SJIF IMPACT FACTOR: 2.912 CRDEEPJournals
Global Journal of Current Research Chaudhary & Nirula Vol. 3 No. 4 ISSN: 2320-2920
Global Journal of Current Research Vol. 3 No. 4. 2015. Pp. 117-121

Full Length Research Paper

Oracle Environment Optimization for Query Optimization using Fuzzy Logic

[1]Arti Chaudhary and [2]Rajnesh Nirula

Pursuing M.Tech (Information Technology), Adesh Institute of Engineering and Technology, Punjab Technical University, Punjab, India.

[2]Academics Dean and Supervisor (Computer Science), Adesh Institute of Engineering and Technology, Punjab Technical University, Punjab, India.

***Corresponding author: Arti Chaudhary**

Abstract

DBMS are an integral part of any corporate house of the online systems and E-Commerce applications. The performance of these systems is affected by several factors. DBMS plays an important role in terms of performance but data is growing day by day as the need of enterprises growing in this competitive business environment to solve different business problem and this performance enhancement need only quality data from huge bytes of data. Hence to tune database management systems it is necessary to tune database in a cost effective manner for which different optimization techniques have been used. To minimize the cost there is a need to build self-managing database system which can continually monitor performance at the time of any performance bottleneck because of lack of any resource, memory etc. Complete knowledge about each parameter is necessary because optimization of single parameter can be degraded the performance of another parameter. One more thing need to be noticed system should not be over tuned as it may lead to degradation of performance. DBMS should be scalable so that it can perform well at a high load. The objective of this research is to analyze the DBMS, by proactively monitoring the performance indicators like buffer miss ratio, number of active processes and the table's size that are showing signs of rapid growth and initiate control measure using fuzzy control. Oracle is taken for the Experimentation purpose.

Keywords: *Optimization, Oracle Environment*

Introduction

Database performance tuning or database optimization, is the activity of making a database system run faster. Tuning the performance of the system involves adjusting various parameters and design choices to improve its performance for a specific application (Rodd. S. F. *et al.* 2012). Although the Relational Database Management system (RDBMS) has become s de-facto standard, its main strength have to be found in its ease-of-use and querying capabilities, rather than its efficiency in terms of hardware and system overhead. With the constantly growing amount of data being accumulated and processed by companies information systems, database performance issues become more likely.

Systematic tuning follows these steps:

1. *Assess the problem and establish numeric values that categorize acceptable behavior.*
2. *Measure the performance of the system before modification*
3. *Identify the part of the system that is critical for improving the performance called the bottleneck.*
4. *Modify that part of the system to remove the bottleneck.*

Oracle includes the following components:

1. *Oracle Server – The server includes files, processes and memory structures used for processing the SQL statements and to maintain the database*
2. *Oracle Instance – A database instance provides user access to a database. When an instance is started, Oracle database allocates a memory area called the SGA (System Global Area) and starts one or more background processes.*
3. *Oracle Database: The database consists of operating system files that provide the actual physical storage for database information. (ICONS 2007).*

Instance tuning

Instance tuning includes tuning the SGA components. The SGA is shared by Oracle processes; which includes server processes and background processes, running on a single computer. The SGC serves various purposes:

1. *Caching data blocks read from disk*
2. *Buffering Redo Data before writing it to the online redo log files*
3. *Storing SQL execution plans*
4. *Maintaining a database of an enterprise involves considerable effort on part of a Database Administrator (DBA) as it is a continuous process and required in depth knowledge, experience and expertise.* (Wiese. David ,*et al.* 2009).

Understanding database tuning

The concept of database tuning involves many factors beyond the DBS, but understanding what these factors are permits as assessment of its integration in database management. The Operational adequacy and degradation most often come from performance bottlenecks. A performance bottleneck occurs when a database component is assigned an excessive amount of work at a given time, which it is incapable of satisfying adequately. Also, it is possible that these bottlenecks can extend beyond the DBMS and encompass other external components, such as the client application, network communications, operating systems or some combination thereof.

Setting performance targets

Whether you are designing or maintaining a system, you should set the specific performance goals so that you know when to tune. You can spend needless time tuning your system without significant gain if you attempt to alter initialization parameters or SQL statements without a specific goal.

Oracle SQL Tuning goals:

1. *Remove Unnecessary Large-Table Full-Table scans*
2. *Cache Small-Table Full-Table scans*
3. *Verify Optimal index usage*
4. *Materialize your aggregations and summaries for static tables.*

Objective

DBMS are an integral part of any corporate house, the online systems, and E-Commerce applications. The performance of these systems is affected by several factors. The important among them include database size which grows with its usage over a period of time, increased user base, sudden increase in the user processes, improperly or un-tuned DBMS. All of these tend to degrade the system response time and hence call for a system that anticipates performance degradation by carefully monitoring the system performance indicators and auto tune the system.

