Query Processing for Time Efficient Data Retrieval

**Muhammad Qasim Memon, Jingsha He, Aasma Memon, Khurram Gulzar Rana,
Muhammad Salman Pathan**
School of software Engineering, Beijing University of Technology, 100124, Beijing, China
Beijing Engineering Research Center for IoT Software and Systems, Beijing University of Technology, 100124, Beijing, China

Article Info

Article history:

Received Oct 27, 2017
Revised Jan 30, 2018
Accepted Feb 17, 2018

Keywords:

Database tuning
Query execution
Query processing

ABSTRACT

In database management system (DBMS) retrieving data through structure query language is an essential aspect to find better execution plan for performance. In this paper, we incorporated database objects to optimize query execution time and its cost by vanishing poorly SQL statements. We proposed a method of evolving and inserting database constraints as database objects embedded with queries either to add them for the sake of transactions required by user to detect those queries for the betterment of performance. We took analysis on several databases while processing queries itself and assimilate real time database workload with the bunch of transactions are invoked in comparison with tuning approaches. These database objects are coded in procedural language environment pertaining rules to make it worth and are merged into queries offering improved execution plan.

Corresponding Author:

Muhammad Qasim Memon,
School of software Engineering, Beijing University of Technology, 100124,
Beijing, China.
Email: memon_kasim@yahoo.com

1. INTRODUCTION

For optimizing database, it is necessary for performance to tune queries. Although it is also important as compare to optimize the other phases of database server installation such as configuration of software and hardware causes the performance as well. If database server manipulated with eminent powerful hardware that have all i/o resources with processing speed with at least one bad query pertaining poor performance affects the speed of execution [1]. It is distinctively stated about any database server after optimizing all hardware configurations, remaining issues may appear to lessen performance of database those are: expensive and time consuming queries, reminiscence of indexes, inappropriate statistics, debugging of queries and uses of cursors [2]. In this work, we focus on two important phases of tuning queries. First, to tune poorly performing queries such as analyzing execution plan, optimizing execution plan and its estimated cost. Secondly, inclusion of database constraints to restrict or acquire data from the database based on defined rules. In this work, our focus is to evolve procedural language functionalities invoked in runtime environment through queries rather relying on modifying SQL structure, access design amendments and employing local constraints within queries. There are many databases and tools offers tuning into to two ways. First, tuning offered by commercial databases known as automatic and externally tuning by venders tool into to optimize queries known as manual tuning. This simplifies our comparison analysis resembles between both types of tuning approaches. We devote our method of deploying this tuning comparison with a bunch of queries in several databases. Internal tuning offered by databases like oracle, MySQL, DB2 named as automatic tuning, and external tuning named as SQL tuning offered by different vendors tools into databases like Microsoft SQL Server developer 8.0 and Oracle SQL developer. This simplifies our comparison analysis resembles over automatic and SQL tuning. We devote our methods of deploying two phases of tuning as we mentioned above, and bunch of queries in different commercial databases [3].

Rest of paper presented as, in the next phase current work and recent optimization approaches are highlighted. In third phase database schema, structure and query processing and proposed procedures will be discussed. Fourth phase of this paper contributes analysis of each commercial database compared with both types of tuning approaches discussed earlier. Finally we elaborate findings and conclusion of our proposed framework.

