International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 1 Issue 10, December- 2012

Architecture Of Database Tuning

Ms. Rushita Patel
Lecturer, Computer Application,
Anand commerce college, Anand

Mr. Alpesh Rabari
Lecturer, Computer Application,
Anand commerce college, Anand

Abstract

Every day there is a huge amount of growth in business data. it growth from kilo byte, mega byte, giga byte, tera byte, peta byte, and so far .unless & until business is running the increasing rate of data will be growing. So to manage those huge amount of data is the most critical part of information system. Because of this issue, database tuning is necessary. Tuning a 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. In order to reach a trade-off between the desire to maintain efficient databases and coherent databases, we propose a framework in which a database should be able to self-tuning its logical database schema with respect to SQL workloads and the data themselves. We discuss the main points of this framework, its feasibility and its relationships with some data mining problems. It is critical that database systems be easy to manage, predictable in their performance characteristics, and ultimately self-tuning.

1. Introduction

We begin with a database and collect, for use as a workload input, a sequence of queries that were executed during normal usage of the database. We empirically validate as facts the heuristics that Database Administrators (DBAs) currently use as in doing this task manually: for tables that have a high ratio of update, delete, and insert to retrieval queries one should horizontally partition, but for a small ratio one should fully replicate a table. Such rules of thumb are reasonable, however we want to parameterize some common guidelines that DBAs can use. As we have seen, hardware costs fall rapidly while human costs remain relatively static. This leads to a condition there the human costs of manual tuning activities outpaces the costs of faster hardware (see figure 1). Most large databases are managed by DBAs who are responsible for the good performance of the database.

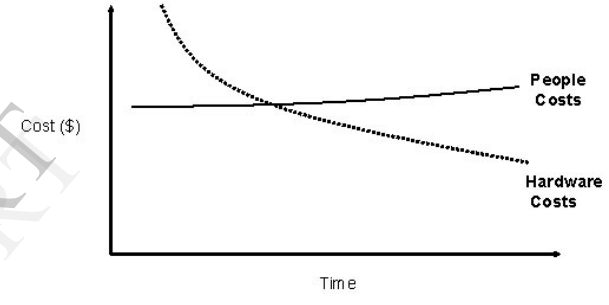


Fig.1. H/W cost vs. Human Cost

Today's Database Management Systems (DBMS) require Database Administrators (DBA) to tune more and more parameters for an optimal use of their databases. Due to the difficulty of such a task and since a large number of companies cannot justify a full-time DBA presence, simplifying administration of DBMS is becoming a new challenge for the database community: The idea is to have databases adjusting themselves to the characteristics of their applications [6]. Database technology has an extremely successful track record as a backbone of information technology (IT) throughout the last three decades. High-level declarative query languages like SQL and atomic transactions are key assets in the cost-effective development and maintenance of information systems.

2. Crisis Indicators

Roughly speaking, a physical database schema without data redundancy can be either an

”efficient” database at the cost of having a lot of null values on join attributes or a ”null free database” at the cost of performing more joins to perform equivalent SQL queries. database administrators (DBA) have to maintain on a daily basis ”efficient databases”, i.e. databases on which main SQL queries have to perform efficiently for end-users, while keeping ”coherent” databases, i.e. databases without update problems. Moreover, the huge number of null values occurring in practice may incur a significant overhead on his daily work, either to optimize the memory layout or to maintain (or design new) SQL queries. Motivated not only by the difficulty of tuning but also from the need to reduce the total cost of ownership in their products, several commercial DBMS vendors offer automated physical design tools with several features. administration and tuning staff dominate the cost of ownership for a database system [4]. Database systems offer more and more features, leading to extremely broad and thus complex interfaces. Quite often novel features are more a marketing issue rather than a real application need or technological advance; for example, a database system vendor may decide to support a fancy type of join or spatial index in the next product release because the major competitors have already announced this feature. As a result, database systems become overloaded with functionality, increasing the complexity of maintaining the system’s code base as well as installing and managing the system. The irony of this trend lies in the fact that each individual customer (e.g., a small enterprise) only makes use of a tiny fraction of the system’s features and many high-end features are hardly ever exercised at all. Identification of the root cause of a performance problem is not easy [3, 2, 5, 7]. It is not uncommon for DBAs to spend large amounts of time and resources fixing performance symptoms, only to find that this had marginal effect on system performance. Often a symptom is treated mainly because the DBA has seen that symptom before and knows how to treat it. Lack of a holistic view of the database leads to incorrect diagnosis and misdirected tuning efforts resulting in overconfigured systems increasing the total cost of ownership [8, 9, 10].

