RODAIN: A Highly Available Real-Time Main-Memory Database System

Tiina Niklander and Kimmo Raatikainen
University of Helsinki, Department of Computer Science
P.O. Box 26 (Teollisuuskatu 23), FIN-00014 University of Helsinki, Finland
Email: {tiina.niklander,kimmo.raatikainen}@cs.Helsinki.FI

Introduction

The RODAIN Database System has been designed to be used in telecommunications. It is a highly available real-time object-oriented database system [1, 2]. The main challenge is that typical transactions are heterogeneous: short simple queries, simple updates, and long massive updates. Further discussion on databases in telecommunications can be found in [3].

A RODAIN node consists of a primary node and of a mirror node that are symmetrical and have the same subsystems but their functionality depends on the role. The User Request Interpreter Subsystem (URIS) accepts client requests and forwards them to the TRansaction Process (TRP). Each TRP executes one transaction at time but can handle sequential execution of multiple transactions. The Run-time Transaction Controller (RTC) allocates a TRP for an arriving transaction. TRPs execute the transactions and access the database through Object Request Dispatcher (ORD). ORD sends transaction logs to Fault-Tolerance and Recovery Subsystem (FTRS) for storing. The FTRS on Primary Node communicates with the FTRS on Mirror Node to store the logs permanently in the Secondary Storage Subsystem (SSS). Watchdog monitors the other processes and initiates node recovery actions when needed.

Issues in High Availability

The requirement of high availability has lead us to use node replication. In RODAIN we have two identical nodes that can change their functionality according the changes in the other node. The whole system can tolerate failure of either one of the identical nodes. The remaining node will still maintain the system functionality and execute the transaction requests but with a lower level of performance.

The concept of multiple nodes requires some communication between the nodes. FTRS is responsible of the communication between Primary and Mirror Nodes. On the Primary Node FTRS sends the log records to the Mirror Node and receives the acknowledgements of commit records. On the Mirror Node it receives the log records and sends the acknowledgements. Since we assume the communication channel to be reliable and to have a bounded message transfer delay, one node can notice the failure of the other node simply from the missing messages.

The Watchdog subsystem exists to notice failures of other subsystems. It can start new transaction process if one has failed. Failure of any other subsystem except TRP requires a full restart of the node.

Third subsystem that might have features for high availability is the User Request Interpreter Subsystem. It may support failure masking from the client. If there is no support then a client notices the failure of Primary Node only from nonexisting reply to a request it has made. On the other hand URIS (on Mirror Node) can keep a list of all arrived transaction requests that have not yet been answered. Should the Primary fail, the URIS on Mirror Node can re-execute all the requests. Our architecture does not support migration of partially executed transactions. The transaction can be executed totally on the Mirror Node or the URIS can simply send indication of execution failure to the client.

In the case of Primary Node failure, Mirror Node becomes new Primary Node and starts transaction processing. The failed node always recovers to Mirror Node by loading database image and logs from permanent storage on SSS.

References

[1] J. Kiviniemi, T. Niklander, P. Porkka, and K. Raatikainen. Transaction processing in the rodain real-time database system. In A. Bestavros and V. Fay-Wolfe, editors, *Real-Time Database and Information Systems*, pages 355–375, London, 1997. Kluwer Academic Publishers.

[2] T. Niklander, J. Kiviniemi, and K. Raatikainen. A real-time database for future telecommunication services. In D. Gaïti, editor, *Intelligent Networks and Intelligence in Networks*, pages 413–430, Paris, France, 1997. Chapman & Hall.

[3] K. Raatikainen. Real-time databases in telecommunications. In A. Bestavros, K.-J. Lin, and S. H. Son, editors, *Real-Time Database Systems: Issues and Applications*, pages 93–98. Kluwer, 1997.