I J C T A, 9(9), 2016, pp. 3995-4001

Predictive and Reactive Approach for Differentiated Real-time Data Services

A.S. Shakthi Priya[1] **and R. Karthikeyan**[2]

ABSTRACT

Generally handling real-time data services is a challenging process in applications like traffic monitoring, stock trading and target tracking. Therefore these real-time service requests which depends on the service classes and data access needs for differentiated real-time data services are handled by deadline assignment when a system is busy and supports specific target delay. DDR, the ratio of deadline to the actual data service delay is done by feedback control mechanism under dynamic workload. Here VM's allocation for all purpose is being used for all purpose as the in-memory systems tends to be sensitive to other sources of overhead which is not considered in a traditional I/O bounded system. Thereby problems such as fault-tolerance, consistency, overload and congestion are been treated by a different perspective of virtual machine allocation during overflow in queuing and assignment.

Keywords: Real Time Data Base System, Differential Real-Time Data Service, Deadline Assignment, Virtual Machine.

1. INTRODUCTION

In real-time applications, like stock trading, target tracking, traffic monitoring, etc., which are high data-intensive, using updated temporal data which represent the current real world stats it is difficult to for a real-time database to handle a data service request within its allocated deadline. A no. of different service classes e.g. gold, silver, bronze classes which offer different delay and costs are desirable by different subscribers for an RTDB to be allowed.

Deadlines are been determined by the application in which data service requests like agile manufacturing and target tracking are done in certain RTDB applications. Users would tend to assign short deadlines to their own request for data service if they are allowed to in an open system like traffic monitoring and online trade. this would result in poor service due to that system would be severely overloaded. Relative work on assignment of dead-line to real-time service request and also to support such data service request is done.

Every request is been handled and addressed based on the estimated time taken for a transaction or query in real-time in terms of no. of data accesses required for a particular transaction. Therefore depending on the user's service classes to which they belong and the estimated transaction size a differentiated approach is taken foe the dead-line assignment as e.g. 3:2:1 for Bronze : Silver : Gold as their relative delay ratio. Until and unless a overload condition occur, scheduling for all the data service request is done by EDF (earliest Deadline First) algorithm is assigned for all deadlines. When similar no. of data access service has been approached by both a high class user and low class client as on an average favorable services are being provided for the latter in terms of relative delay depending on their service classes.

However, with dynamic change occurrence in real world status the process of a real-time transaction to occur in a differentiated manner and on time is a challenge in application like stock market, etc. In such

[1] Master Degree, Department of Computer Science and Engineering Veltech Multitech College of Engineering, Chennai, India, *Email: bubly_blu28@yahoo.com*

[2] Head of Department, Assistant Professor, Department of Computer Science and Engineering VeltechMultitech College of Engineering, Chennai, India, *Email: rkarthikeyan@veltechmultitech.org*

dynamic environment, effective theoretical techniques support the performance of the system as desirable to overcome the challenges. Therefore such control over the performance of the system are done by the designing of an RTDB for the implementation of a feed-back control mechanism. Hence the design for a dynamic RTDB is done by the relation between DDR (Delay to Deadline Ratio) which is given as the ratio of actual data service delays to the deadline and the backlog of a database.

In this paper for the feedback control mechanism, database backlog and DDR is being handled by a different approach of virtual machine allocation in order to enhance the performance matrices for the RTDB application. The DDR is based on the database backlog which increases with increase in backlog and its inverse. Whereas the backlog is directly proportional to the database load as its representation. Depending on how fine grained the data service requests are handled close to its deadline is given by DDR. The desired DDR for the RTDB is designed by a feedback control system. The Target DDR in order to avoid overload occurrence is done by introducing a admission control mechanism at the feedback control loop for every data service request by the user.

2. PROBLEM DEFINITION AND OBJECTIVE

Contrast to the deadline assignment many other existing methods for real-time system design based on feedback control mechanism do not support RTDB, which miss a deadline ratio control. These miss ratio are given for overload conditions and underutilized scenarios as 1 and 0 respectively, which tends to asymmetric and instability for feedback control problems. These problems are been approached by deadline assignment depending on data access needs for each and every real-time data service request and also by the feedback control mechanism in order to support the specific target delay to DDR. Unlike any other real-time database system an extended open source database is implemented. This approach shows that the experimental results support the requirements by service differentiation and DDR bound.

