Report on First International Workshop on Real-Time Database Systems

Azer Bestavros[†] (best@bu.edu)
Kwei-Jay Lin[‡] (klin@uci.edu)
Sang Son[§] (son@cs.virginia.edu)

1 Introduction

On March 7 and 8, 1996, the First International Workshop on Real-Time Databases (RTDB'96) was held in Newport Beach, California. There were about 50 workshop participants from 12 countries. Twenty two papers were presented and actively discussed in the 2-day single-track 6-session technical program. There were also two panel sessions to assess the state of RTDB research. In this report, we provide highlights of the workshop.[2]

2 Temporal Constraints and Issues

The first session aimed at solidifying the notions of temporal constraints so that workshop participants may achieve a common agreement on the temporal constraint aspects of RTDB. Three papers were presented, each considering the origins and implications of temporal constraints in a specific environment or for a specific platform, including feedback control process control systems, Active Rapidly Changing data Systems, and real-time databases in which transaction deadlines are implied from temporal constraints on data.

Several approaches to maintain data temporal consistency were mentioned during the presentations and during the discussions that ensued, including determining the periods for sensor transactions, data version selection, forcing user transactions to delay for a more up-to-date version, *etc.* One of the messages that came out of this session is that using traditional concurrency control techniques may not be appropriate for active RTDB because the effect of transaction "chaining" is not adequately addressed. Another open problem is the issue of how updates from the environment (through sensors) should be scheduled against user transactions and other updates. Finally, the merits of relating transaction scheduling decisions to transaction completion probabilities were discussed.

[†] Computer Science Department, Boston University.
[‡] Elect. & Comp. Eng. Dept., University of California, Irvine.
[§] Computer Science Department, University of Virginia.

[2] For a more detailed RTDB'96 workshop summary, we refer the reader to Technical Report TR-96-013 from Boston University at `http://cs-www.bu.edu/techreports/96-013-rtdb96-report`. The RTDB'96 program and proceedings are available on the WWW from `http://www.eng.uci.edu/ece/rtdb/rtdb96.html`

3 Real-Time Database Applications

The second session was chaired by John Stankovic, who opened the session with a "challenge" to the RTDB community—to make the case for RTDB technology by showing the *"value-added"* it brings to as many applications as possible. In a later panel, he presented a list of application domains that came up during the various sessions and added his belief that there are many more.

Four papers were presented in this session. Each one of these presentations overviewed the constraints imposed by a particular application domain and how these constraints were accomodated in a RTDB design targeted at that domain. The RTDB applications considered represented telecommunication systems, process-control systems, real-time simulation systems, and Web-based systems.

One of the points made during the presentations was that as any engineering practice, RTDB systems must provide the *"best compromises"* as opposed to the *"most elegant solutions"*. One of the messages that came out of the discussion that followed was that sometimes deadlines are set based on what the system can do (in other words, if the system is built to be twice as fast, then laxities will be halfed). Another point made during the presentations was that researchers must look at the notion of *cost* in a more pragmatic way (*e.g.* the learning curve of new techniques and the man/machine interaction issues are important for systems that involve a "person in the loop").

4 Scheduling Techniques for RTDB

Four papers were presented in this session. One of the promising research techniques discussed in the first paper is *performance polymorphism* for RTDB systems, whereby ideas from imprecise computation research (*e.g.* FLEX language) are applied to declarative query languages and to query optimization techniques. Another interesting dimension of scheduling for RTDB systems is that of *overload management*. The second paper of the session showed evidence that *admission control* is much more important than other RTDB resource management techniques (*e.g.* concurrency control and scheduling). In particular, admission control makes simple concurrency control protocols (*e.g.* 2PL-HP protocols) perform as well as sophisticated ones (*e.g.* Wait-50 and SCC protocols).

During the discussion that followed these presentations, there were questions about the "cost" of elaborate RTDB protocols versus the cost of admission control. The motivation of the question was that RTDB must be *light weight* and that unless solutions we provide are easily portable, they will never be implemented in real-systems. Krithi Ramamritham intervened to give evidence of this from concurrency control research, where hundreds of algorithms are proposed, but only few (mainly 2PL) are implemented in real systems. In reference to admission control and overload management techniques in general, Azer Bestavros suggested that these techniques in effect reduce the overall overhead, because they make the use of other complicated real-time protocols unnecessary! Another point brought up was that the techniques must be possible to add as a *layer* on top of existing off-the-shelf DB systems. Admission control and overload management is a good candidate layer.

To conclude, the session Chair Al Mok challenged the participants to identify a set of properties that are worth proposing as *standards* for RTDB (akin to the ACID properties of traditional DBMS). The discussion that ensued questioned whether a single model will ever be possible, given the richness of RTDB systems—a richness that comes from the disparity of application requirements as pointed out by John Stankovic. Another thread in this discussion had to do with the *quantification* of predictability and perhaps the use of a probabilistic model (akin to the ideas from the third paper of the session) or the use of values and various QOS guarantees (akin to the ideas from the second paper of the session).

5 RTDB Models and Benchmarks

This session consisted of three paper presentations. The first paper addressed an important (and often neglected) aspect of RTDB systems—that of query optimization. It proposed an elegant paradigm for cost modeling based on the relative "cost" of various operations—a modeling that abstracts away many of the details of the underlying machinery and algorithms. The discussions that ensued focused on two possible extensions to the cost modeling approach to better support RTDB systems, namely adding *absolute* timing analysis, and allow for *load parametrization* (*i.e.* to allow the estimation of cost to be parametrized based on the system load).

The third paper in this session offered a number of benchmarks that could be used to define the structure of an avionics database and test the various aspects of RTDB systems performance. The benchmarks were abstracted out from actual avionic systems. Seven test scenarios were defined ranging from periodic-readers-only to periodic-readers plus sporadic-readers plus periodic-updates. The paper fills in a void that was identified throughout the workshop—that of quantifying RTDB properties. The discussion that ensued reaffirmed the need for similar benchmarks for other applications.

6 RTDB Design Issues

The last two sessions of the workshop were dedicated to the presentation of a potpourri of new design and implementation techniques for RTDB and consisted of six papers.

Among the issues that were highlighted as worthy of further investigation was the problem of priority assignment for sub-transactions in distributed real-time databases. The presentation of K.Y. Lam suggested two such approaches: one is based on the real-time constraint of the base transaction, and the other based on the data contention with other transactions.

Another implementation optimization that stirred quite a bit of discussions was the use of lock-free objects to synchronize transactions in main-memory RTDB systems. This technique was suggested by James Anderson of the University of North Carolina as an implementation technique that gets rid of priority inversion by substituting blocking with efficient busy-waiting. The *hidden* overhead cost and reduced predictability were some of the concerns voiced regarding this technique.

Two new concurrency control protocols were also presented as candidates for improving the timeliness of transaction execution in RTDB systems. The first concurrency control protocol extends the read/write priority ceiling protocol by defining a compatibility table for all the methods defined for an object in an OO RTDB system, and then using this table to come up with an *affected set priority ceilings* that could be used to regulate the access to the object to ensure consistency. During the discussion, there were questions about the scalability and practicality of such an approach (and similar semantic-based techniques), because they depend on the ability of designers to identify the conflict modes. The second protocol attempts to reduce the hazards of blocking-based algorithms and restart-based algorithms by suggesting an algorithm that combines them (using alternative *shadows*). The work is similar to the Speculative Concurrency Control (SCC) work in RTSS'94 and VLDB'95.

Another issue that was identified by workshop participants as worthy of further investigation is the issue of recovery for RTDB systems. In a real-time system, the recovery of a database may be subject to timing constraints, which imply that during normal operation, activities such as logging and checkpointing must be performed in fashion that would guarantee that in the event of a failure, the database will be able to recover in a timely manner. The paper by Le Gruenwald presented a first step in studying this issue for main-memory RTDB. The main argument in the paper is to have the rate of data checkpointing related to the time-constraints associated with the data.

7 "Are we looking at the right issues?"

