A Database Optimization Model with Quantitative Benchmark

J A D C Anuradha Jayakody
Faculty of Computing
Sri Lanka Institute of Information Technology
New Kandy Road, Malabe, Sri Lanka
anuradha.j@sliit.lk

Iain Murray
Department of Electrical and Computer Engineering
Curtin University
Perth, Western Australia
i.murray@curtin.edu.au

Johannes Herrmann
Department of Computing
Curtin University
Perth
Western Australia
hannes.herrmann@curtin.edu.au

Shashika Lokuliyana
Faculty of Computing
ISE Department
Sri Lanka Institute of Information Technology
New Kandy Road, Malabe
Sri Lanka
shashika.l@sliit.lk

D.S.A.Kandawala
Faculty of Computing
ISE Department
Sri Lanka Institute of Information Technology
New Kandy Road, Malabe
Sri Lanka
Sach7691@gmail.com

S.E.C.Nanayakkara
Faculty of Computing
ISE Department
Sri Lanka Institute of Information Technology
New Kandy Road, Malabe
Sri Lanka
elainnj@gmail.com

***Abstract*— Query optimization and indexing have an immense impact on database optimization. This has been considered in many different perspectives which provide several different solutions in each case. The purpose of this paper is to primarily provide a comprehensive review and discussion of the core problems through the research on query optimization technology and indexing, based on a number of optimization techniques commonly used in the general approach of a query. A new database optimization model is designed and experiments show that this model can significantly reduce the amount of query execution time to improve the optimization efficiency. Regardless of research on the database optimization model, significant work has been done to comparatively evaluate three scenarios; non-optimized query, optimized query without following any optimization standard, and optimized query via proposed optimization model using the same experimental methodology. Further, this paper describes a benchmark that was developed specifically for the purpose of measuring the quantitative evaluations of the proposed database optimization model.**

Keywords— database optimization; query optimization; indexing; benchmark; quantitative evaluation

I. INTRODUCTION

The research of optimizing database systems has been an ongoing process where Database Management Systems (DBMS) have been evolving rapidly [1]. With the enhancing and growing applications, data queries are increasingly complex, and thus the need for efficiency requests are increasingly high. Due to that, processing optimized queries is a key concern when it comes to optimizing a database management system. To find what factors affected the efficiency of database queries, and to come up with a proper Database Optimization Model which provides necessary instructions to follow in different situations in a database management system, little research has been conducted based on the following: Query Optimization and Indexing [2],[3], [15].

Based on the experimental Results of testing the proposed Database Optimization Model, a new benchmark has been introduced which calculates the maximum number of queries that can be executed simultaneously on a database. Web services are used for the purpose of storing data. In simple terms, the database resides in the cloud, so that it has the centralized access from any location and also fault tolerance,

backups and other security aspects are no longer a burden to the user.

This paper is organized as follows: Section II illustrates work related to this research area. The methodology of the conducted research is explained in Section III. The findings and evidence of the research are presented in Section IV. Finally, the conclusion and the future works are stated in the final part.

II. LITERATURE SURVEY

According to Fan Yuanyuan, MiXifeng on "Distributed Database System Query Optimization Algorithm Research", a new algorithm is designed through the research on query optimization technology, based on a number of optimization algorithms commonly used in distributed query, a new algorithm is designed, which can significantly reduce the amount of intermediate result data, effectively reduce the network communication cost, to improve the optimal efficiency [2]. This is a research on producing an algorithm to improve the efficiency of database by using query optimization, but the research is not focusing on the use of indexing.

Lin Hong, Zhuhai, Mingda Lu and Weiting Hong on "A Business Computing System Optimization Research on the Efficiency of Database Queries", discuss the performance evaluation, measurement and business computing system optimization based on some experiment researches on the efficiency of database queries [3] and not considering about the use of indexing.

Ivo Jimenez, Jeff LeFevre et-al. on "Benchmarking Online Index-Tuning Algorithms", they outline the development of a performance benchmark for the specific problem of online index tuning [4] and not considering about the query optimization techniques.

Surajit Chaudhuri, Vivek Narasayya on "Self- Tuning Database Systems: A Decade of Progress", propose advances in self-tuning database systems over the past decade and it primarily focuses on the problem of automated physical database design [5].