3. RELATED WORKS

3.1. EDF Scheduling For Temporal Consistency Maintenance

In real-time system every object state depends on the passage of time for validity which depends on the temporal validity interval. Depending on the real-time system track its environment correctly the temporal consistency of data in maintained. Hence by EDF algorithm, problem is handled by assigning period and deadline, providing solution which results process utilization found in linear time than that of a traditional technique. This enhances sensor transaction for arbitrary deadline which may not be optional for problem with increase in size.

3.2. Adaptive Scheduling Of Web Transaction

User satisfaction is based on successful transmission in dynamic web database system which are highly interactive that supports web transactions and data queries generating web-pages as workflow. Before every deadline relevant data update is done for each instance of transaction periodically for sensor transaction with soft deadliness. The performance and successful transaction of the system are evaluated by minimized weight tardiness and minimized deviation of deadline i.e. tardiness using frame work.

3.3. Admission Control Mechanism In Cloud For Continuous Quires

Continuous queries in data stream monitoring and management services computing systems like Amazon, Google, IBM, etc, a setting for profile auction based mechanism is done with admission control. This auction of admission controls is used in deciding the charge and permission for a query depending on how much a user can afford to pay for that particular request to be processed.

3.4. Feedback of RTDB System Performance

In a time fashion system, it is a challenging process for using fresh data in transaction process due to variation of dynamic resource contention, which acts as a result of database workload leading to accessibility for system resource in further data freshness and transaction timeliness. Thereby management of ready queue size and feedback control model, indicating amount of backlog which supports the data service delay. For concurrent request from systems of thousand clients, experiment in the tested are done for performance evaluation supporting data freshness in time compare baseline with desired response time.

3.5. Estimation of Accurate Latency

Over long running high-event-rate, in a distributed even processing system a Directed Acyclic Graph (DAG) for one or more nodes of operator called query is done. Goodness for a given input i.e. the cost estimator for every node or assignment of operator and graph is processed. Many problem such as distributed operator placement, provisioning, plane, section, optimization, admission control, etc, is done for the cost estimation evaluation. A new cost estimation which provably is a maximum system latency called Mace (maximum cumulative excess) is done.

4. AN OVERVIEW OF THE SYSTEM

The new approaches enhanced in the system are assignment of deadline for request depending on service classes and data access needs along with DDR when the system is busy and in under dynamic work load. This approach of traditional I/O bounded with the implementation of same source VM's being used, the in-memory system do not tend to be more sensitive to other sources. With the challenge in handling fault tolerance, overload, consistency and congestion in existing environment, is being improvised by virtual machine allocation which overcomes the condition of deadline and congestion under a different perspective during queuing and assignment. Thereby the dynamic workload pressure is being supported by timely data service for bursty self-similar workload by a configurable and intelligible synthetic generator (BURSE) in a computing environment for predictive and reactive approach along with superposition.

5. SYSTEM ARCHITECTURE

The system architecture for a RTDB with virtual machine allocation at every element of system is been designed. Fig. 1. This system is scheduled with DDR at presence of dynamic work load under feedback control. The BURSE technique at the self-similar workload which supports bursty conditions by the predictive

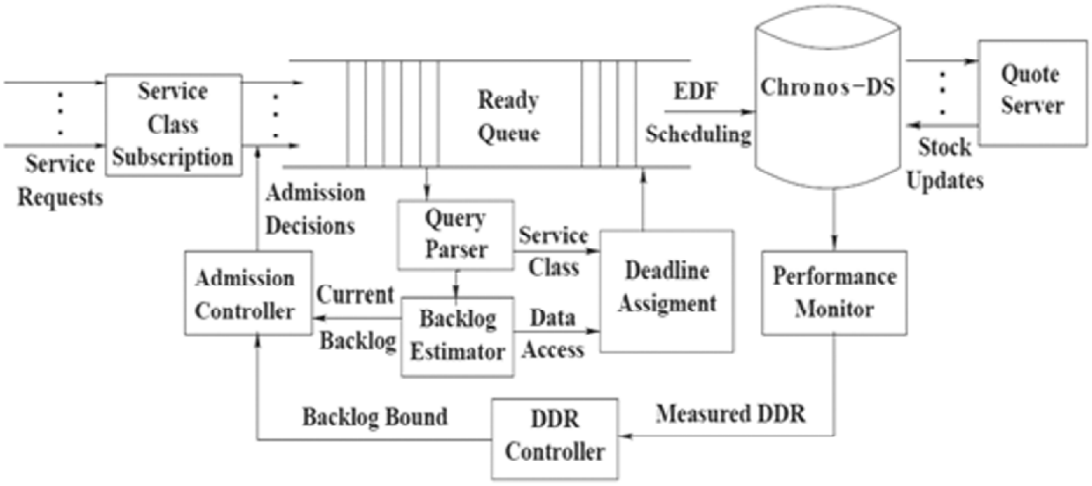

Figure 1: System Architecture

and reactive approach, the system at every individual process with VM performs leading to uninterrupted service for fresh-data streamline. By the predictive fashion along with the reactive control scheme the service delay is handled with a comparatively timely throughput.