2. RELATED WORK

Cost effective performance based upon distributed query plan for optimizing queries to generate execution plan by comparing Teaching-Learner Based Optimization (TLBO) and generic algorithms [4]. To enhance performance database objects are deployed to make better execution plan [5]. Procedure language like PL/SQL make an impact in creating and implementation of programs for finding association and sequential pattern in data and enlightened most of database associated applications are discussed, and proposed tuning utility and benchmark to tune PL/SQL queries through integrating database objects [6]. Performance tuning of modern databases using operation research techniques [7], query driven approach improves efficiency of query processing through amendments in database structure. Recently powerful processing units are deployed such as GPUs and FPGAs by adhering parallel processing units and making efficient optimizer by utilizing execution models to make query processing efficient and distributes workload on several processing units. There are many techniques related to query optimization in relational databases even on large scale of databases taken into consideration persistently [8]. Query optimizer is a main part of database management system that manipulates in different ways for better execution plan. Data administrator attempts to optimize database and generate optimal execution plan for poor queries. Apparently, database vendors are offering automatic features to minimize human effort. Although many researcher find enough space with major consideration [9, [10], by remaking and processing of query optimization in several ways such as optimizers are identified for their automatic features exploits pros and cons and produces way of promoting the current position of automatic computing. Learning based models are proposed in [11, [12]. Online transactional processing analysis for the processing of complex and join queries [13], author has proposed cost based query optimization in multiprocessing environment.

3. RESEARCH METHOD

Rational database system conceptually defines the logical structure of database, data definition to measure its scalability and performance like data consistency, data manipulation and data control. In this work, different rational database like Oracle, MySQL and DB2 is equipped to run all queries against retrieving data. This requires to differentiate several databases as we make changes in queries at same level we see how much different occurs when deploying several commercial databases. Among those databases systems, we seek to acquire best execution plan and estimated cost of provided queries through automatic and manual tuning approaches.

3.1. Queries

a. List the student and sort their CPN in ascending order from district of Hyderabad.
b. List student who are from rural areas, are graduated in from rural areas and Hafiz e Quran as well.
c. List those student who are fresh
d. Provide those candidates who are in minorities and have less than 1000 dollars salary of their parents.

3.2. Constraints

a. CPN derives as total aggregated calculated score.
b. Hafiz e Quran student provided with bonus of .2 in their CPN score.
c. Rural district areas student have 0.1 bonuses in their CPN.
d. Student who are not fresh or attempt twice for admission, deduct 0.2 score in their CPN.
e. Minorities student are considered from rural areas, they have 0.2 score bonus in their CPN.

Procedure and rule may be written in pseudo code form.

```
Create or replace procedure_rule
  When (rule description)
   Begin
   If
  Inserting/updating (condition/call procedure)
   Select table_attribute(s)
  From table; //SQL Query
  End
End procedure_rule;
```

```
Create or replace procedure_attribute Is
  If
  Constraint violation; //invoking integrity constraints
  Procedure_rule; //calling local constraints
End
```

4. RESULTS AND ANALYSIS

In this paper, to our best knowledge we Embedded database constraints to fetched data based upon the specified rules in the form of databsase procedures, function and triggers by making the better execution plan and cost based optimization [14]. First we continue to tune queries in commercial databases with their features and optimizers provided within it. We took analysis on poorly performing queries by imparting resistance to merge procedural language features and where ever it is required either by adhering constraints as required.