P. Karthik, G. Thippa Reddy and E. Kaari Vanan on "Tuning the SQL Query in order to Reduce Time Consumption", present a new query optimization method which gives the high performance of the system and less stress on the database when data transmission occurs and the efficient usage of a database engine with reduced memory requirements [6].

According to the above literature reviews which are related to database optimization in the area of rational databases, there is still a need for further improvement in the field of both the indexing and parallel query optimization parallel. This paper has achieved optimization output using both the techniques when it comes to data retrieval for any real time application.

III. RESEARCH METHODOLOGY

Even though there are a vast number of fields which relate to Database Optimization, this research is focused on Query Optimization and Indexing [7]. As the previous section explained, even though plenty of query optimization and indexing techniques are available recently [16], [17], due to situational dependencies of above mentioned optimization techniques, still there is a possibility of getting an unnecessary burden and as well as an added risk, when these optimization techniques are used inappropriately [9],[10],[11]. As a solution for this research problem, a convenient Database Optimization Model was introduced, which will provide a precise standard to follow up when applying optimization techniques mostly related to query optimization and indexing [12],[13],[14].

Queries have been tested on three main scenarios in order to evaluate the proposed database optimization model which is shown in (Fig.2) to prove that it is productive and practical in real-time environment. These queries have been tested and simulated using the phpMyAdmin Graphical User Interface (GUI) software. MYSQL is run on Intel® Core™ i7, 8GB RAM.

Scenarios can be sequentially discussed as follows.

- Scenario1 – Tested results with a Non-Optimized query
- Scenario2– Tested results without following any optimization Model
- Scenario3 – Tested results for the Proposed Database
- Optimization Model

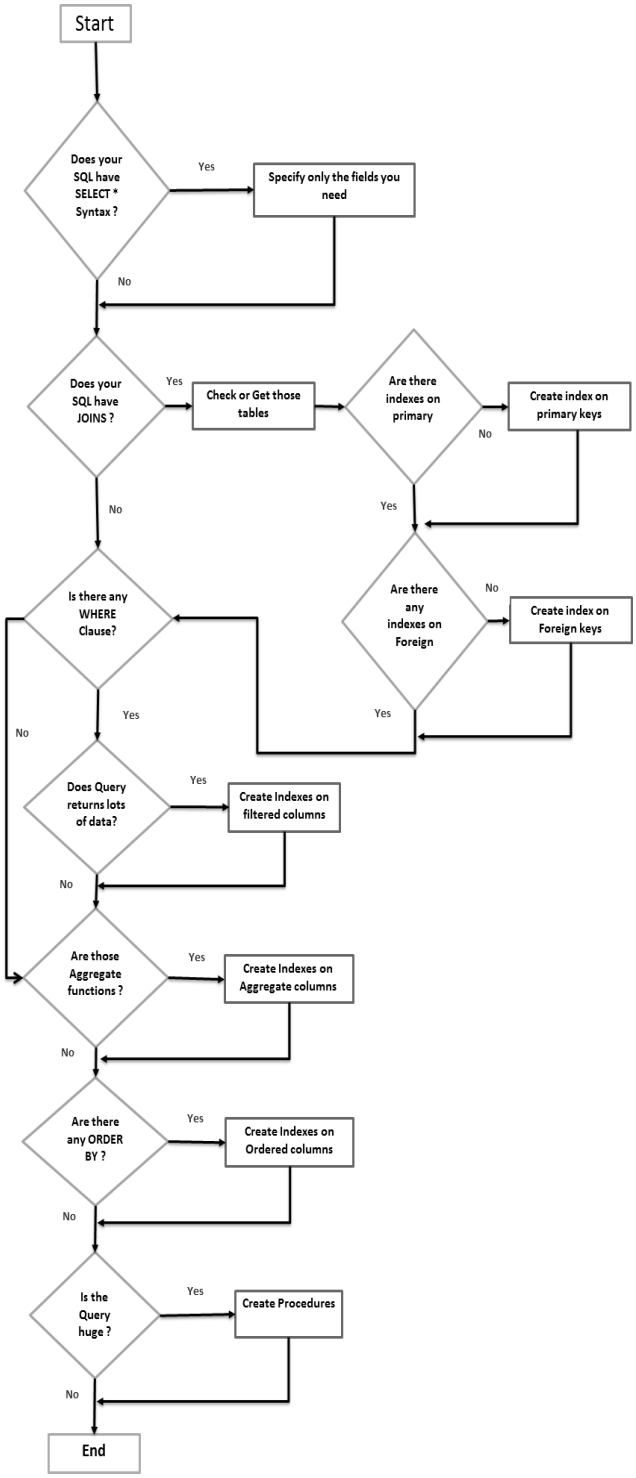
With respect to above mentioned scenarios, the query can be generated as following.

```
SET profiling = 1;
SELECT donors.name, products.name, donations.quantity
FROM donors, products, donations
INNER JOIN donations ON donors.id =
donations.donor_id
INNER JOIN products ON products.id =
donations.product_id WHERE products.category_id = 1;
SHOW profiles;
```

