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

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.

Keywords:
Database tuning
Query execution
Query processing

Copyright © 2018 Institute of Advanced Engineering and Science. All rights reserved.

Corresponding Author:
Muhammad Qasim Memon,
School of software Engineering, Beijing University of Technology, 100124, Beijing, China.
Email: memo_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 vendors 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].

Query Processing for Time Efficient Data Retrieval (Muhammad Qasim Memon)

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 database 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](image)

Secondly we did comparison queries execution plan which are improved in estimated cost and execution time in commercial tools compared to tuning approaches offered by database itself. These queries executed in several databases just to make worth the type of databases utilized 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 comparison 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 further 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 estimated 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 queries 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>
<thead>
<tr>
<th>Database Server</th>
<th>Best Query Execution Time</th>
<th>Worst Query Execution Time</th>
<th>Maximum Estimated Cost</th>
<th>Minimum Estimated Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle</td>
<td>50</td>
<td>150</td>
<td>0.327</td>
<td>0.055</td>
</tr>
<tr>
<td>MySQL</td>
<td>60</td>
<td>190</td>
<td>0.431</td>
<td>0.066</td>
</tr>
<tr>
<td>DB2</td>
<td>70</td>
<td>210</td>
<td>0.55</td>
<td>0.077</td>
</tr>
</tbody>
</table>

Table 1. Comparison of Query Execution Time and Estimated Cost in Each Database
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 define 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 thousands records and twenty seven queries. We present our proposed database 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 tuned manually by manipulating with database constraints. To our best knowledge, optimization of queries can be considered more efficient 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