3. Performance Tuning

Modern database systems have complicated interactions between their sub-components and have the ability to work with a variety of applications. This results in a very large list of potential performance issues such automatic diagnosis solutions could identify. Also, as new database technologies and applications are invented and older ones are obsolete, it is pivotal that automatic diagnostic and tuning solutions can easily be adapted to accommodate such changes.

The objectives of an automatic performance diagnosis and tuning solution can be summarized as follows:

• Should posses a holistic view of the database and
understand the interactions between various database components.

• Should be capable of distinguishing symptoms from the actual root-cause of performance bottlenecks.

•Should provide mechanisms to diagnose performance issues on their first occurrence.

•Should easily keep up with changing technologies.

with some degree of decreasing performance. A system's ability to accept higher load is called scalability, and modifying a system to handle a higher load is synonymous to performance tuning.

Systematic tuning follows these steps:

Assess the problem and establish numeric values that categorize acceptable behaviour. Measure the performance of the system before modification.

Identify the part of the system that is critical for improving the performance. This is called the bottleneck. Modify that part of the system to remove the bottleneck. A performance problem may be identified by slow or unresponsive systems. This usually occurs because high system loading, causing some part of the system to reach a limit in its ability to respond. This limit within the system is referred to as a bottleneck. A handful of techniques are used to improve performance.

Tuning operating system performance on computers running different operating systems.

One should do performance tuning for the following reasons [11]:

The speed of computing might be wasting valuable users' time (users waiting for response); Enable your system to keep-up with the speed business is conducted; and Optimize hardware usage to save money (companies are spending millions on hardware).

4. RELATED WORK

Database Management Systems have a wide range of application in real life such as in any corporate house, the online systems, and e-commerce applications. Thus, to provide reliable services with quick query response times to the customers, the Database Management Systems (DBMS) must function efficiently and should have built-in support for quick system recovery time in case of partial failure or system resource bottlenecks. The performance of these systems is greatly influenced by several factors which include database size which keeps on growing with its usage over a period of time, increased user base, sudden increase in the user processes, improperly or un-tuned DBMS. All these factors degrade the system response time and hence a system is required that would anticipate the performance Degradation by carefully monitoring the System performance indicators and would auto tune the system. In other words, a self-tuned database system is desired for the optimization of query response time.

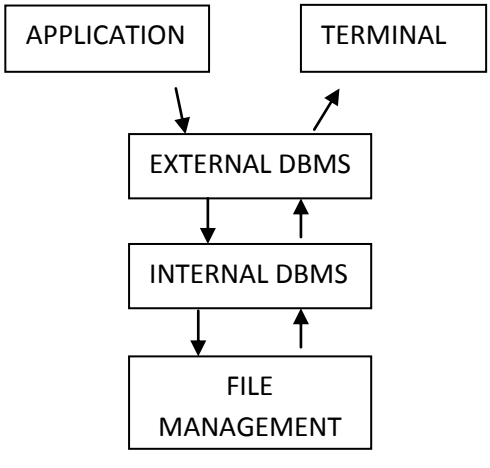


Fig.1: Database Management System

Many commercial applications require huge databases for the purpose of data aggregation as well as data distribution. As there are limited storage resources available performance enhancing features such as selection of indexes, materialized views, horizontal and vertical partitions is a challenging optimization problem. Thus, the task of data aggregation as well as data distribution can be made somewhat automatic so that it may incorporate dynamic changes effectively.

