Replica Update Technique in RDRTDBS: Issues & Challenges

Pratik Shrivastava
Computer Science and Engineering
M.M.M.U.T.
Gorakhpur, India
pratik.shrivastav10@gmail.com

Uday Shanker
Computer Science and Engineering
M.M.M.U.T.
Gorakhpur, India
udaigkp@gmail.com

Abstract—The replicated distributed real-time database system (RDRTDBS) has gained great interest in providing highly available and scalable services. This technique provides the additional benefits of performance in terms of availability, fault-tolerance and reliability. These desired performance metrics are achieved via proper adaptation of replica update technique (RUT). In this paper, our aim is to discuss the state-of-art of RUTs, issues & current development, proposed approach & its significance and directions for future research in RDRTDBS.

Keywords—RDRTDBS, RUT, Availability, Fault-tolerance, Reliability

I. INTRODUCTION

The database system [1-3] is a collection of information that could access by many users at the same time. This system can be categorized as centralized or distributed. A centralized database system is confined to only a single location and consists of single processor with its associated devices. Centralized database system offers more reliability, less overhead and a single point of control whereas distributed database system (DDBS) is an evolved version of centralized database. A DDDBS [4,5] is a collection of sites which are connected through the communication network which provides extra benefits such as fault tolerance, reliability, etc.

Real time database system (RTDBS) provides the predefined features of traditional/conventional database along with enforcement of real time constraint [6-13]. This database is commonly used in real time computing environment where applications have strict requirement of consistency and deadline. Due to increasing demand of such database system, it requires predictable processing for the large amounts of temporal and non-temporal data objects [6-13]. Therefore, distributed real time database system (DRTDBS) is designed for effective processing of temporal and non-temporal data objects. There are several research papers published in different sub areas of DRTDBS such as concurrency control protocol [14], commit protocol [15,16] and other related issues [17] to improve its performance.

In DRTDBS, replication technique [18-41] is used to improve performance in terms of availability, scalability, reliability and fault-tolerance. In this technique, data copies are redundantly placed at multiple sites so that real time transaction can be executed locally. As per current scenario, data copies are either fully replicated, partially or not replicated. Apart from this, designing of RUT is a complex problem because access to data copies does not happen by a single site only. Rather, its accessibility spans over multiple sites having same data copy. Therefore, it is necessary to maintain the mutual consistency between the different replica sites holding the same data copy. This can be done via ordering the transaction operations at local and global level such that conflicting accesses can be prevented. Additionally, it must be ensured that all replica sites will eventually or immediately receive all updates as well. Overall, this whole explanation suggests that an important issue in replication technique is to design an effective and efficient RUT that follows a specific consistency model as per application requirement. The following sections discuss the research issues and challenges of RUT in RDRTDBS and present many stimulating problems. Meeting the challenges requires more extensive and coordinated research efforts towards quality of services (QoS), security constraint and dependency relationship.

II. BACKGROUND

Early efforts [18] done in designing of RUT for RDRTDBS exploited the semantic information of read only transaction to improve efficiency. Token based scheme for replication control is proposed in [19] and integrated version of replication protocol together with scheduling is presented in [20,21]. These all proposed protocols follow epsilon correctness criteria i.e. a weaker than one copy serializability (1SR). ClustRa [22], a parallel and distributed DBMS has been designed to provide real-time capabilities for telecom applications. However, such database does not guarantee QoS during overloaded situation. Apart from QoS, 1SR based replication protocol named as Managing Isolation in Replicated Real-time Object Repositories (MIRROR) is proposed in [23]. This protocol is augmented version of Classical Optimistic Two-Phase Locking (O2PL) with novel state-based real time conflict resolution mechanism. However, MIRROR suffers from the issue of deadlock, unbounded delay and overload. Just-In-Time Real-Time Replication (JITRTR) protocol designed for distributed real time object-oriented database presented in [24] performs well in static environment. However, it is unable to handle dynamic request. Solution of continues conflict detection and resolution based on conflict sets is proposed in [25]. But, eventually this solution also does not support strict consistency. Continuous convergence protocol for DRTDBS [26] has three main terms; local predictability, local consistency and eventual global consistency. This protocol has been designed to tolerate inconsistency for read transaction. Replication protocol PRiDe based on optimistic approach with deterministic detection and forward resolution of conflicts is designed in [27]. This protocol was meant to maintain the mutual consistency of the temporal data object but the Real Time Replication Control Protocol (RT-RCT) [28] works to manage the stability for non-temporal data objects. Simulator for examining the performance of different replication protocols/RUT for RDRTDBS is proposed in [29]. Virtual full replication based on adaptive segmentation [30] resolves the serious drawbacks of full replication but such approach also suffers from overloading issue. Recently, middleware-based replica control technique
(MBRCT) following 1SR is proposed in [31] for increasing the performance and scalability. This protocol is designed for multi masters and slave sites where master sites will execute write & update transactions and slave sites will execute only read transactions. These all such developed RUTs [18-41] offer better performance but they all deprived from security issues, QoS, dependency relationship and database independence.

