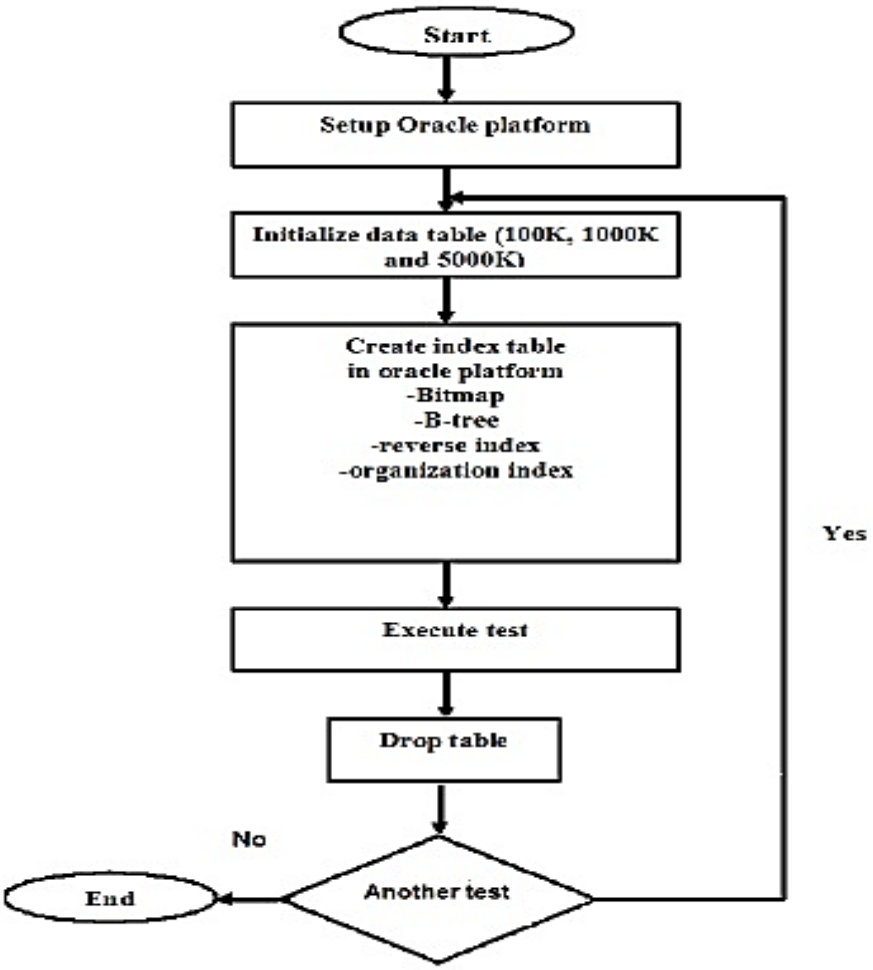

Figure 1 Flowchart of index evaluation on Oracle platform.