5. Manual System Architecture

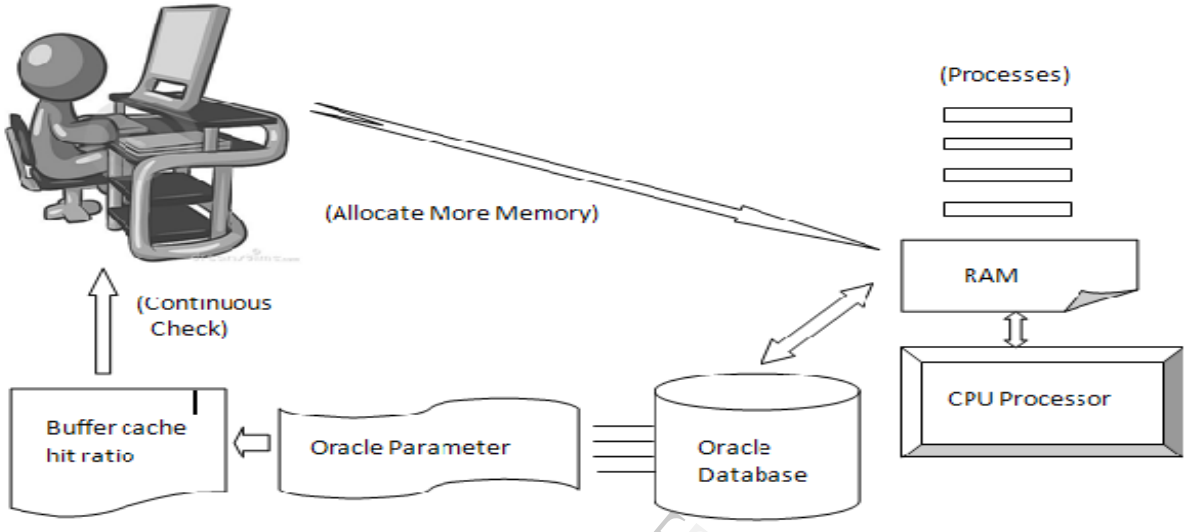
Database Administrator is responsible for enhancing the performance of database system. The detection of performance degradation is achieved by continuously monitoring system performance parameters. Several methods including the usage of materialized views and indexes, pruning table and column sets, usage of self healing techniques, usage of physical design tuning etc have been proposed that proactively monitor the system performance indicators, analyze the symptoms and auto tune the DBMS to deliver enhanced performance.

The performance degradation is due to increased workload on the system. This increased load has to be minimized to enhance the response rate of the system. In order to achieve this objective, either the administrator decreases some amount of load by closing some files or he may increase the RAM. The administrator has to check continuously or we can say, at regular intervals the Buffer Cache Hit (BCH) ratio. Based on this hit ratio, the database administrator determines if more amount of RAM has to be allocated. This task of load reduction by increasing RAM requires manual intervention and thus may take even years to complete.[1]

However, Oracle manages RAM memory demands according to the demands of each task by using sophisticated algorithms to improve the speed of RAM intensive tasks. Oracle DBA can dynamically de-allocate RAM memory as well as re-allocate it. But since database administrator is a normal human being, he cannot calculate the actual amount of RAM memory required by an application.

Due to this limitation of DBA, the allocation of RAM manually for optimizing Sometimes, more amount of RAM is allocated than needed which wastes the extra portion of RAM. [12] Thus, there

is a great need of dynamic memory allocation features to create a self tuning database.