The first panel of the workshop was entitled "Are we looking at the right issues of RTDB?". The panel was moderated by Kwei-Jay Lin. Doug Locke first spoke of the application requirements for Aerospace RTDB systems. He emphasized that transactions that miss their deadlines must finish and argued for value-cognizant RTDB systems as opposed to the hard/firm deadline paradigm. Several workload requirements in the Air Traffic Control, Spacecraft Control, Training Simulation, *etc.* were presented and discussed. Lui Sha then talked about the need for an equivalent to the ACID properties for RTDB systems. He suggested that the notion of *stream data* is fundamental for RTDB. In other words, one may think of stream-data RTDB as different enough from traditional RTDB that deal with closed systems where changes to data are only carried by transactions from within the system. Brad Adelberg described the STRIP project at Stanford, which is aimed at financial applications. Much emphasis have been placed on the data update streams received by the database and how they can be effectively handled to provide a real-time view in the database. Krithi Ramamritham finally discussed the impact of RTDB technology in commercial products. He pointed out that while temporal and active database ideas have found their way into commerical products as well as into SQL, real-time database ideas have not.[3] One plausible reason is that developing time-cognizant extensions to database protocols requires a fairly substantial overhaul. The second reason is that a large proportion of the techniques developed thus far apply only to soft real-time constraints with the percentage of missed deadlines being the metric. This implies that the use of the protocols is intended more to improve performance and not for increasing functionality, unlike in temporal and active databases. He emphasized that RTDB researchers must aim at achieving greater predictability in real-time databases so that we get imporved perfomance as well as predictability that is quantifiable, the latter is a property that is not achievable simply by "faster hardware".

8 "Lessons learned and places to go"

The second panel of the workshop was entitled "What have we learned and where to go from here?" The panel was intended as a concluding session for the workshop. It was moderated by Program Co-Chair Sang Son from the University of Virginia, who pointed out that the RTDB'96 workshop was timely, since demand for advanced functionalities and timely management of data in new applications require practical solutions. He then asked each panelist whether the current research is on the right track and what are the remaining critical issues to be addressed.

John Stankovic first identified the key issues for RTDB systems, including predictability, fault-tolerance, and QOS for multimedia management. He argued that the technologies developed by the RTDB research community should show that RTDB systems can do significantly better than traditional approaches vis-a-vis properties such as cost, performance, functionality, and availability. It was generally agreed that we should focus on a few driving applications in which traditional DBMS cannot serve or are very inefficient to serve. He also pointed out that integrated solutions for distributed RTDB systems architecture are needed. Jane Liu talked about the lessons we learned: how to schedule transactions using the timing constraints and how to maintain temporal consistency of data. She emphasized that we need to utilize semantic information and different query processing methods for QOS management. She also felt that deciding on a small set of effective concurrency control algorithms is important. Before this can be accomplished, however, some benchmarks for RTDB systems must be developed. Kang Shin argued that we need to develop real systems, demonstrating usefulness using benchmarks and real applications. The first step is to form a consensus on terminology and concepts being used within the RTDB research community. He also discussed some technical issues that are yet to be addressed, including OS interface, ACID-equivalent properties for RT transactions, and fault-tolerance issues. Finally, Al Mok pointed out that there are strong motivations behind the ACID properties: granularity, consistency, non-interference, and failure semantics. He argued that we need to consider what should be the right characterization of the requirements for real-time transactions.

9 Conclusion

One of the goals of RTDB'96 was to create a forum for recent advances in real-time databases. We hoped to, and indeed we did, bring together researchers and engineers from academia and industry to explore the best ideas in real-time database systems research, and to evaluate the maturity and directions of real-time database systems technology.

At the conclusion of the workshop, most participants showed strong support to have the workshop continued in the future. Plans for a second workshop to be held in the fall of 1997 are underway. For more details, please check RTDB'97 Home Page on the WWW at `http://cs-www.bu.edu/pub/rtdb97/`.

[3] In the last session of the workshop, Janet Prichard updated workshop participants on the state of current real-time SQL standardization efforts. Since then a WWW Real-time SQL Home Page has been created and is accessible from `http://www.math.ecu.edu/rtsql`.