Enhancement of timeliness in data services is approached in terms of memory consumption and lightweight CPU cycles than those of the other real-time data management systems.

5.1. Data Flow

The levels at which flow of data is analysed into three stages given by first at the entry level for access request from user end to that of verification of table availability by its tuples, variety, and update which varies with respect to the user level of priority and access control. Followed by which performance control action by feed-back mechanism due to service delay is handled for significant workload at operating range.comparatively timely throughput.

The free table verification and assignment for the tuples by the data ware-housing platform for all second level data flow is approached by job portioning which is the major process of the system for its next operation/execution. Admission control at the feedback control for every user and data access operation which is analysed depending on user request and data availability which is obtained by prioritization timely response of data queue length.

Feedback operation takes control action in prediction of delay and performance problem where it is handled and a remedy process is encountered by the reactive approach by BURSE preventing it deviating from its operation performance range. The fresh lively data update in RTDB system on continuous database refreshment, the data request are processed based on the total amount of data by the backlog estimator in the database system by DEF for maintaining consistency in every class.

6. EXPERIMENTAL PROCESS

6.1. RTDB Prototype

A prototype for data transaction in order for size estimation is done which is based on online stock system for unlimited data intensive applications. The prototype is set neither as a standard benchmark nor even as a publicly available RTDB in which operation is performed in four type of data transaction ateach level of

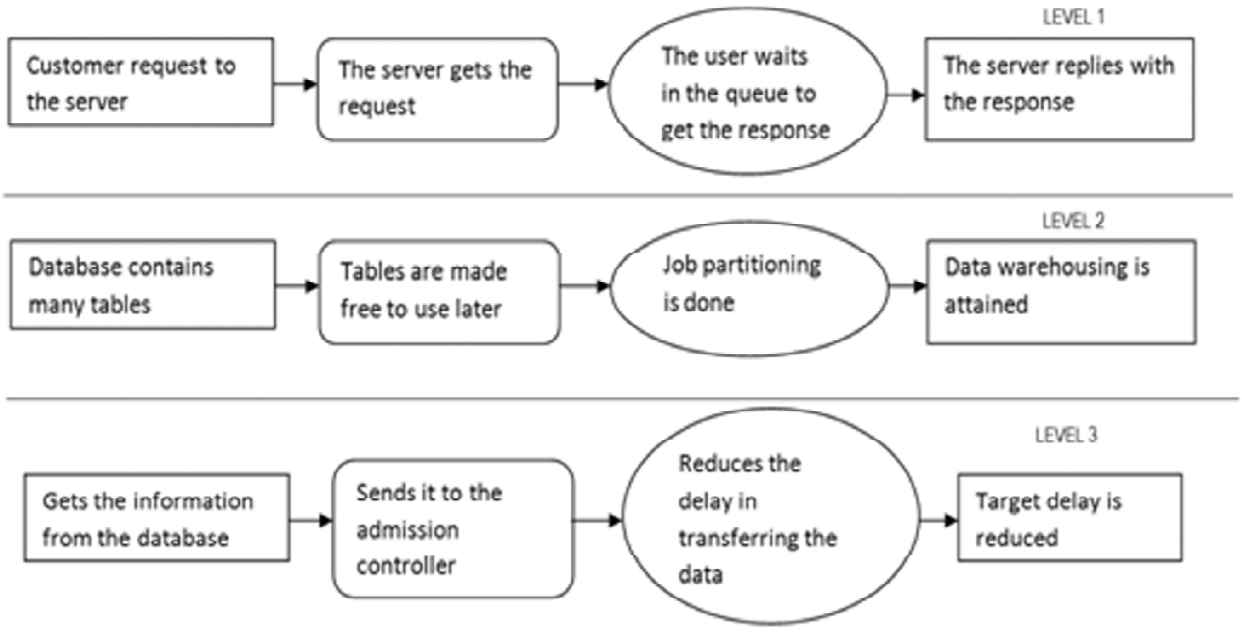

Figure 2: Data Flow

data entry to system as: viewing stock, stock portfolio, buy a stock, re-sale a stock. Like the RUBis in eBAY with online auction prototype model and in e-commerce TCP-W database system benchmark, temporal database update is to be done for every data query/service.

