Available online at www.sciencedirect.com
SCIENCE DIRECT®
Data & Knowledge Engineering 57 (2006) 148–178
DATA & KNOWLEDGE ENGINEERING
www.elsevier.com/locate/datak

Utilization of execution histories in scheduling real-time database transactions

Erdogan Dogdu

Department of Computer Engineering, TOBB Economics and Technology University, Ankara, Turkey

Received 5 January 2005; received in revised form 5 January 2005; accepted 27 April 2005
Available online 31 May 2005

Abstract

Real-time database systems support data processing needs of real-time systems where transactions have time constraints. Here we consider repetitively executed transactions, and assume that execution histories are logged. A well-known priority assignment technique called earliest-deadline-first is biased towards short transactions in which short transactions have better chances of completing their executions within their deadlines. We introduce the notion of "fair scheduling" in which the goal is to have "similar" completion ratios for all transaction classes (short to long in sizes). We propose priority assignment techniques that overcome the biased scheduling and show that they work via extensive simulation experiments.

Keywords: Real-time databases; Transaction processing; Priority assignment techniques; Execution histories

1. Introduction

Real-time databases are databases with real-time properties. Real-time databases are first conceptualized to serve the data processing needs of real-time systems. Real-time systems are those used in digital control, manufacturing, telecommunications, signal processing, command and control systems [1–4]. In simple term, the distinguishing difference of these systems is that the

E-mail address: edogdu@etu.edu.tr
URL: http://www.etu.edu.tr/~edogdu

0169-023X/$ - see front matter
doi:10.1016/j.datak.2005.04.008

tasks in these systems have to be completed on a timely-basis [5]. Since processes and transactions in real-time systems are time constrained, databases that serve these systems also need to consider the same time constraints. Typical database management systems are only concerned with the logical correctness of data; processing times of queries, execution times of transactions are not strictly controlled in traditional database systems as it is required in real-time systems. Therefore, the research in real-time database systems is initiated to address the time dimension of the data in real-time systems and the timely processing of tasks in database systems. Previous work in real-time database systems typically deals with the problem of maximizing the system throughput via priority-based scheduling methods, and/or admission and load control techniques. System throughput can, for example, be the number of database transactions that are completed within their given time constraints usually in the form of time deadlines.

In real-time systems some tasks are executed repeatedly. Unlike a computer system, which is a general-purpose computing machine, a real-time system solves a specific problem. This could be, for example, controlling the breaking system automatically in an automobile when the car is running; or for example, collection of weather data from scattered nodes of a wireless sensor network. Therefore, most tasks in real-time systems are routine and executed repeatedly, but not necessarily periodically. Therefore, there is a need to address the issue of repeated execution of tasks, or as called in database terminology ''transactions'', in real-time database systems.

In this paper, we consider the processing of transactions that are executed repeatedly in real-time database systems. Transactions in database systems do not have predictable execution times like the tasks in real-time systems. Data in database systems are accessed by many transactions concurrently. To preserve database consistency and the correctness of transactions, database systems employ strict concurrency control techniques while processing transactions, such as the two-phase locking method. These concurrency control techniques put transactions into waiting mode during conflicting access to the data and this in the end makes the transaction execution prediction an impossible job. Therefore, estimation of transaction execution time is not a feasible approach in predictive scheduling of transactions in real-time database systems. Instead, we assume the use of ''transaction execution histories'' of repeated transactions towards enhancing the systems throughput.

Previous work in real-time database systems research focused mainly on priority assignment techniques for optimized scheduling transactions. Most performance studies use earliest-deadline-first (EDF) policy for priority assignment [6]. EDF assigns a higher priority value to a transaction with an earlier deadline so that transactions with earlier deadlines have the advantage of accessing data first against transactions with later deadlines that are concurrently accessing the same data. The logic behind is that transactions with later deadlines have more time to process their work within their time deadlines, and by assigning higher priorities to transactions with earlier deadlines, overall system throughput (i.e., the number of transactions that meet their deadlines, or successful transactions) can be increased. Although EDF has been shown to improve the average success ratio of the system (fraction of transactions completing successfully within their deadline), it discriminates against longer transactions under overload conditions [7–9].

We propose priority assignment techniques that consider the biased nature of EDF, and attempt to eliminate the discriminatory behavior by adjusting the priorities using transaction execution history information. Two basic pieces of data that are kept in transaction execution histories are the transaction completion ratio and the transaction completion time. The completion ratio of a