A SEMINAR SYNOPSIS ON

Real-time Databases

Presented by

Aravind B 1ST14CS023

Real-time Databases

ABSTRACT

A real-time database is a database system which uses real-time processing to handle workloads whose state is constantly changing. This differs from traditional databases containing persistent data, mostly unaffected by time. For example, a stock market changes very rapidly and is dynamic. Real-time processing means that a transaction is processed fast enough for the result to come back and be acted on right away.

INTRODUCTION

Real-time databases are traditional databases that use an extension to give the additional power to yield reliable responses. They use timing constraints that represent a certain range of values for which the data are valid. This range is called temporal validity. A conventional database cannot work under these circumstances because the inconsistencies between the real-world objects and the data that represents them are too severe for simple modifications. An effective system needs to be able to handle time-sensitive queries, return only temporally valid data, and support priority scheduling. To enter the data in the records, often a sensor or an input device monitors the state of the physical system and updates the database with new information to reflect the physical system more accurately. When designing a real-time database system, one should consider how to represent valid time, how facts are associated with real-time system. Also, consider how to represent attribute values in the database so that process transactions and data consistency have no violations.

When designing a system, it is important to consider what the system should do when deadlines are not met. For example, an air-traffic control system constantly monitors hundreds of aircraft and makes decisions about incoming flight paths and determines the order in which aircraft should land based on data such as fuel, altitude, and speed. If any of this information is late, the result could be devastating. To address issues of obsolete data, the timestamp can support transactions by providing clear time references.

PRESERVING DATA CONSISTENCY

Although the real-time database system may seem like a simple system, problems arise during overload when two or more database transactions require access to the same portion of the database. A transaction is usually the result of an execution of a program that accesses or changes the contents of a database. A database must let only one transaction operate at a time to preserve data consistency. In real-time databases, deadlines are formed and different kinds of systems respond to data that does not meet its deadline in different ways.

In a real-time system, each transaction uses a timestamp to schedule the transactions. A priority mapper unit assigns a level of importance to each transaction upon its arrival in the database system that is dependent on how the system views times and other priorities. The timestamp method on relies on the arrival time in the system. Researchers indicate that for most studies, transactions are sporadic with unpredictable arrival times. For example, the system gives an earlier request deadline to a higher priority and a later deadline to a lower priority. Below is a comparison of different scheduling algorithms.

Earliest Deadline

PT = DT — The value of a transaction is not important. An example is a group of people calling to order a product.

Highest Value

PT = 1/VT — The deadline is not important. Some transactions should get to CPU based on criticalness, not fairness. This is an example of least slack that can wait the least amount of time. If the telephone switchboards were overloaded, people who call 911 should get priority.

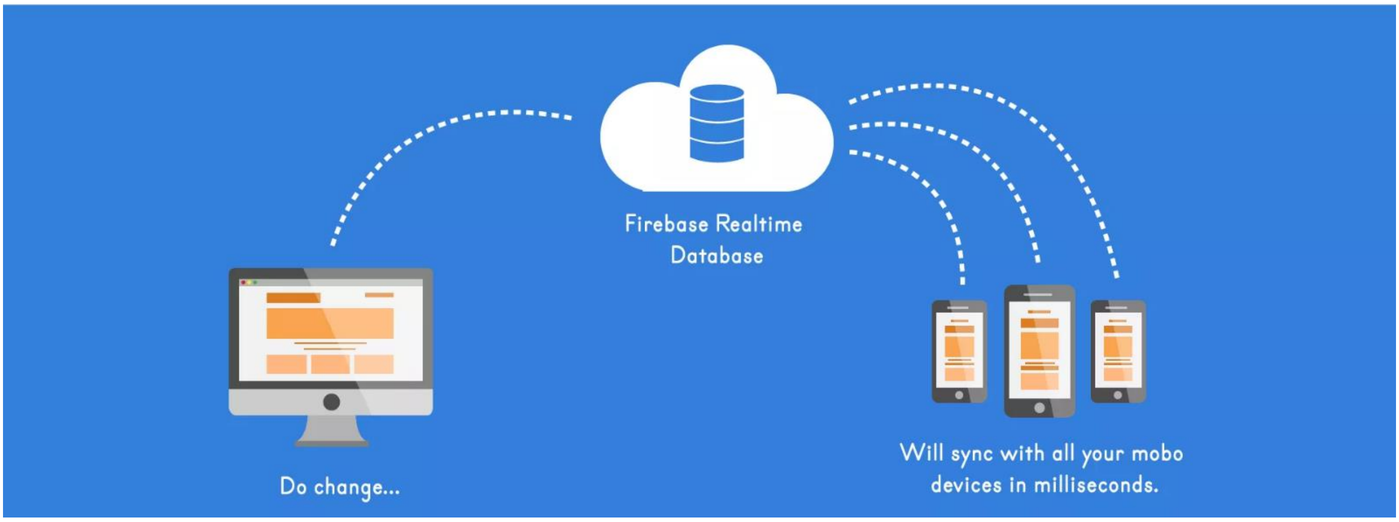
Value inflated deadline

PT = DT/VT — Gives equal weight to deadline and values based on scheduling. An example is registering for classes where the student selects a block of classes that he wishes to take and presses submit. In this scenario, higher priorities often take up precedence. A school registration system probably uses this technique when the server receives two registration transactions. If one student had 22 credits and the other had 100 credits, the person with 100 credits would take priority (Value based scheduling).

ADVANTAGES

Traditional databases are persistent but are incapable of dealing with dynamic data that constantly changes. Therefore, another system is needed. Real-time databases may be modified to improve accuracy and efficiency and to avoid conflict, by providing deadlines and wait periods to insure temporal consistency.

Real-time database systems offer a way of monitoring a physical system and representing it in data streams to a database. A data stream, like memory, fades over time. In order to guarantee that the freshest and most accurate information is recorded there are a number of ways of checking transactions to make sure they are executed in the proper order. An online auction house provides an example of a rapidly changing database.


Firebase Realtime Database
Do change...
Will sync with all your mobo devices in milliseconds.

EXAMPLE

Firebase Realtime Database

Firebase provides a real-time database and backend as a service. The service provides application developers an API that allows application data to be synchronized across clients and stored on Firebase's cloud.

CONCLUSION

Realtime databases enables new technologies such as web-video conferencing and instant messenger conversations in sound and high-resolution video, which are reliant on real-time database systems.