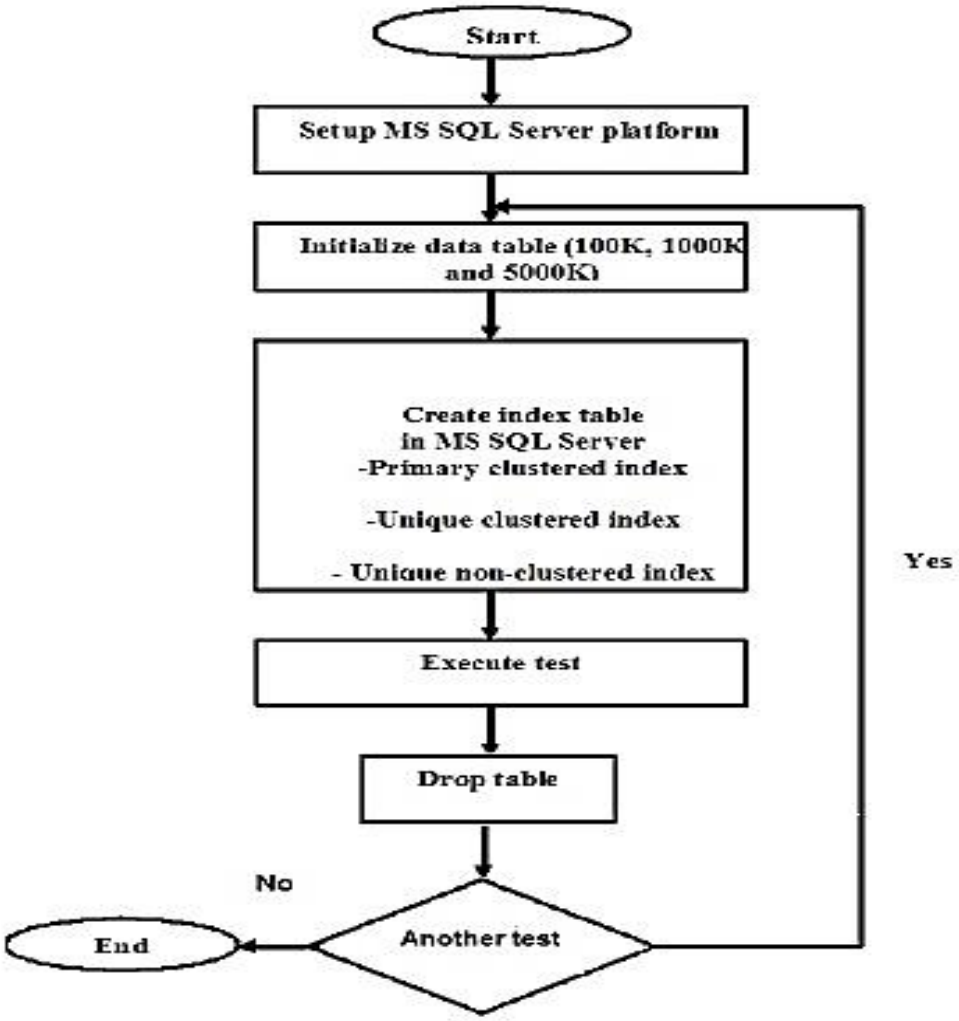

Figure 2 Flowchart of the index evaluation on MS-SQL Server platform.

Experiments and Test Results

This chapter describes the tests for selecting the indexing techniques in detail and presents the results measured. We have evaluated the experimental tests through measuring the performance (response time) of SELECT operation over Oracle 10g and MS SQL Server on different data table sizes (5000K).

SELECT Performance

The SELECT test measured the performance on four different types of SELECT statements:

Test a. Single-row SELECT performance —in this test, each SELECT statement retrieved a single row.

Test b. Rangie SELECT performance —in this test, each SELECT statement retrieved a number of rows depending on the condition in the SELECT statement.

Test c. Fullii scan SELECT performance —in this test, each SELECT statement retrieved all rows.
Test d. Single-row SELECT performance —in this test, each SELECT statement retrieved non row.

Test 3: Oracle 10g and 5000K

Test 3-a: Single-row SELECT performance

The single-row SELECT test has been conducted on Oracle 10g platform with 5000K as table size. As shown in Table 3.3 the performance of Organization index is less costly compared with the other indexing techniques including B-tree (Unique index), Reverse index, and Bitmap index. The Organization index consumes less response time (2 ms) than B-tree (3 ms), Reverse index (3 ms) And Bitmap index (3 ms).

Test 3-b: Range SELECT performance

In this test, we have performed a range SELECT experiment on STUD_GRADE attribute. This attribute has five values A, B, C, D and F. Table 3.3 illustrates that the Bitmap index and Reverse index have the same response time (1840 ms).

Test 3-c: Full scan SELECT performance

The full scan test has been conducted on Oracle 10g platform with 5000K as table size. As shown in Table 3.3, the performance of B-tree and Reverse indexes are relatively the best compared with the other indexing techniques including Bitmap, Organization indexes.The B-tree and Reverse indexes consume less response time (1840 ms) than Bitmap (1880 ms) and Organization index (3220 ms). Another full scan test has been conducted on Oracle 10g platform with 5000K as table size. The data attribute is STUD_GRADE in the SELECT statement. Figure 4 shows the full scan SELECT performance with 5000K on Oracle 10g. The Bitmap index and Reverse index consume the same response time (1840 ms).

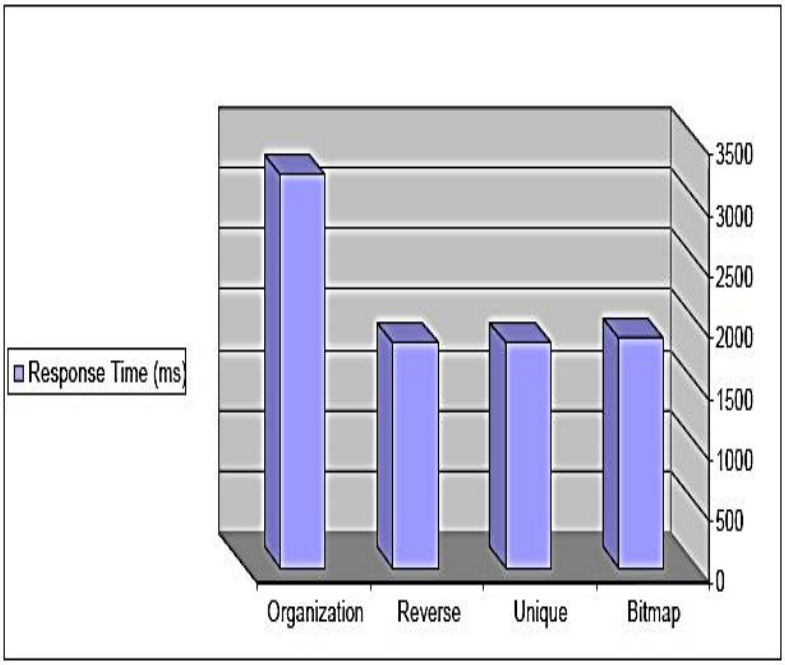

Figure 4 Full scan SELECT executions time for different indexes with 5000K on Oracle 10g

Test 3-d: Single-row SELECT performance with non-row

This test has been conducted on Oracle 10g platform with 5000K as table size. The data attribute is STUD_ID in the SELECT statement. The indexing techniques (such as Bitmap index, B-tree, Reverse index and Organization index) approximately have the same response time (3 ms).