Fig .1. The query used for the evaluation proposed three Scenarios

According to the query specified in above figure (Fig.1), each and every query line can be described in order to satisfy the proposed database optimization model (Fig.2)

1. Enable profiling by setting it to 1.
2. Select the donor name, donation quantity and product name from the donors, donations and products table.
3. Join both the donations and donors table using the corresponding ids.
4. Join both the products and donations table using the corresponding ids, where product id equal to1.
5. Displays a list of the most recent statements sent to



the server.

Fig. 2. Proposed Database Optimization Model

In order to test the three scenarios, 1500 records have been used from three main tables namely, Donors table, Donations table and Items table. One-blood System has been used for the purpose of testing in real-time environment. The test results were obtained with respect to the number of queries that were executed simultaneously by the database. Hence in each scenario the test results were taken while changing this factor (number of queries executed simultaneously from 1 to 10. Profiling has been used to measure the execution time of the queries that were executed concurrently. Typically the query execution time is a very small value which comes in milliseconds. Hence in order to obtain a considerable difference between these three tested scenarios, the frequency has been calculated by taking one over execution time.

$$\text{Frequency} = 1/\text{Execution Time} \rightarrow (1)$$

The tasks that followed in this phase can be summarized as below;

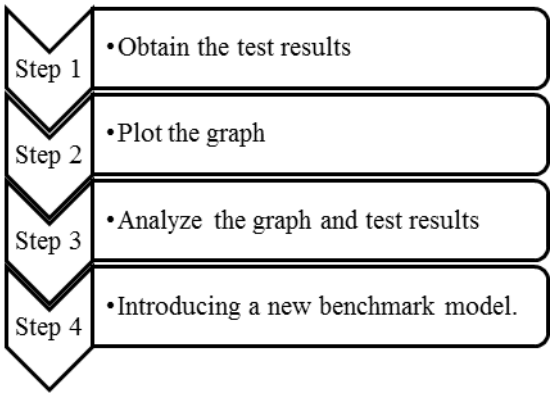


Fig. 3 The Major Tasks accomplished in the methodology

IV. Research findings & Discussion

The following broad analysis of the test results will prove that the proposed Database Optimization Model is fruitful in real-time environment. Further justifications along with the qualitative and quantitative evaluations have been performed on three scenarios as mentioned earlier.

Following table 1 shows the three scenarios, with respect to y-axis. Normally the execution time produced by the database is a small value. To take a considerable value gap between execution time, (Eq. 1) has been used to produce the frequency which resembles execution time. Sequentially the scenarios for the y-axis can be described as following.

Y1-Test result with query optimization procedure Frequency
Y2-Test result without query optimization procedure Frequency
Y3- Non optimized query Frequency

Relevant frequency calculations for the above mentioned scenarios can be tabulated as follows.

TABLE I
TEST RESULTS FOR THE COMPARISON BETWEEN THREE SCENARIOES

No.of occurrences (x)	Test result with query optimization procedure Frequency (y1)	Test result without query optimization procedure Frequency (y2)	Non optimized query Frequency (y3)
1	2074.69	783.09	781.25
2	1219.51	483.33	443.66
3	766.28	354.48	313.28
4	634.12	241.84	240.96
5	558.66	222.52	195.01
6	508.13	183.28	162.73
7	422.48	153.12	140.61
8	382.26	139.63	124.39
9	343.64	126.89	113.42
10	324.46	111.74	103

According to the test results corresponding chart can be shown as following.

