

## **EXTENSIVE REVIEW OF DATABASE SYNCHRONIZATION**

Khushpreet Kaur<sup>1</sup> & Harpreet Kaur<sup>2</sup>

**Abstract-** A lot of research has been done under the subject of data synchronization. To the best of our knowledge, little or no publications exist that review research problems faced under this field. In this paper we will summarize the current issues in synchronization of data. This paper will identify current research gaps and open areas for further research.

**Keywords –** Database, Synchronization, DBMS, Client, Server.

### **1. INTRODUCTION**

Formally, "database" [1] refers to the data themselves and supporting data structures. Databases are created to operate large quantities of information by inputting, storing, retrieving and managing that information. Databases are set up so that one set of software programs provides all users with access to all the data.

Traditional databases are organized by fields, records, and files. A field is a single piece of information; a record is one complete set of fields; and a file is a collection of records. For example, a telephone book is analogous to a file. It contains a list of records, each of which consists of three fields: name, address, and telephone number.

An alternative concept in database design is known as Hypertext. In a Hypertext database, any object, whether it is a piece of text, a picture, or a film, can be linked to any other object. Hypertext databases are particularly useful for organizing large amounts of disparate information, but they are not designed for numerical analysis.

A database is an organized collection of data. The data are typically organized to model relevant aspects of reality in a way that supports processes requiring this information. For example, modeling the availability of rooms in hotels in a way that supports finding a hotel with vacancies.

Databases are used to support internal operations of organizations and to underpin online interactions with customers and suppliers. Databases are used to hold administrative information and more specialized data, such as engineering data or economic models. Examples of database applications include computerized library systems, flight reservation systems and computerized parts inventory systems.

A database management system (DBMS) is an aggregate of data, hardware, software, and users that help an enterprise manage its operational data. The main function of a DBMS is to provide efficient and reliable methods of data retrieval to many users. If our college has 10,000 students each year and each student can have approximately 10 grade records per year, then over 10 years, the college will accumulate 1,000,000 grade records. It is not easy to extract records satisfying certain criteria from such a set, and by current standards, this set of records is quite small. Given the current concern for "grade inflation", a typical question that we may try to answer is determining the evolution of the grade averages in introductory programming courses over a 10-year period. Therefore, it is clear that efficient data retrieval is an essential function of database systems.

Database system is a system to achieve an organized, store a large number of dynamical associated data, facilitate for multi-user accessing to computer hardware, software and data, that it is a computer system with database technology.

Most DBMSs deal with several users who try simultaneously to access several data items and, frequently, the same data item. For instance, suppose that we wish to introduce an automatic registration system for students. Students may register by using terminals or workstations. Of course, we assume that the database contains information that describes the capacity of the courses and the number of seats currently available. Suppose that several students wish to register for cs210 in the spring semester of 2014. Unfortunately, the capacity of the course is limited, and not all demands can be satisfied. If, say, only one seat remains available in that class, the database must handle these competing demands and allow only one registration to go through.

#### *1.1 Data synchronization*

A distinctly different (but related) concept [2] is that of data synchronization. This refers to the need to keep multiple copies of a set of data coherent with one another. Data synchronization is the process of establishing consistency among data from a source to target data storage and vice versa and the continuous harmonization of the data over time.

Data synchronization technologies are designed to synchronize a single set of data between two or more devices, automatically copying changes back and forth. For example, a user's contact list on one mobile device can be synchronized with other mobile devices or computers. Data synchronization can be local synchronization where the device and computer are side-by-

<sup>1</sup> