# Real-Time Database Systems in the New Millenium

Jayant R. Haritsa* Krithi Ramamritham†

## 1 Introduction

A variety of real-time applications, covering diverse scientific, financial, communication, manufacturing and military requirements, have to interface with large amounts of data. For example, the 1-800 telephone service in the United States receives thousands of calls per minute, each of which requires performing a complex query on a large distributed database in order to convert the virtual number dialed by the customer into a real telephone number that is closest to the customer's calling location. This query has to be completed in a timely manner, otherwise the customer may lose patience and hang up the phone. Administrative queries and updates to the distributed database have to be concurrently processed and the database has to be, of course, kept consistent during all these various operations.

Real-time database systems (RTDBS) are designed to address this challenge of *simultaneously* enforcing data integrity constraints and satisfying application timing constraints. Studies on RTDBS have been underway for over a decade, with the first major publication in the area being the classical paper of Abbott and Garcia-Molina [1], which appeared in the 1988 VLDB Conference and received the Best Paper award. During this time, a mature body of research has emerged, signaling the fusion between the real-time and database communities (see [11, 9, 14, 2, 7] to track the evolutionary trajectory of RTDBS research). A limitation of this research, however, is that it has been mostly devoted to extending traditional transaction processing issues such as resource scheduling policies, concurrency control, memory management, etc., to the real-time environment. As a consequence, several important aspects have remained largely unaddressed, especially in light of recent information technology trends. Therefore, we felt that a special issue of the Real-time Systems journal that focussed on these lacunae would be of value at this time, when we are entering the new millenium.

We solicited technical papers on a variety of new topics, including theoretical frameworks that substantiate the experimental results of earlier simulation-based performance studies; processing techniques for environments where the database includes objects that have limited temporal validity; strategies for efficiently integrating data security into the RTDBS framework; design and

*SERC, Indian Institute of Science, Bangalore 560012, India. Email: haritsa@dsl.serc.iisc.ernet.in

†Dept. of Computer Science, Univ. of Massachusetts, Amherst, MA 010003, USA. Email: krithi@cs.umass.edu

evaluation of RTDBS protocols for supporting electronic commerce applications; meeting timing constraints in mobile databases with their unique features of wireless clients, low power and limited communication bandwidth; integration of OLAP technologies such as database mining and warehousing with "imprecise" real-time query processing techniques/ comprehensive RTDBS architectural designs for major applications such as telecommunications, network management, flexible manufacturing, etc.

### 1.1 Papers in this Issue

In response to the above call-for-papers, we received submissions that covered a range of applications including temporally-consistent transaction processing in sensor databases, indexing for digital libraries, versioning algorithms and service provisioning in telecommunication databases, control systems for factory automation, flexible security policies that permit sophisticated tradeoffs between security and real-time requirements, learning-based techniques for scheduling in flexible manufacturing systems, and resource scheduling for multimedia servers.

The received papers were subjected to a comprehensive review process and three were finally selected to appear in this special issue. The first paper, "Triggered Updates for Temporal Consistency in Real-Time Databases", by Ahmed and Vrbsky, considers real-time monitoring applications wherein the current state of real-world objects is kept track of through sensors whose output is stored in the RTDBS. For this environment, the paper describes a technique to improve the temporal consistency of *derived objects*. A derived object's value is dependent on the values of associated sensor data and it is updated sporadically. Their technique is designed to ensure that maintaining temporal consistency does not result in a significant deterioration of the real-time performance with respect to meeting transaction deadlines. Its efficacy is evaluated through a detailed simulation study.

While data concurrency control has been extensively researched in the RTDBS literature, there is virtually no work on *index* concurrency control, [5, 6] being the exceptions. This is in spite of the fact that we can expect RTDBS to make extensive use of indexes to quickly access the base data and thereby enhance their ability to meet transaction deadlines. While the earlier work considered single-key index operations, the second paper in this issue, "Real-time Access Control and Reservation of B-tree Indexed Data", by Kuo and Lam, moves on to developing efficient real-time processing algorithms for *batch operations*, wherein a *set* of keys are handled at a time. Batch operations are relevant for applications such as digital libraries and on-line newspapers. The paper also presents techniques to minimize the number of index-related priority inversions for critical real-time transactions. The performance of the proposed algorithms is evaluated through a simulation study.

Effective scheduling of real-time transactions is greatly hampered by the large variance that typically exists between average-case and worst-case execution times in a database system [12].

This unpredictability arises due to a variety of factors, including data-value dependent execution patterns, blocking or restarts due to data contention, and disk latencies that are a function of the access history. The third paper in this issue, “Improving Predictability of Transaction Execution Times in Real-Time Databases” by Rastogi, Seshadri, Bohannon, Leinbaugh, Silberschatz and Sudarshan, attempts to eliminate some of these sources of unpredictability and thereby facilitate better estimation of transaction running times. In particular, they employ main-memory database technology and present new logical and physical versioning techniques that ensure that read-only transactions are *completely* isolated from update transactions with respect to data contention. These techniques have been successfully implemented in both research and commercial systems.

### 1.2 Research Agenda

While, as mentioned above, we did receive several submissions covering a wide range of topics, an unfortunate aspect is that there were *none* that dealt with electronic commerce, mobile databases or OLAP technologies. We feel that these applications represent compelling areas in which to showcase RTDBS technology and therefore strongly encourage researchers, especially students scouting for thesis topics, to take them up for investigation. Focusing on these new applications will also help to spur renewed interest in RTDBS research, which has seen declining activity in recent years as compared to the heydays of the early nineties.