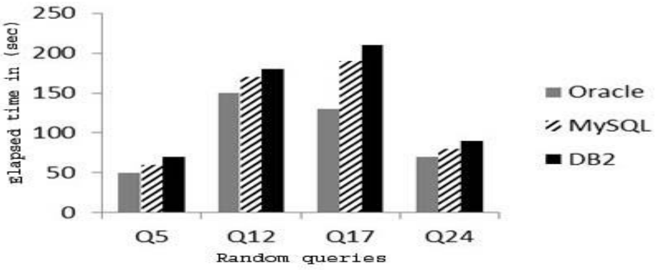

Figure 1. Random queries execution time in different databases

Secodly we did comparison queries execution plan which are improved in estimated cost and execution time in commercial tools compared to tuning approcaches offered by database itself. These queries executed in several databases just to make worth the type of databases ustilized such as oracle, MySQL, DB2 has exploited magnificent changes over tuning as with real time workload and draws attention to those queries that are optimized adhering proposed functionalities within them in sophisticated manner [15]. Table 1, shows the comparsion of query execution time and estimated cost among all databases used in this paper. Oracle database fetched the worst execution time of 150 seconds with less number of rows whereas DB2 takes 30 seconds exceeds 30 seconds futher while retrieving same number of rows. The minimum amount of time required to execute a query of 50 seconds in Oracle database consuming .055 estimated costs. The maximum amount extimated cost was 0.55, conceived by DB2 database while retrieved 300 rows as shown in Figure 3. Estimated cost and execution time for four queries have shown in Figure 1 and Figure 2, higher the estimated cost consist while executing queries conceding more execution time [16]. We tried our best to optimize cost and time execution of real time workload. Figure 3 depicts number of rows fetched by some of quereis having optimal. Figure 4 shows the total number of queries with their execution time in different databases exhibit variation with respect to execution time of each query. We imparted some of queries without constraint and some of them are embedded with database constraints and difference of time of each query is relevant in every type of database.

Table 1. Comparison of Query Execution Time and Extimated Cost in Each Database

Database Server	Best Query Execution Time	Worst Query Execution time	Maximum Estimated Cost	Minimum Estimated Cost
Oracle	50	150	0.327	0.055
MySQL	60	190	0.431	0.066
DB2	70	210	0.55	0.077

