

# PERFORMANCE OF CONCURRENCY CONTROL MECHANISMS IN CENTRALISED DATABASE SYSTEMS

Manohar Srinivasan<sup>1</sup>

Department of cse, Arignar Anna Institute of Science and Technology, Pennalur,  
Chennai, Tamilnadu 602 105, India  
yesmanohar@gmail.com

Revathi Manohar<sup>2</sup>

Department of Computer Science, KMG College of Arts and Science, R.S. Road,  
Gudiytham, Tamilnadu 632 602, India  
revsoft86@gmail.com

## Abstract

The objective of this study is to propose user control system that definitely increases the performance of real-time data service. However the most existing work on this RTDB was based on very critical closed loop control system. To overcome this problem we design a system called user control database model which stimulate the overload transaction during run time. We also design quality of service scheme and it allows to setting requirements for QOS transaction. The performance of proposed algorithm evaluated in different experiments. So the proposed system will definitely satisfies our needs even critical conditions like overload or run-time errors.

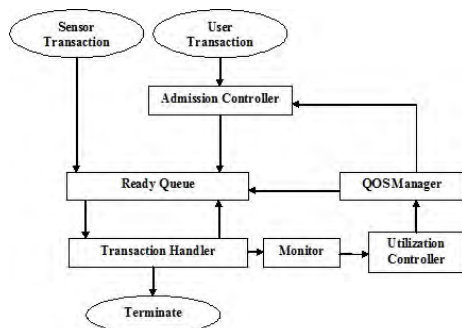
**Keywords:** RTDB; F2M; COMM API.

## 1. Introduction

Real-time systems are nowadays covering wide range of application domains like telecommunication, networking system, censor monitoring, traffic signals and e-commerce etc. The real-time databases (RTDBs) are not only needs to read and process the data, but also to proceed in a timely manner without producing unexpected errors. In traditional databases mostly become failure due to work load or resource contention. In this study, we propose a user control system approach it reduce run time errors during overload. We also design quality service during transaction.

## 2. Real-time Database Modeling and Performance Management

The following diagram illustrates the operation of real-time database modeling and measure the performance of concurrency control system.



In this approach we design three modules to manage real-time database functionalities.

- Temperature Identification (via Serial Communication Application Programming Interface –COMM API)

- Admin Control
- User Control

### **2.1. Temperature Identification and Transmission**

In this module we are going to get the temperature as an input for our real-time database. This input is getting from the F2M (female-to-male) cable and it is transmitted to the admin control. For this input transmission we are using serial communication Java API i.e (COMM API).

This temperature input will store the current input into the database as well as update it into the admin controller.

### **2.2. Admin Control**

In admin control module we are going to monitor the temperature as input, and it control the request from users. The temperature input is updated into the database as well as the output is monitor into the admin control as plotted graphs concurrently.

When the users sends request for that temperature, first we maintain a queue for that requests. That the queue is altered based on the success ratio. We also provide utilization controller for improving success ratio. Then the user gets the output temperature as plotted graphs.

### **2.3. User Control**

In this user control module the user will receive the output plotted graphs for current temperature. The user send request to admin control for getting temperature request. Also the user control will get plotted graphs at specific period of timing interval.

### **3. Measurement of QoS**

We also design quality of service (QoS) management scheme that allows the setting of several QoS requirements for transaction of equal importance is proposed. The performance of the proposed algorithm is evaluated in s different experiments. This simulation studies show that the proposed solution satisfies the requirements even during overloads and inaccurate run-time estimation errors. We show that our approach can expect the desire output and considerably enhancing the success ratio.

The success ratio is defined as follows:

$$SR = 100 \times (N_{\text{timely}} / N_{\text{submitted}}) (\%)$$

$N_{\text{timely}}$  is the number of transactions committed before their deadlines and

$N_{\text{submitted}}$  is the total number of transactions submitted to the RTDB.

### **4. Future Extraction**

Multi version concurrency control mechanism could reduce data conflicts in RTDBs. This investigation is reserved for future work.

### **References**

- [1] Mr. Jisu Oh and Mr. Kyoung-Don Kang: "An Approach for Real-time Database Modeling and Performance Management", 13<sup>th</sup> IEEE Real Time Technology and Application Symposium 2007.
- [2] R. Abbott and H. Garcia-Molina: "Scheduling Real-Time Transaction: A Performance Evaluation ACM Transaction on Database System", 17:513-560, 1992
- [3] Nicolas Chaufette: "Generalized Performance Management of Multi Class Real-Time Imprecise Data Services", June 5, 2006
- [4] Mehdi Amirijoo, Jorgen Hansson: "Specification and Management of QoS in Real Time Databases Supporting Imprecise Computations", March 2006.
- [5] Jayant R. Haritsa, Micheal J. Carey, and Miron Livny: "Value based Scheduling in Real-Time Database System", Oct 6 1992.
- [6] Piotr Krzyzagoski, "Concurrency Control in Real-Time Database Systems", 2000.
- [7] Chenyang Lu, John A Stankovic, Tarek F. Abdelzaher, Gang Tao, Sang H. Son and Michael Marley, "Performance Specifications and Metrics for Adaptive Real-Time Systems" December 2000.