As a case in point, consider the extremely popular “electronic auctions” on the Web, whose requirements appear to mesh perfectly with several of those postulated for RTDBS research. Firstly, there are time constraints associated with transaction completion. For example, in the Flash Auction at `http://www.firstauction.com`, to guarantee that one’s bid is registered, it must be received at the auction site within *five minutes* of the registry of the previous bid. Secondly, the data has temporal consistency constraints – bid updates have to be “publicized” to the remaining bidders within the expiry of the bid validity period, otherwise the data is rendered useless. Thirdly, the information about bid updates should ideally be received by all participating bidders at the same time, so that no individual bidder obtains an unfair advantage due to early receipt – that is, the information dissemination should appear to be an atomic action, analogous to distributed transaction commit. Finally, there is also the security aspect – typically, the auction database contains secret information such as bidders personal details including private keys, credit-worthiness and past bidding patterns; the purchase price and ownership history of the items that are being auctioned; the list of “in-house bidders”, who are bidders planted by the auction house to provoke other bidders by artificially hiking the maximum bid; etc. It is expected that the secret information is known only to the auctioneers since a bidder who is able to gain access to this information derives financially lucrative advantages over other competitors. However, a particularly devious security loophole involves establishment of a “covert channel” [8] between a corrupt auctioneer and a conspiring bidder. In a covert channel, information is passed on through *indirect* means and nothing "obviously illegal" is done in the process by the conspiring transactions, making these channels extremely difficult to detect. The challenge here is to ensure that the security mechanisms put into place to plug such loopholes should not seriously impact the real-time performance.

### 1.3 Closing Remarks

Real-time database systems have been recognized as a vital technology for a host of important application areas – for example, see [10] for a detailed analysis of its relevance to avionics. However, although research and development of RTDBS predates applications such as data mining and warehousing which have become household words, it has been hampered in realizing its full potential due to a variety of factors. Apart from several fundamental misconceptions about the real-time aspects of databases, highlighted in the recent article by Stankovic et al [13], major causes include the lack of clearly-defined "killer apps", the almost complete absence of foundational results, and the dearth of industry benchmarks.

On the bright side, however, the future looks promising with the recent appearance of commercial RTDBS products such as EagleSpeed-RTDB [3] and Clustra Parallel Data Server [4]. The challenge now is to percolate the rich body of available research to such systems. As a case in point, in the quest for more deterministic processing, EagleSpeed-RTDB has chosen to revert to the early database technologies of the network data model and static two-phase-locking for concurrency control! An alternative would be to use, for example, the versioning techniques of the Rastogi et al paper.

In closing, we hope that readers of this special issue will find the articles both timely and valuable, and that it will spur further research in the real-time database arena, especially in the areas highlighted in this introduction.

### Acknowledgements

The work of Jayant Haritsa was supported in part by research grants from the Dept. of Science and Technology and the Dept. of Bio-technology, Govt. of India.

## References

[1] R. Abbott and H. Garcia-Molina, "Scheduling Real-time Transactions: A Performance Evaluation", *Proc. of 14th Intl. Conf. on Very Large Data Bases (VLDB)*, September 1988.

[2] A. Bestavros, K. Lin, and S. Son, editors, "Real-Time Database Systems: Issues and Applications", Kluwer Academic Publishers, 1997.

[3] Lockheed Martin Corporation, "EagleSpeed RTDB User's Guide (Version 2.2)", *http://www.lmco.com/orss/eaglespeed*, July 1998.

[4] "Clustra DBMS White Paper", *http://www.clustra.com.*

[5] B. Goyal, J. Haritsa, S. Seshadri and V. Srinivasan, "Index Concurrency Control for Firm Real-Time Database Systems", *Proc. of 21st Intl. Conf. on Very Large Data Bases (VLDB)*, September 1995.

[6] J. Haritsa and S. Seshadri, "Real-Time Index Concurrency Control", *IEEE Trans. on Knowledge and Data Engineering*, 2000 (in press)

[7] Tei-Wei Kuo and Kam-yiu Lam, editors, "Real-Time Databases: Issues and Design", Kluwer Academic Publishers (in press).

[8] W. Lampson, "A Note on the Confinement Problem", *Comm. of ACM*, October 1973.

[9] K. Ramamritham, "Real-Time Databases", Intl. Journal of Distributed and Parallel Databases, 1993.

[10] M. Roark, M. Bohler and B. Eldridge, "Embedded Real-Time and Database: How Do They Fit Together", *Software Technology Conference*, 1996.

[11] S. Son, editor, "Real-Time Databases", *ACM Sigmod Record*, 17(1), 1988.

[12] J. Stankovic and W. Zhao, "On Real-Time Transactions", *ACM Sigmod Record*, 17(1), 1988.

[13] J. Stankovic, S. Son and J. Hansson, "Misconceptions about Real-Time Databases", *IEEE Computer*, 17(1), June 1999.

[14] O. Ulusoy, "Research Issues in Real-Time Database Systems", *Tech. Report BU-CEIS-94-32*, Dept. of Computer Engineering and Information Science, Bilkent University, Turkey, 1994.