Database performance tuning is done for following reasons, and either or both can apply for any given situations:

1. *Reduce response time should be lowered for maintaining the efficiency of the application.*
2. *Reduce resource usage: Running any sort of query against a database generates a load on the database server, and the less efficiently that query runs the more CPU and I/O it will require.*
3. *A performance goal is to reduce the physical I/O overhead as much as possible, either by making it more likely that the required data is in memory or by making the process of retrieving the required data more efficient.*

Review of literature

Rodd.S. F, *et al,* 2010 proposed a tuning algorithm based on neural network to estimate buffer cache size based on the trained data set. The correction is applied in accordance with the tuning algorithm so as to scale up system performance. This architecture learns from a training set to fine tune the system and thus it relieves the DBA of the tedious process of tuning the DBMS and also need for an expert DBA. However, the system needs further refinement that takes into account sudden surge in work load and also the neural network training dataset must be derived based on proper database characterization.

Verma Ankit, 2011 stated that for the purpose of optimizing performance in data base systems, a new automated physical design has been proposed. The ability of our automated database to adapt to dynamically changing inputs makes them ideal candidates for proactively monitored and fed as input to the proposed presented results show that the proposed method is effective in improving the query response time for a variety of workload types. So it is self-tuned, automated database system whose main focus is performance optimization. An improvement in the physical design automatically has become the top research field both in academics and industry.

Wise David *et al*, 2009 gives importance of best practice approach that is (SOP+A.C Concept) for autonomic database tuning. This paper presented an architecture called autonomic tuning expert. ATE combined key autonomic computing concept and notation of database tuning SOPs to implement a feedback control loop having minimal human interaction. Here best tuning knowledge have been formed to propose a novel framework for tuning database that is autonomous.

Hitesh Kumar Sharma *et al,* 2012 stated that as the business data always grows from byte, megabyte, tera byte, peta byte, and so on and there is no way to avoid this increasing rate of data. Because of this issue, database tuning is critical part of information system. So tuning a database in database in a cost-effective manner is a growing challenge. The total cost of ownership of information technology needs to be significantly reduced by minimizing people costs. One way of addressing the challenge of total cost of ownership is by making information systems self-managing is the automation of database performance tuning, an automated architecture should be provided for the tuning of the system instead of the DBA to perform all the tuning tasks because he cannot perform calculations within short intervals in order to decrease the value of response time.

Rodd S.F *et al*, 2012 proposed a new tuning architecture based on fuzzy logic is presented, where in the control action is is expressed in linguistic terms. In this system the key performance indicator is fuzzified, appropriate fuzzy rules are employed to estimate the extent required for a few important tuning parameters. After de fuzzification, a control is initiated to scale up the system performance.

Problem formulation

The problem formulated for the research is to tune the major components of the oracle instance that includes the SGA parameters in order to decrease the overall response time for execution over the database.

The parameters of the SGA and PGA components that are considered for performance tuning are library cache hit ratio for shared pool, log buffer space and redo buffer allocation retries for redo-log buffer, buffer hit ratio for the database buffer cache, cache hit percentage for pga, the size of the tables of database and no. of the users.

While in the case of pga-components it is obtained by monitoring parameters pga cache hit percentage so as to calculate the pga aggregate target. Generally, adding PGA memory requires reducing memory for some of the SGA components, like the shared pool for the buffer cache. This is because the overall memory dedicated to the Oracle instance is often bound by the amount of physical memory available on the system. As a result any decisions to increase PGA memory must be taken in the larger context of the available memory in the system and the performance of the various SGA components.

In this, the key performance indicators are fuzzified, appropriate fuzzy rules are employed to estimate the extent of tuning required for a few important tuning parameters. After de fuzzification, a control action is initiated to scale up the system performance. (Wiese David, *et al* 2009).

Working methodology

Fuzzy logic is most suitable choice for many control applications for the fact that fuzzy control systems are robust can be tweaked easily to improve the system performance dramatically and most importantly they are much simpler in design to implement. Moreover, there is no need to measure rate of change of the input parameters and the number of inputs and outputs are not limited to small number.

The objective of this system is to analyze the DBMS, by proactively monitoring the performance indicators like buffer miss ratio, number of active processes and the table's size that are showing signs of rapid growth and initiate control measure using fuzzy control. This architecture comprises of a module that continuously tracks the system performance my noting the important performance indicators.