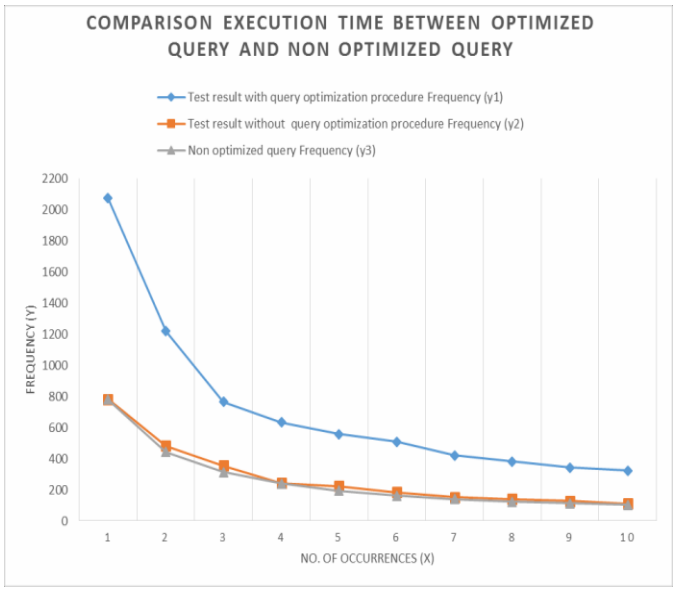


Fig. 4 Chart for the Comparison between three Scenarios

The x-axis of the above chart represents the number of queries that were executed simultaneously by the database (no. of occurrences). The y-axis represents the frequency of the relevant query set. This chart illustrates that the proposed Database Optimization Model provides the best results.

According to the chart, it was identified that the plotted graphs based on the practical values, inherit the properties of a hyperbola. Hence it is legitimate to say that n – number of occurrences approaches infinity when the f – frequency approaches to zero.

Based on these assumptions and concepts, the conducted research has been led towards introducing a new benchmark which will calculate the maximum number of occurrences when the frequency approaches the minimum. Major factors for this equation are as follows,

T_D – Time taken to select the database
T_N – Time taken to execute one query
f – Frequency of executing the query
n – Number of occurrences

Based on the analytical values few assumptions have been made as follows,

- T_D is a constant which proves that logically as well, since it does not matter how well the query is optimized, still the time that will take to select the database will be the same.
- T_N has a constant value within a particular scenario, but varies across different scenarios.

The way of deriving the benchmark model can be illustrated as follows,

$$T_{total} = n.T_N + T_D$$

$$f_{total} = \frac{1}{T_{total}}$$

$$f_{total} = \frac{1}{(n.T_N + T_D)}$$

$$n = (\frac{1}{T_N}(\frac{1}{f} - T_D))$$

$$\lim_{f \to 1}(n) = \lim_{f \to 1}(n) \quad (\frac{1}{T_N}(\frac{1}{f} - T_D))$$

Since n goes to infinity when f goes to 0, to obtain an exact value for n, limits has been applied to the final equation while f goes to 1. By deriving this equation final result would be the newly introduced benchmark that calculates the maximum number of query executions that can be occur simultaneously when having the minimum frequency.

According to derivation the simplified benchmark equation can be present as following.

$$\lim_{f \to 1}(n) = \left(\frac{1 - TD}{TN}\right) \longrightarrow (2)$$

In order to prove that the newly invented benchmark provides accurate results, it has been tested and verified under practical values in a real-time test scenario. With the use of this newly introduced benchmark, the proposed database optimization model has been evaluated with quantitative measures as follows. The below calculation shows the maximum number of occurrences that can be calculated in the second scenario. (T_n – Constant T_n value for the second scenario)

$$T_D = 0.000154$$

$$T_n = 0.000921$$

$$\lim_{f \to 1}(n) = \left(\frac{1 - TD}{T_n}\right)$$

$$\lim_{f \to 1}(n) = \left(\frac{1 - 0.000154}{0.000921}\right)$$

$$\text{Max.} \quad n = 1074$$

The following calculation shows the maximum number of occurrences that can be calculated for the proposed database optimization model. (T_n – Constant T_n value for the first scenario)

$$T_D = 0.000154$$

$$T_n = 0.00031$$

$$\lim_{f \to 1}(n) = \left(\frac{1 - TD}{T_n}\right)$$

$$\lim_{f \to 1}(n) = \left(\frac{1 - 0.000154}{0.00031}\right)$$

$$\text{Max.} \quad n = 3225$$

From the above calculations, it is confirmed that the proposed database optimization model provides the best results out of the rest.