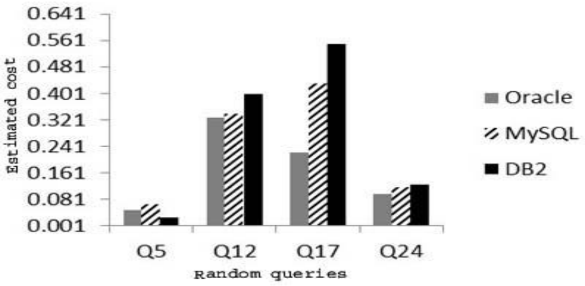

Figure 2. Random queries Estimated cost in different databases

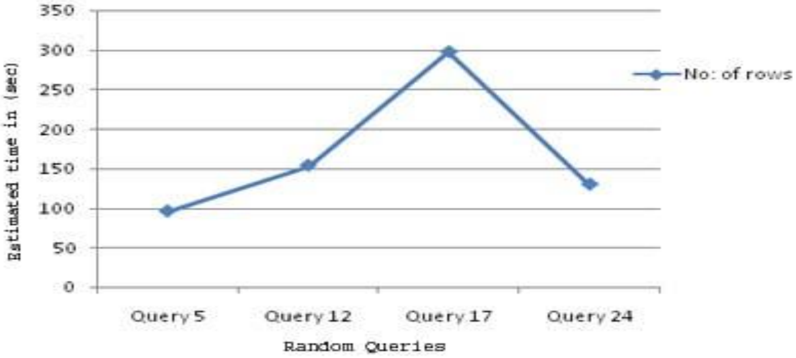

Figure 3. Number of rows retrieved by random queries

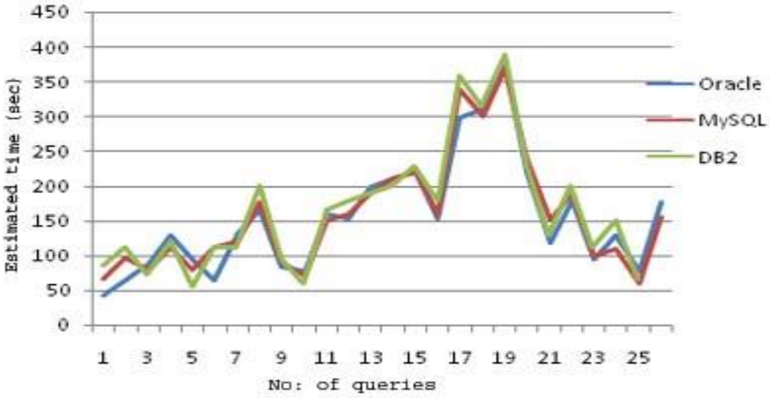

Figure 4. Execution time for each query in different databases

5. FINDINGS AND CONCLUSION

We focus on the result oriented analysis, query optimization based on manual tuning on three commercial databases. We eradicated some of queries and focus on those which are manipulated with proposed procedures, functions and triggers and embedded as in shape of database constraints. We refer constraint as to restrict database to defind specific rules so that we can retrieve data at ease. In this paper, we utilized three commercial databases and used assimlite with workload of more than three thounsands records and twenty seven queries. We presend our proposed databased constraints as they embedded within queries, run them against database and highlighted queries execution time and estimated cost considering either some of queries only tuned by optimizers of specific database type or some of them are tunned manualy by manipulating with database constraints. To our best knowledge, optimization of queries can be considered more effiecient if database objects are programmed functionally well enough as required to retrieve the number of rows and optimized elapsed estimated cost with better execution plan.

ACKNOWLEDGEMENTS

The work in this paper has been supported by National Natural Science Foundation of China (61272500), National High-tech R&D Program (863 Program) (2015AA017204) and Beijing Natural Science Foundation (4142008).

REFERENCES

[1] Susana and suharjito. Query optimization using fuzzy logic in integrated database. *International Journal of Electrical Engineering and Computer Science (IJEECS)*. 2016; 4(3): 637-642.

[2] Corlatan CG, Lazar MM, Luca V, Petricica OT. Query Optimization Techniques in Microsoft SQL Server. *Database System Journal.* 2014; 5(2): 33-48.

[3] Sharma P. Retrieval of Information Using Fuzzy Queries. *International Journal of Engineering and Techniques*. 2016; 2(3).

[4] Barbhuiya SZ, Nandi MG. Query Optimization Using a Genetic Programming Approach. *ADBU-Journal of Engineering Technology*. 2016: 158-163.

[5] V.ishra, V. Singh. *Generating optimal query plan for distributed query processing using teacher learner based optimization*, Procedia Computer Science (IMCIP) 2015; 281-290.

[6] M. M. Qasim, H. Jingsha, Aasma, A. Ditta, K. G. Rana. Dynamic integration of PL/SQL for complex queries. *International Journal of Database Theory and Application*. 2016; 9(10): 351-362.

[7] L. Dandan, H. Lu, D. Yi. SQL Query optimization methods for rational database system. *International Conference on Computer Engineering and Application*, 2010; 557-560.

[8] F. Sun and L. Wing. P*aging query optimization of massive data in oracle 10g database.* Computer and Information Science and Service System (CSSS) IEEE Conference. 2011.

[9] H. Herodotou, N. Borisov and S. Babu. *Query optimization techniques for partitioned tables.* ACM SIGMOD International Conference on Management of data. 2011.

[10] Donald Kossmann and Konrad Stocker. Iterative dynamic programming: a new class of query optimization algorithms, *ACM Transactions on Database Systems*. 2000; 25(1): 43–82.

[11] M. Khan and M. N. A. Khan. Exploring query optimization techniques in relational database, *International journal of Database Theory and Application*. 2013; 6(3).

[12] Rimma V. Nehme, Karen Works, Chuan Lei b, Elke A. Rundensteiner and Elisa Bertino. Multi-route query processing and optimization. *Journal of Computer and System Sciences*. 2010; 312–329.

[13] T. David, D. Gillian, L. Cheng. *Implementation of data mining techniques using oracle 8i PL/SQL*. Proceeding of Artificial Neural Networks in Engineering Conference. 2002; 12; 471-476.

[14] Andrew N. K Chen. Robust Optimization for database tuning for modern database systems, *European. Journal of Operation Research*, 171; 412-429.

[15] Hemalatha Gunasekaran and Thanushkodi Keppana Gowder. *The interntional Arab journal of Information Technology*. 2015; 12(6).

[16] Donald Kossmann and Konrad Stocker. *ACM Transactions on Database Systems*. 2000; 1: 43–82.