

## Chapter 8.10

# Control-Based Database Tuning Under Dynamic Workloads

**Yi-Cheng Tu**

*University of South Florida, USA*

**Gang Ding**

*Olympus Communication Technology of America, Inc., USA*

### INTRODUCTION

Database administration (tuning) is the process of adjusting database configurations in order to accomplish desirable performance goals. This job is performed by human operators called database administrators (DBAs) who are generally well-paid, and are becoming more and more expensive with the increasing complexity and scale of modern databases. There has been considerable effort dedicated to reducing such cost (which often dominates the total ownership cost of mission-critical databases) by making database tuning more automated and transparent to users (Chaudhuri *et al.*, 2004; Chaudhuri and Weikum, 2006). Research in this area seeks ways to automate the hardware deployment, physical database design, parameter configuration, and resource management in such systems. The goal is to achieve acceptable performance on the whole

system level without (or with limited) human intervention.

According to Weikum *et al.* (2002), problems in this category can be stated as:

**workload  $\times$  configuration (?)  $\rightarrow$  performance**

which means that, given the features of the incoming workload to the database, we are to find the right settings for all system knobs such that the performance goals are satisfied. The following two are representatives of a series of such tuning problems in different databases:

- **Problem 1: Maintenance of multi-class service-level agreements (SLA) in relational databases.** Database service providers usually offer various levels of performance guarantees to requests from different groups of customers. Fulfillment

of such guarantees (SLAs) is accomplished by allocating different amounts of system resources to different queries. For example, query response time is negatively related to the amount of memory buffer assigned to that query. We need to dynamically allocate memory to individual queries such that the absolute or relative response times of queries from different users are satisfied.

- **Problem 2: Load shedding in stream databases.** Stream databases are used for processing data generated continuously from sources such as a sensor network. In streaming databases, data processing delay, i.e., the time consumed to process a data point, is the most critical performance metric (Tatbul *et al.*, 2003). The ability to remain within a desired level of delay is significantly hampered under situations of overloading (caused by bursty data arrivals and time-varying unit data processing cost). When overloaded, some data is discarded (i.e., load shedding) in order to keep pace with the incoming load. The system needs to continuously adjust the amount of data to be discarded such that 1) delay is maintained under a desirable level; 2) data is not discarded unnecessarily.

Such problems can hardly be solved by using rules of thumbs and simply throwing in more hardware. In the following section, we shall also see that the traditional approach of treating tuning problems as static optimization problems does not work well for dynamic workloads such as those with OLAP queries. In this chapter, we introduce an emerging new approach to attack self-tuning database problems that is based on well-established results in feedback control theory. Specifically, we address the core issues of the approach and identify critical challenges of applying control theory in the area of self-tuning databases.

## BACKGROUND

Current research in automatic tuning (or self-tuning) of databases tend to treat the problem as an optimization problem with the performance metrics and workload characteristics as inputs. The main drawback for this strategy is: real-world workloads, especially OLAP workloads, are highly unpredictable in that their parameters and behaviors can change very frequently (Tu *et al.*, 2005). Such uncertainties in workloads can bring dramatic variations to system performance and cause the database to run in suboptimal status. In order to maintain consistently good performance, we need to develop means for the database to quickly adapt to the changes in workload.

One way to address the above challenge is to treat the problem as an online optimization (Chaudhuri and Weikum, 2006) and solve it by incremental algorithms. However, there is generally no guarantee on the accuracy and convergence of such algorithms, and some problems have no incremental solutions. Another important question, which is either ignored or answered empirically in current studies, is *how often do we need to rerun the optimization?* Our observation is that people tend to follow *ad hoc* strategies for individual problems in this field. It would be desirable to have a common theoretical framework under which a series of problems can be approached.

In this chapter, we argue that control theory provides such a foundation to approach the aforementioned problems in self-tuning databases. The reason for this is: designing systems with resistance to internal/external uncertainties is one of the main goals of control theory (Hellerstein *et al.*, 2004). Note that control theory is not a single technique. Instead, it is the collection of a rich set of mathematical tools for analyzing system dynamics and designing mechanisms with guaranteed performance. We discuss some of the core issues of using control techniques in self-tuning databases. Currently, we have seen

6 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/control-based-database-tuning-under/8051](http://www.igi-global.com/chapter/control-based-database-tuning-under/8051)

## Related Content

---

### An Evaluation Framework for Component-Based and Service-Oriented System Development Methodologies

Zoran Stojanovic, Ajantha Dahanayake and Henk Sol (2004). *Advanced Topics in Database Research, Volume 3* (pp. 45-69).

[www.irma-international.org/chapter/evaluation-framework-component-based-service/4353](http://www.irma-international.org/chapter/evaluation-framework-component-based-service/4353)

### Privacy in Multidimensional Databases

Francesco M. Malvestuto and Marina Moscarini (2003). *Multidimensional Databases: Problems and Solutions* (pp. 310-360).

[www.irma-international.org/chapter/privacy-multidimensional-databases/26973](http://www.irma-international.org/chapter/privacy-multidimensional-databases/26973)

### Adaptation of an Agile Information System Development Method

Mehmet N. Aydin, Frank Harmsen, Jos van Hillegersberg and Robert A. Stegwee (2007). *Research Issues in Systems Analysis and Design, Databases and Software Development* (pp. 54-88).

[www.irma-international.org/chapter/adaptation-agile-information-system-development/28433](http://www.irma-international.org/chapter/adaptation-agile-information-system-development/28433)

### Using Ontology and Rule-Based Reasoning for Conceptual Data Models Synonyms Detection: A Case Study

Ljubica Kazi and Zoltan Kazi (2019). *Journal of Database Management* (pp. 1-21).

[www.irma-international.org/article/using-ontology-and-rule-based-reasoning-for-conceptual-data-models-synonyms-detection/230292](http://www.irma-international.org/article/using-ontology-and-rule-based-reasoning-for-conceptual-data-models-synonyms-detection/230292)

### Detecting Users' Dietary Preferences and Their Evolutions via Chinese Social Media

Qingqing Zhou and Chengzhi Zhang (2018). *Journal of Database Management* (pp. 89-110).

[www.irma-international.org/article/detecting-users-dietary-preferences-and-their-evolutions-via-chinese-social-media/218928](http://www.irma-international.org/article/detecting-users-dietary-preferences-and-their-evolutions-via-chinese-social-media/218928)