The fuzzified module maps these performance indicators to fuzzy variables. The fuzzy control modules uses the fuzzy rules comprising of IF THEN kind of statements on the fuzzy input variables to decide on the fuzzy output variables. The fuzzy output variables in this case would be the tuning parameters of the DBMS. Having obtained the extent of tuning required in fuzzy terms, a de fuzzyfier module generated the crisp output parameters that are eventually used by the tuner module to fine tune the DBMS.

The input parameters namely Buffer Hit-Ratio (BHR), library-cache hit ratio (LCHR), cache-hit percentage and number of Active Users (N) are gathered from the DBMS and are used as inputs to the fuzzy control system. In the proposed system, the SGA parameters are tuned. The tuner may tune one parameter at a time or may alter several of them simultaneously as warranted by the dynamic conditions of the system.

Presented database architecture

The concept of fuzzy logic is not a control methodology, but it is a way of processing data by allowing partial set membership rather than crisp set membership of non-membership. Fuzzy logic is a problem solving control system methodology that lends itself to implementation in the system ranging from simple, small embedded micro controllers to large, networks, multi-channel PC or workstation based data acquisitions and control systems. It can be implemented in hardware, software or a combination of both.

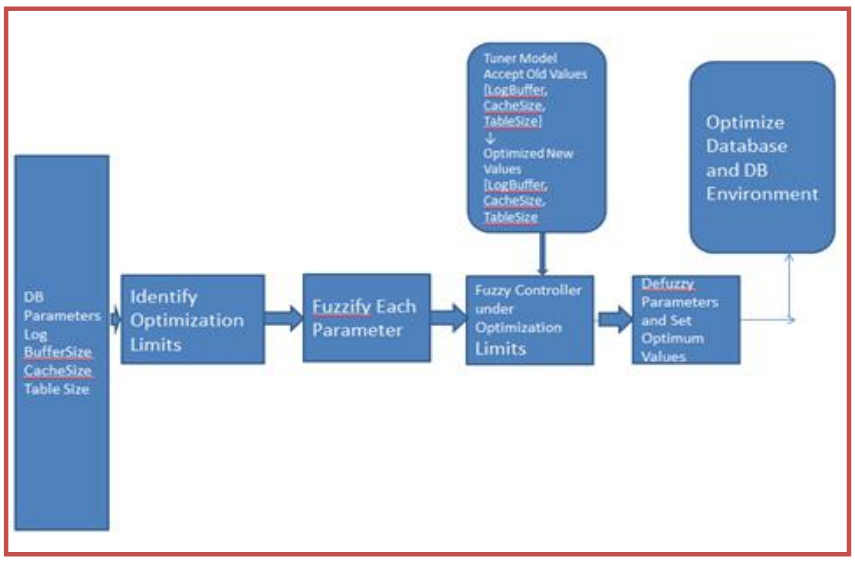

Fig.1: Proposed Architecture for Fuzzification

Here in this architecture first the value of buffer size , cache size , table size in terms of rows is given to tuner .First of all values of these input variables or parameters is identified . Here these are passed to normalization formula (b-value)/b-a where these parameters are fuzzyfied and resulted set is fuzzy set in terms of low buffer hit ratio , medium buffer hit ratio ,high buffer hit ratio, very high buffer hit ratio, similarly low cache hit ratio, medium cache hit ratio, high cache hit ratio, very high cache hit ratio, similarly with other parameters like log buffer and size of tables. After this these set of values are fed to fuzzy controller under limits where different fuzzy control rules are applied.

Fuzzy Control Rules

Some fuzzy set and operators together make fuzzy logic hence IF-THEN statements are used to make control statements example if *x* is *A* then *y* is B .where A and B are Linguistic terms. Same rule is applied in controller to these resulted fuzzy set and Linguistic terms defines extent of tuning.

Some of the fuzzy rules for the presented database architecture are:

Rule 1. IF buffer hit ratio is high and number of user load is low THEN set buffer cache size to low.
Rule 2. IF buffer hit ratio is moderate AND number of user load is high THEN set buffer cache to high.
Rule 3. IF the value of library_cache_hit_ratio is small, THEN increase the value of library_cache_size.
Rule 4. IF the value of redo_buffer_allocation_retries is approaching 1, THEN increase value of log_buffer_space.
Rule 5. IF cache_hit_percentage is lesser, THEN set sort_area_size to high