Based on these evaluations it is confirmed that the proposed Database Optimization Model provides an accurate, efficient and effective solution for the identified research problem.

V. Conclusions & Future Work

The proposed Database Optimization Model is to perform Query Optimization and Indexing in accordance with the query procedures that differ from one situation to another. The advantage of the proposed model is that it is simple to process and gives out the best performance as the model instructs how to use the query optimization and indexing techniques based on the approach of the query to be optimized. The deficiency is that the model does not consider how to optimize the stored query procedures further to reduce the network communication costs. This paper designs a new database optimization framework along with a benchmark model, which calculates the maximum number of queries that can be executed simultaneously by the database, and also which defines a function to determine the optimization benefits of the proposed database optimization model. Experimental results show after comparing with the general optimization techniques without following a proper optimization standard, the proposed database optimization model has higher optimization efficiency, significantly reduces the amount of intermediate result data, and eventually reduces the total cost of the network communications.

For database optimization, previous researchers have been demonstrated many query optimization procedures according to their vision. Unfortunately, there are not proper ways to apply them in the real-time environment. Nevertheless, this proposed benchmark model gives an opportunity to apply the concept to the real world scenarios. This model is mainly focused on scenarios where data retrieved from the database using the SELECT statement. For further enhancements the optimization models can be introduced for INSERT, UPDATE and DELETE statements as well.

Acknowledgment

This work has been supported by the Curtin University, Perth, Western Australia and Sri Lanka Institute of Information Technology, Malabe, Sri Lanka.

References

[1] Ong Chun Lin et al., “Database Optimization for Novelty Detection”, School of Electrical & Electronic Engineering, Nanyang Technological University, Singapore, 2009.

[2] Fan Yuanyuan and MiXifeng, “Distributed Database System Query Optimization Algorithm Research”, Department of Computer and Information Engineering, Jiaozuo Teachers College, 2010.

[3] Lin Hong et al., “A Business Computing System Optimization Research on the Efficiency of Database Queries”, in International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, Int. Conf., 2013.

[4] Ivo Jimenez et al., “Benchmarking Online Index-Tuning Algorithms”, University of California Santa Cruz, 2011.

[5] Surajit Chaudhuri and Vivek Narasayya, “Self-Tuning Database Systems: A Decade of Progress”, Microsoft Research.

[6] P. Karthik et al., “Tuning the SQL Query in order to Reduce Time Consumption”, IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 4, No 3, School of Information Technology and Engineering, VIT University Vellore, Tamil Nadu, India, July 2012.

[7] Maxim Martynov and Boris Novikov, “An Indexing Algorithm for Text Retrieval”, University of St. -Petersburg, Russia, 1996.

[8] “Data Profiling”, 2015. [Online]. Available: https://en.wikipedia.org/wiki/Data_profiling.

[9] LiWu Chang, Ira S. Moskowitz, James Tracy, "An Agent-based Approach to Inference Prevention in Distributed Database Systems",2003.

[10] S.F. Rodd, Dr, U.P. Kulkarni, "Adaptive Tuning Algorithm for performance Tuning of database Management System,” 2010.

[11] Hitesh Kumar Sharma, Aditya Shastri, Ranjit Biswas, "Architecture of Automated Database Tuning Using SGA parameter", 2012.

[12] David J. Montana and Lawrence Davis, Training Feedforward," Neural Networks Using Genetic Algorithms",1994.

[13] S. Agarwal and et.al, “Automated selection of materialized views and indexes”, VLDB, 2007.

[14] K. Schnaitter, “On-line index selection for physical database tuning,” Ph.D. dissertation, University of California Santa Cruz, 2010.

[15] J.A.D.C.Anuradha Jayakody, Iain Murray, Johannes Herrmann “Database modelling for vision impaired indoor navigation systems",2015.

[16] Jingbo Shao,"Database Performance Optimization for SQL Server Based on Hierarchical Queuing Network Model. International Journal of Database Theory and Application",2015.

[17] Chaudhuri,"An Overview of Query Optimization in Relational Systems",2014.