III. PERFORMANCE ISSUES AND CHALLENGES

Conventional/traditional databases are not capable to meet the strict requirement of timeliness and predictability of RTS [42, 43]. Therefore, DRTDBS is widely used to manage temporal and non-temporal data objects of real-time system (RTS). DRTDBS is often distributed on a set of sites to fulfill the real time requirements of users. In a DRTDBS, data replication/transaction replication/merge replication or snapshot replication is widely used to increase availability, scalability, fault-tolerance and reliability [20,23,24].

Although, replication in DRTDBS is the most focused area for research, very little work is focused towards virtual full replication such as describe in [30]. Other side, replica consistency with respect to replicated data is stayed most explored area for research [18-41]. Because of being based on replica consistency, various performance metrics get affected such as availability, scalability, and reliability. That's why, maintaining the immediate or eventual consistency of replicated data via RUT has got more attraction for research. In addition to address past issues such as data consistency and availability, we strongly believe that conflict detection & resolution, scheduling and other operations [44,45] are still an active research area. Additionally, RDRTDBS will have to deal with the following additional problems generated from the obstacles imposed by its specific environment and applications.

A. Replica Update Techniques

Replication protocols/RUTs in RDRTDBS mainly focus on data consistency. In addition to this, it should also ensure that the replication system always presents the most current position under the existence of network or site failures. Existing RUTs [18-41] are optimized version of the database kernel. But such solutions are not independent of the database and they do not show good performance due to the factors involved such as asynchronousized master-slave sites, random transaction requests, variable local CPU scheduling time, data access conflict, network failure, etc. Comparatively lesser works have been done to make database independent with small overhead. Therefore, in RDRTDBS, our main objective is to shed light on the development of MBRCT. This protocol will be responsible to process with respect to conflict detection, resolution and schedule by following 1SR on the middleware. On the other hand, master sites will have to execute only write & update transactions and slave sites will have to execute only read transactions. Due to different place location for executing read and update transactions, it increases the opportunity to satisfy the real time constraint requirement of RDRTDBS. Overall, this approach increases the percentage for transaction completion within its deadline.

B. Dependencies Relationship

RDRTDBS adheres the sub-transaction model proposed in [46] that consists of coordinators, groups and their updaters. A transaction in RDRTDBS may execute in either sequential or parallel fashion. The difference depends on the way of executing the cohorts. Dependencies [47,48] is an effective means to express and coordinate the relationship between different sub transactions belonging to same coordinator or different coordinators. Dependency can be in the form of strong commit, weak abort, termination, exclusion etc. Previously developed RUTs [18-41] follow theirs defined transaction execution model through considering factors such as criticalness and slack time. However, no RUT is available today which expresses and coordinates dependencies between two or more subtransactions executing on same or different cohorts. Therefore, some light has to be shed in specifying and adapting proper measures with respect to different dependency relationship in RDRTDBS such that transaction miss ratio may get improved.

C. Security Constraint

RDRTDBS contains various types of important and confidential data. It is used by different types of users. In addition to the real-time barrier, RDRTDBS requires strict security constraints. Failure to adhere a security barrier can cause confidential disclosure of information. Hence, data security is more important than ever. It is very difficult to safely meet the triple requirements of deadline, consistency and security. Very little work has been done to support the security issue in DRTDBS [49-60]. To the best of my knowledge, none of the work supports the security constraint in RDRTDBS. Therefore, thinking about the existing and future real time applications, this area also requires attention for further research.

D. Quality of Services (QoS)

RDRTDBS can also be termed as a constrained system which has a stringent requirement of consistency and time. Due to high demand from real time applications, RDRTDBS can get easily overloaded. This overloading occurs when the processing time of the transactions exceeds its defined deadline [61-67]. The failure of meeting such deadlines causes serious damage to the system. In RDRTDBS, very little research work has been done to address the QoS [61-63]. Therefore, this area needs serious investigation to further improve performance.