Result Screen Shots

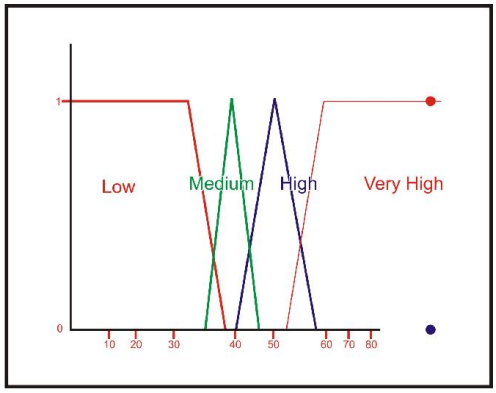

Fig. 2.1: Buffer Hit Ratio Fuzzy Analysis

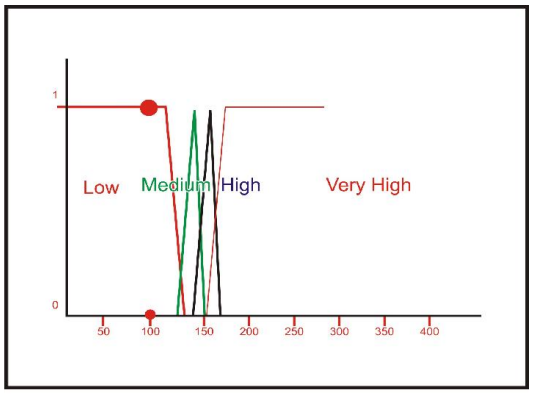

Fig. 2.2: Cache Hit Percentage Fuzzy Analysis

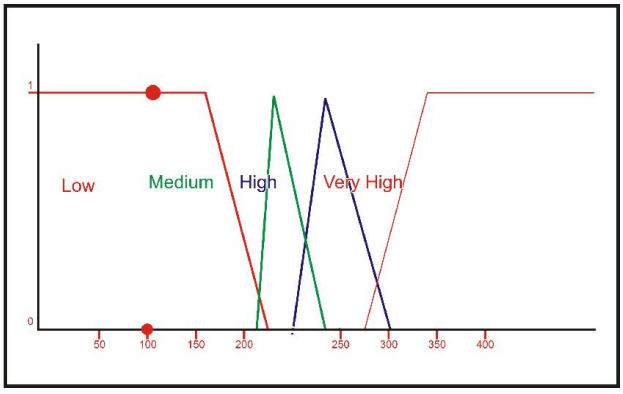

Fig. 2.3: Size of Table Fuzzy Analysis

Conclusion

In this present work an optimized approach is defined to optimize the SQL query by performing the setting of optimization parameters. The parameters include the buffer size, expected hit ratio etc. To perform this optimization and effective fuzzy query based model is presented. The obtained results show the effective generation of system model along with effective SQL optimization. In this present work, the optimization of oracle environment is performed for Oracle database under optimization parameters. The optimization is here performed using fuzzy approach. The work can be improved in future under following parameters. In this work, fuzzy approach is used to optimize the different parameters for SQL queries to less response time in Oracle environment. In future work can be done for other databases. The parameters considered for SQL optimization can also be improved.

References

Agawam. S and et.al. "Automated Selection of Materialized Views and Indexes", Conference Proceedings, VLDB, 2007.
Cheng Tu. Yi, Gang Ding, "Control-Based Tuning
Choudhuri. Surjit, Narasayya. Vivek, "Self tuning database systems: A Decade progress", Microsoft Research. 2007.
ICONS '07. Second International Conference, April 2007
Koopman. Philip, "Elements of the Self-Healing System Problem Space", IEEE Data Engineering Bulletin. 2004.
Krumpass. Stephan, Scholz. Martina. Andreas, "Quality of Serve Enabled Management of Database Workloads, Bulletin of IEEE Computer Society on Data Engineering, 2008. International Conference, April 2007.ISSN (Online): 2231 –5268 Performance enhancements using Materialized Query Table for LUW Systems", 2007.
Liu. Peng, "Design and Implementation of Self-healing Database system", IEEE Conference, 2005.
Rodd.S. F., Kulkarni.P. Uma Kant, A. R. Yard International Journal of Computer Applications 2012
Sharma Kumar Hitesh & etal. Database Systems Journal vol. III, no. 4/2012
Soror.A. Ahmed, Aboulnaga. Ashraf, Salem.Kenneth, "Database Virtualization : A new Frontier for Database Tuning and Physical Design", IEEE 2007, pp 388-394.
Verma Ankit IJCSMS International Journal of Computer Science & Management Studies, Vol. 11, Issue 01, May 2011
Wiese.David; Gennadi .Rabinovitch, "Knowledge Management in Autonomic Database Performance Tuning", 20-25 April 2009.