- *Viewing Stock:* These queries needs access to the database system for the access to tables containing STOCK and QUOTES with details of every stock metadata such as symbol, name, ID, etc, of that specific stock which is estimated based of average row size i.e. average no. of bytes in a row.
- *Stock Portfolio:* The details of the stock essential for purchasing it arethe pricing along with other information clubbed up together as its portfolio, which varies at each item in the system. The no. of stock items owned by the client in the portfolio for the client ID gives the estimated time.
- *Buy A Stock:* Queries which place order a particular stock based on the portfolio, requires the current price from the database system in order to process since the information is available t the shares portfolio, the purchase transaction along decides the estimated amount of data access.
- *Stock Sales:* This requires looking up for details of both portfolio of that particular stock along with the client details and price, followed by an intimation in order to indicate sale of the stock. The data access of a sale is the no. of stock items sold is its estimated amount of data transaction.

6.2. Average Performance Analysis

The DDR with EDF scheduling for an open RTDB differential across different service classes depending on the client and queries which leads to large DDR because the system negotiates its current status. Since gold and silver prioritized class request always pre-empts the bronze prioritized class request, the DDR control of gold and silver over bronze is given by the estimation time allocation of around 0.9.

6.3. Average Service Delay

Average delay for the system arises due to class-vice variation under heavy workload for different degree of service being encountered in the user perceived transaction. By service requests in classes of DR system, delays of open-RTDB is given three times larger per-class delay in bronze class undergo excess service delay

6.4. Transient Performance Analysis

A good transient response of open RTDB at the DDR overshoot observed at the high work load for golf and silver at the cost of bronze is seen. Quick cancellation of overshoot leads to similar transient DDR patterns, which supports its desired set-point. A general consistent manner of transient delay in RTDB for service class differentiation is seen with delay varying from time to time.

7. CONCLUSION AND FUTURE ENHANCEMENT

The challenging process of providing timely service in a differentiated manner for a process in soft real-time data intensive application is modeled by assignment of DDR at feedback control by DEF for maintaining consistency in every class of prioritization. The in-memory system sensitivity of system sensitivity of overload data queue/query at overload of source is approached by VM allocation which handles the deadline service delay enhancing the performance leading to better efficiency than those of the existing systems. This is due the BURSE approach of predictive and reactive technique of problem detection and analysis in a different perspective.

Future work in the RTDB system can be modeled by reduction of VM, allocating two or more system process to one VM and also by a different approach for more effective algorithm to enhance control mechanism and data service.

