

Chapter LXXX

Self-Tuning Database Management Systems

Camilo Porto Nunes

Federal University of Campina Grande, Brazil

Cláudio de Souza Baptista

Federal University of Campina Grande, Brazil

Marcus Costa Sampaio

Federal University of Campina Grande, Brazil

INTRODUCTION

Computing systems have become more complex and there is a plethora of systems in heterogeneous and autonomous platforms, from mainframes to mobile devices, which need to interoperate and lack effective management.

This complexity has demanded huge investments to enable these systems to work properly. It is necessary to invest on software acquisition and installation: management, administration, and update. These costs compound the *Total Cost of Ownership* (TCO), which tends to increase exponentially according to the software complexity.

Information Technology (IT) focuses mainly on providing information services in order to achieve simplicity, agility, large access to information, and competitiveness. **Database Management**

Systems (DBMS) are part of the IT infrastructure in large, medium, and even small enterprises.

According to LIGHTSTONE (2003), the complexity of a DBMS enhances the difficulty in administrating it, as there are too many tasks a Database Administrator (DBA) should consider. Thus, more people are needed to be trained to efficiently manage this software. As a result, the maintenance costs increase and the problem is not properly solved. Performance, time-to-market, throughput, robustness, availability, security, are some concerns that a **DBA** should consider.

Suppose that, due to an unexpected event (e.g., too many clients accessing the DBMS), the average load in a database server changes quickly and the query response times may become unacceptable. The **DBA** must, rapidly (a) detect the problem (unacceptable response time); (b) find the root cause

of the problem (e.g., too many swap operations); and (c) tune the server to maintain response time goal (e.g., to increase memory size).

These three activities—symptom detection, diagnosis, and tuning—if executed manually by a DBA, can take a long time. Worse still, the DBA could be absent by the time the symptom arises.

On the other hand, **autonomic computing** has become an important research theme, aiming to reduce the manual efforts of administration of complex systems. Regarding DBMSs, they should be capable to react automatically to negative events, under the form of adaptive continuous tuning (LIGHTSTONE, 2003). For example, when detecting a load change, with risk of degrading the global performance of the system, a DBMS could automatically adjust some query execution plans; change its memory pool configuration; or still adjust data on disk (through index creation, data clustering, table partition, and so on) in order to maintain its overall performance. DBMSs with some capabilities of adaptive continuous tuning are called **self-tuning** ones.

This chapter addresses the issue of self-tuning DBMS. In the remainder of the chapter we present a background on this topic, followed by a discussion focusing on performance, indexing, and memory issues. Then, we highlight future trends and conclude the chapter.

BACKGROUND

Manual **DBMS tuning** requires a collection and analysis of some performance metrics by the DBA. Examples of such metrics include hit ratio in data cache, number of I/O executed in a given table, etc. After collection and analysis of these metrics, the **DBA** may decide whether it is necessary to adjust the DBMS. If an intervention is due, then the **DBA** needs to know which components are likely to be adjusted and what are the impacts of such adjustments.

In order to adjust the DBMS to improve performance, the **DBA** may act in several components such as memory buffers, index creation, SQL tuning, etc. In this chapter we focus on index and memory tuning, due to their importance in the overall performance of the DBMS.

Memory buffers consist of memory areas which are used for a given purpose. DBMS uses them, for example, to store the query execution plans, for both data and SQL statement caching, as an area for data sorting, and for storing session data. The tuning of these buffers may improve the performance of the submitted queries. For example, by increasing the memory buffer size, the number of physical I/O operations may be reduced, which results in performance gains.

Index maintenance is one of the most important tasks in DBMS administration. Hence, it deserves special attention due to its importance in the whole tuning process. An **index** may help the execution of a query submitted to the DBMS by reducing the number of accesses to disk. Nonetheless, during data update, insertion and deletion, an index may decrease the DBMS performance, as index reorganization may be necessary in such operations.

Once DBMS tuning involves many parameters, components and metrics, it is very difficult for humans to efficiently complete this task. Therefore, there has been a great effort from both academia and industry in providing automatic solutions for database administration.

Automatic Diagnosis of Performance Problems

The main motivation to automate the task of diagnosing performance problems concerns the complexity and importance for the subsequent decision making to solve the problem. The diagnosis, when executed manually, usually depends on DBA's experience, because of the great number of metrics involved. Then it is difficult and ex-

7 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/self-tuning-database-management-systems/20761

Related Content

A Link-Based Ranking Algorithm for Semantic Web Resources: A Class-Oriented Approach Independent of Link Direction

Hyunjung Park, Sangkyu Rho and Jinsoo Park (2011). *Journal of Database Management* (pp. 1-25). www.irma-international.org/article/link-based-ranking-algorithm-semantic/49721

An Efficient Data Structure for Organizing Multidimensional Data

Guang-Ho Cha and Chin-Wan Chung (1997). *Journal of Database Management* (pp. 3-15). www.irma-international.org/article/efficient-data-structure-organizing-multidimensional/51184

A Quantitative Function for Estimating the Comparative Values of Software Test Cases

Yao Shi, Mark L. Gillenson and Xihui Zhang (2022). *Journal of Database Management* (pp. 1-33). www.irma-international.org/article/a-quantitative-function-for-estimating-the-comparative-values-of-software-test-cases/299559

Semantic Heterogeneity in Multidatabase Systems: A Review and a Proposed Meta-Data Structure

Te-Wei Wang and Kenneth E. Murphy (2004). *Journal of Database Management* (pp. 71-87). www.irma-international.org/article/semantic-heterogeneity-multidatabase-systems/3321

The Algos Center: Information Systems in a Small Non-Profit Organization

Susan J. Chinn, Charlotte A. Pryor and John J. Voyer (2006). *Cases on Database Technologies and Applications* (pp. 46-63). www.irma-international.org/chapter/algos-center-information-systems-small/6204