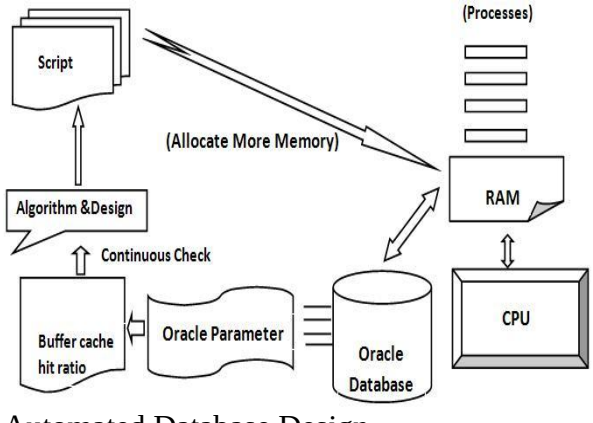


Manual database Design

In Oracle Database 10g, a self tuning feature such as Automatic Memory Management (AMM) allows the database system to detect shortages and adjusts the main memory regions according to the changing demands on the Oracle environment. Therefore, researchers are now focusing on the development of self tuning techniques such as the COMFORT automatic tuning project [13] or the MAPE approach given by IBM [14] for a continuous adaptation.

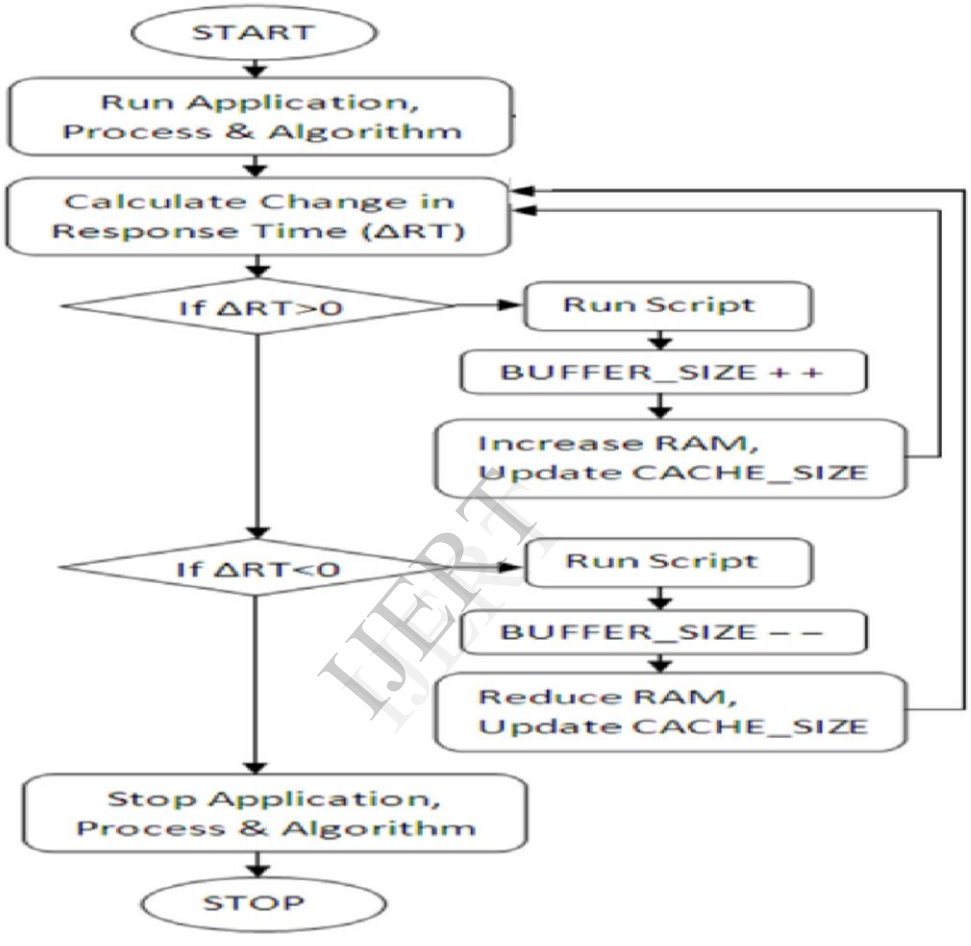
6. PROPOSED ARCHITECTURE

In this research proposal, we would provide a self tuned database system as shown in fig 4 in order to enhance system performance. Since DBA is responsible for administration and optimization of various tasks, he can either increase RAM or can decrease the amount of load on CPU for the purpose of performance optimization. But this would be time consuming technique as DBA is a normal human being who cannot perform complex calculations within seconds like a computer system. DBA may not know exactly how much RAM is to be allocated for enhancing system performance. So, we propose an approach to automate this optimization task of DBA as shown in fig i.e. the task which DBA has to do for performance enhancement would now be done by the computer system within small timelines.



Automated Database Design

Flowchart for Automated Tuning



7. Conclusion

Tuning the database can become quite complex, but Oracle9i offers the administrator an unparalleled ability to control the PGA and SGA. Until Oracle9i evolves into a completely self-tuning architecture, the DBA will be responsible for adjusting the dynamic configuration of the system RAM. Automated SGA adjustment scripts can be used to allow theDBA to grow and shrink the SGA regions. These scripts are placed in dbms_job for scheduled processing. Oracle provides enhanced views in v$process, v$pgastat to allow you to monitor the behavior of the RAM sort area The v$ views in Oracle9i also provides insights about the RAM usage for individual SQL statements within the library cache. Considerations introduced in this paper for self-tuning the logical database design are simple though very important in practice. Between the desire to maintain efficient databases for end-users and the desire to maintain coherent