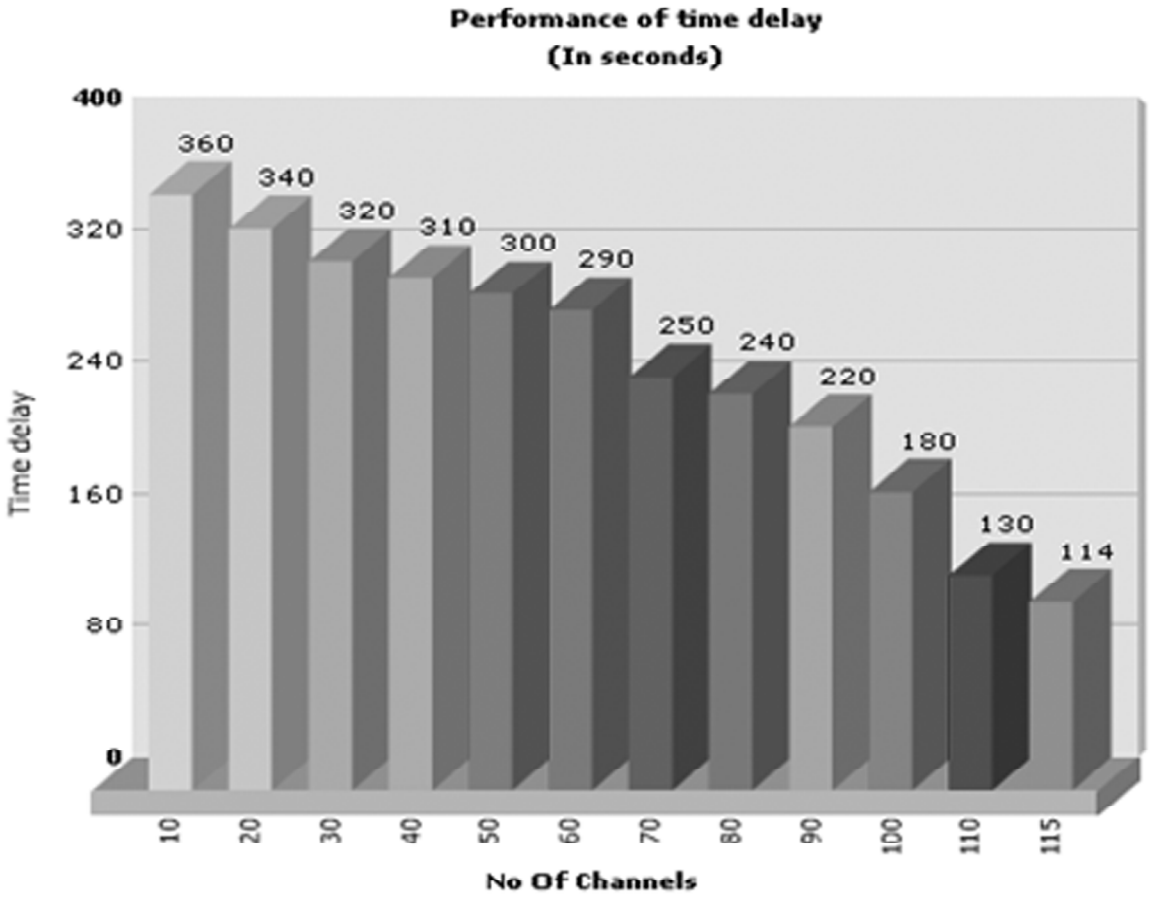

Figure 3: Normal Flow-Time Delay

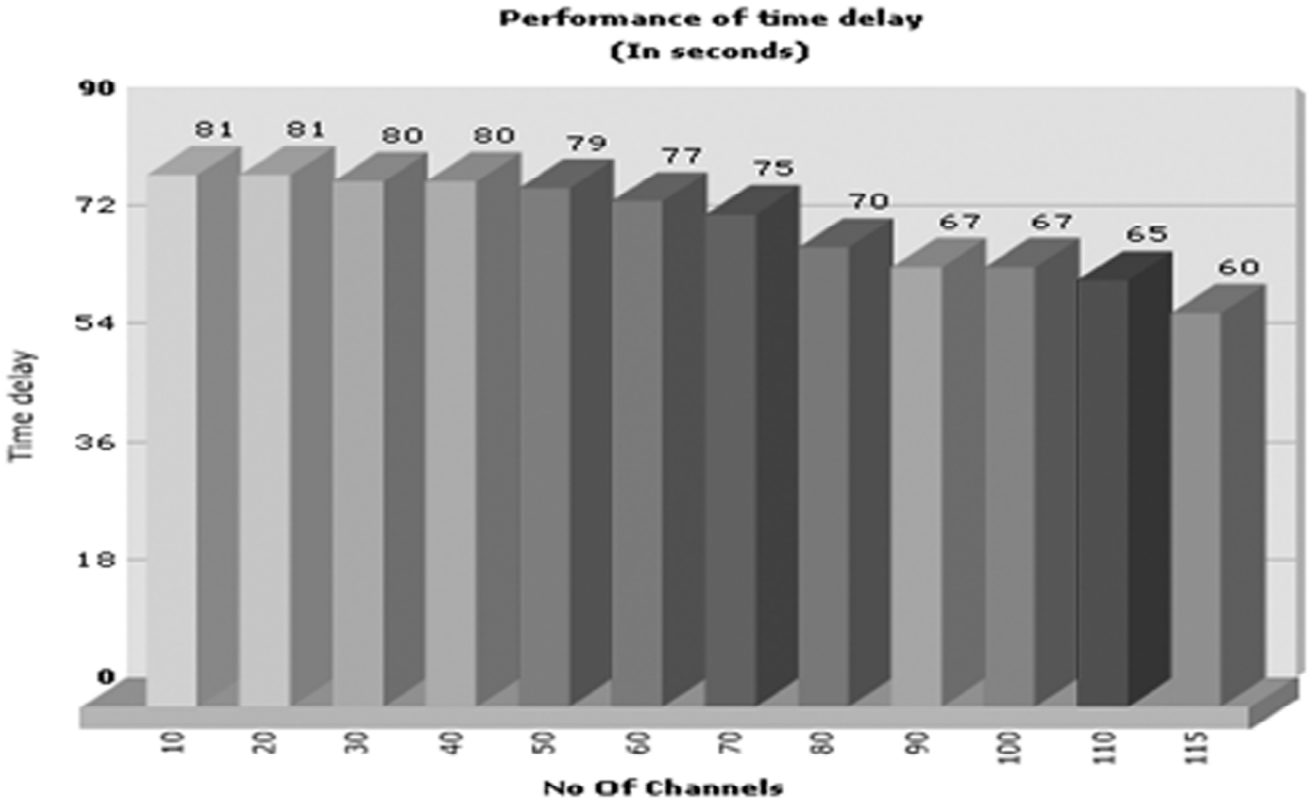

Figure 4: DPS Flow-Time Delay

REFERENCES

[1] Y. Zhou and K.D. Kang, "Deadline Assignment and Tardiness Control for Real-Time Data Services," in *Euromicro Conferenceon Real-Time Systems*, 2010.

[2] M. Amirijoo, J. Hansson, S. H. Son, and S. Gunnarsson, "Robust Quality Management for Differentiated Imprecise Data Services," in *the 25th IEEE Real-Time Systems Symposium*, 2004.

[3] M. Amirijoo, N. Chaufette, J. Hansson, S. H. Son, and S. Gunnarsson, "Generalized Performance Management of Multi-Class Real-Time Imprecise Data Services," in *the 26th IEEEInternational Real-Time Systems Symposium*, 2005.

[4] M. Amirijoo, J. Hansson, and S. H. Son, "Specification and Management of QoS in Real-Time Databases Supporting Imprecise Computations," *IEEE Transactions on Computers*, vol. 55, no. 3, pp. 304-319, 2006.

[5] K. D. Kang, S. H. Son, and J. A. Stankovic, "Differentiated Real-Time Data Services for E-Commerce Applications," *ElectronicCommerce Research*, vol. 3, no. 1-2, January/April 2003, combined Special Issue: Business Process Integration and ECommerce Infrastructure.

[6] K. Ramamritham, S. H. Son, and L. C. Dipippo, "Real-Time Databases and Data Services," *Real-Time Systems Journal*, vol. 28, Nov.-Dec. 2004.

[7] J. W. S. Liu, *Real-Time Systems*. Prentice Hall, 2000.

[8] C. L. Phillips and H. T. Nagle, *Digital Control System Analysis and Design (3rd edition)*. Prentice Hall, 1995.