Test 6: MS SQL Server and 5000K

3.2.6.1 Test 6-a: Single-row SELECT performance

The single-row SELECT test has been conducted on MS-SQL Server with 5000K as table size. As shown in Table 3.6, the Primary Key Clustered and B-tree (Unique Non-clustered) consume the same response time (15 ms). The Unique Clustered consumes (16 ms) to complete the scan. In non-clustered index, the data row has first to be located by using an Index Seek operation with the non-clustered index, followed by Nested Loops with a RID Lookup to extract the set of selected columns that are not a part of the non-clustered index, as illustrated in Figure 6.

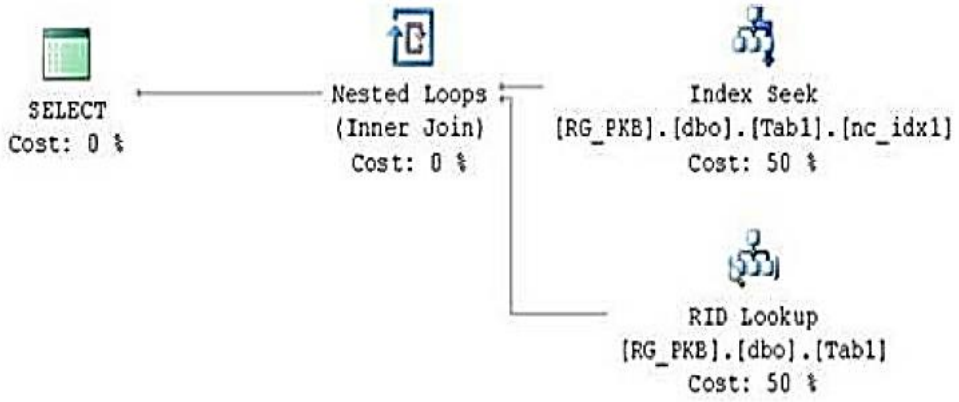

Figure 6 Query execution plan for SELECT statement on table with non-clustered index in MSSQL Server

Test 6-b: Full scan SELECT performance

The full scan test has been conducted on MS SQL Server with 5000K as table size. As shown in Table 3.6, the performance of Primary Key Clustered is relatively faster than B-tree (Unique Nonclustered) and Unique Clustered. The Primary Key Clustered consumes less response time (16 ms) than B-tree and Unique Clustered (17 ms).

3.2.6.3 Test 6-c: Single-row SELECT performance with non-row

As illustrated in Table 3.6, the Primary Key Clustered, Unique Clustered and B-tree (Unique Non-clustered) consume the same response time (18 ms).

Recommendation

After conducting these tests using our sample database table (5000K), we have made the following general observations and recommendations. Thus, it is suggested that Database DBA's are encouraged to use them as standard recommendations only and validate the applicability of the results to certain target scenario. In Oracle with size: 5000K, the performance of Bitmap index, B-tree and Reverse index are much better than Organization index especially at full scan select performance. In MS-SQL Server with size 5000K, the performance of Primary Key Clustered is relatively better than B-tree (Unique Non-clustered) especially at full scan select performance. Some technical studies indicate that using index in the retrieval systems over Oracle platform consumes greater response time than MS-SQL Server. Thus, those studies support our results in this thesis. B-tree is the common between MS-SQL Server and Oracle.

References

[1]. Burleson C., (2010), '*Bitmapped Index Usage'* , viewed at 1 December 2010, <http://www.remote-dba.net/t_grid_rac_bitmapped_index_usage.htm> .

[2]. Lin D., Jensen C., Ooi B., and Saltenis S., (2005), 'Efficient indexing of the historical', present and future positions of moving objects. In MDM, pp. 59–66.

[3]. Graefe G., (2010) ,'A survey of B-tree locking techniques' , *published in ACM*

[4]. *Transactions on Database Systems (TODS)*, Volume 35, Issue 2. 88

[5]. Choi I., Bongki M. and Hyoung J.,(2007), 'A clustering method based on path similarities of XML data', *Data & Knowledge Engineering 60* , pp. 361–376.

[6]. Lo J., Barroso L., Eggers S., Gharachorloo K., Levy H., and Parekh S., (1998), 'An analysis of database workload performance on simultaneous multithreaded processors', in ISCA'98: *Proceedings of the 25th annual international symposium on Computer architecture*, pp. 39–50.

[7]. King J., (2001), ' ORACLE8I INDEXING CHOICES: BEST OF BREED', Designing, Developing and Deploying Applications, viewed 20 June 2010, <http://www.kingtraining.com/confdownloads/downloads/O8index_paper.pdf>.

[8]. Zobel, J., Moffat, A. and Ramamohanarao, K. (1996), 'Guidelines for Presentation and Comparison of Indexing Techniques'. *SIGMOD Record*, Vol. 25., PP 10-15.