IV. NEW RESEARCH DIRECTIONS

The future scope of RDRTDBS is very huge and diverse. Issues and challenges defined in the previous section requires more extensive research efforts such as design of new RCTs that make database independent without any expense of additional overhead, preventing covert channels between low and high priority transactions, configuration of dependency relationship between different sites (i.e. Coordinator, cohorts and updaters) and QoS in replicated real time environment. The main areas of research are listed below.

1) Existing RUTs [18-41] are the optimized version of database kernel and have their own pros and cons. This suggests that there is a need of new RUT that makes database independent and incurs very little overhead.
Additionally, there is also a need for designing the generalized system model that can be used for different real time application and can be easily configured in real time environment.

2) Incorporating security constraints in RDRTDBS is an untouched issue in replicated real time environment. In RDRTDBS, bypassing security constraint can cause unauthorized access via covert channels.

3) Guaranteeing QoS in overloaded situation is another prime area to be given due consideration because overhead involved for successfully completing all the transactions get increases. As a result, transaction miss ratio rises.

4) Dependency relationships such as strong commit, weak abort, termination dependency, exclusion dependency and so on play an important role in improving transaction performance in RDRTDBS. Therefore, specifying and coordinating dependency relationship between different sub-transaction executing on different locations in RDRTDBS need consideration such that system performance in terms of transaction miss ratio can be improved.

5) There is need of lesser complicated conflict detection and resolution strategy to maintain database consistency in RDRTDBS because existing conflict detection and resolution strategies are more complex and degrade system’s performance (i.e. chance to miss transaction deadline). In addition to above, it should be scalable to handle massive amount of data.

6) Predictability of RDRTDBS requires global system information. It is not an easy task to identify the information of whole system globally. This becomes more hectic when system grows from small scale to large scale. So, extensive research must be done in this direction such that global system information can be identified and appropriate actions can be taken to improve the system performance.

7) Deadlock is also one of the major issues in RDRTDBS. Appropriate algorithm should have to be designed to detect and avoid the deadlock. If by chance deadlock arises then recovery from deadlock must be done in an efficient way.

8) Communication delay is the major issue because sending and reception of messages should be done in the order and in a timely manner; otherwise, transactions may miss their deadlines.

9) In case of full replicated RDRTDBS, proper measures must be taken to minimize unnecessary bandwidth utilization and storage.

V. OUR CONTRIBUTION

Presently, there are various performance issues of RDRTDBS which include various existing policies in RUT, QoS, Covert Channel and Dependency relationship discussed above. In all such issues, RCTs is the most focused area for research where potential research avenues are present. The work done by us provides better solutions for some of the research issues mentioned in the section 3.

1) The performance of RDRTDBS depends on different factors such as distributed execution of the transaction [23], asynchronized master sites, unreliable communication channel, unbalanced transaction workload, weaker consistency model and heterogeneous master sites. Thus, these performance issues must be factored to minimize transaction miss ratio. In [31], MBRCT for RDRTDBS is employed that follow 1SR and fulfills time constraint property of real time transaction. This proposed MBRCT performs conflict detection, generates the schedule following 1SR, unicasts read transaction to the single master site and broadcasts update transaction to all master sites. Due to shifting of such overhead in middleware, maximum resource usage is done for transaction execution which means that more number of real time transactions can satisfy the time constraint demand. For improving the transaction response time, we also follow the thread to metadata [68] in place of thread to transaction model.

VI. CONCLUSION

A successful RDRTDBS provides a forum for business, education and research people. Because system that are capable of providing consistent data with availability and scalability become the solid grounds for the success of research and business need. In RDRTDBS, management of temporal and non-temporal data offers new challenging problems. Hence, RUTs development is the most important issue for RDRTDBS. Apart from this, due to growing interest in the real time application field several other issues have also been reported and analysed to improve the performance of RDRTDBS. The ultimate aim of such RUTs would be to maximize number of real time transactions to satisfy their time constraint requirement with strictly maintaining the mutual consistency between different replicas and alleviating the effects of transaction unpredictability. In the current reporting works, our primary focus is to explore promising challenges in RDRTDBS. A critical discussion has also been done on current open issues, future trends and directions including configuring different dependency relationships, guaranteeing QoS and preventing secret channels for unauthorized access.

REFERENCES