[9] J. L. Hellerstein, Y. Diao, S. Parekh, and D. M. Tilbury, *Feedback Control of Computing Systems*. A John Wiley and Sons, Inc., Publication, 2004.

[10] C. Lu, J. A. Stankovic, G. Tao, and S. H. Son, "Feedback Control Real-Time Scheduling: Framework, Modeling and Algorithms," *Real-Time Systems, Special Issue on Control-TheoreticalApproaches to Real-Time Computing*, vol. 23, no. 1/2, May 2002.

[11] C. Lu, J. Stankovic, G. Tao, and S. Son, "Design and Evaluation of a Feedback Control EDF Algorithm," in *Real-Time SystemsSymposium*, December 1999.

[12] L. Bertini, J. Leite, and D. Mosse, "Statistical QoS Guarantee and Energy-Efficiency in Web Server Clusters," in *the 19thEuromicro Conference on Real-Time Systems*, 2007.

[13] "Oracle Berkeley DB Product Family," available at http://www.oracle.com/database/berkeley-db/index.html.

[14] K. Y. Lam and T. W. Kuo, Eds., *Real-Time Database Systems*. Kluwer Academic Publishers, 2006.

[15] M. Xiong and K. Ramamritham, "Deriving Deadlines and Periods for Real-Time Update Transactions," *IEEE Transactionson Computers*, vol. 53, no. 5, pp. 567–583, 2004.

[16] R. Ramakrishnan and J. Gehrke, *Database Management Systems*, 3rd ed. McGraw-Hill, 2003.

[17] "Yahoo! Finance," http://finance.yahoo.com/.

[18] "Transaction processing performance council," http://www.tpc.org/.

[19] C. Lu, T. F. Abdelzaher, J. A. Stankovic, and S. H. Son, "A Feedback Control Approach for Guaranteeing Relative Delays in Web Servers," in *Real-Time Technology and Applications Symposium*, 2001.