databases for database programmers and end-users, we have pointed out how SQL workloads and the data themselves could be used to reach a compromise among contradictory objectives. Through the notion of assistant2, one could envision the self-tuning of logical database design from both SQL workloads and the data themselves. Along the paper, we have pointed out some challenges that remain to be addressed to cope with self-tuning logical database design.

8. References

[1] I. Alagiannis, Towards Adaptive, Flexible, and Self-tuned Database Systems, (DIAS, I&C, EPFL) in EDIC-ru/05.05.2009

[2] S. Chaudhuri and G. Weikum: Rethinking Database System Architecture: Towards a Self-tuning RISC-style Database System. 26th International Conference on Very Large Data Bases, Cairo, Egypt, 2000.

[3]G. Weikum, A. Mönkeberg, C. Hasse, P. Zabback: Selftuning Database Technology and Information Services: From Wishful Thinking to Viable Engineering, 28th International Conference on Very Large Data Bases, Hong Kong, China, 2002.

[4] Rethinking Database System Architecture: Towards a Self-tuning RISC-style Database System in Cairo, Egypt, 2000

[5] J. L. Hellerstein. Automated Tuning Systems:Beyond Decision Support. In the proceedings on Computer Measurement Group, 1997

[6] P. Bernstein and al. The ASILOMAR report on database research. ACM Sigmod Record, 27(4):74–80, 1998.

[7] K. P. Brown, M. Mehta, M.J. Carey, M. Livny: Towards Automated Performance Tuning for Complex Workloads. 20th International Conference on Very Large Data Bases, Santiago, Chile, 1994.

[8]Hurwitz Group: Achieving Faster Time-to-Benefit and Reduced TCO with Oracle Certified Configurations, March 2002.

[9]S. Chaudhuri and G. Weikum: Rethinking Database System Architecture: Towards a Self-tuning RISC-style Database System. 26th International Conference on Very Large Data Bases, Cairo, Egypt, 2000.

[10] Gartner Group: Total Cost of Ownership: The Impact of System Management Tools, 1996.

[11] Frank Naudé, Oracle Monitoring and PerformanceTuning, http://www.orafaq.com/faqdbapf.htm

[12] Foundations of Automated Database Tuning, VLDB ‘06, September 12–15, 2006, Seoul, Korea.Copyright 2006 VLDB Endowment, ACM

[13]AutoAdmin:Self-Tuning Database Systems Technology, Copyright 2006 IEEE

[14]Self-Tuning Database Systems: A Decade of Progress, Copyright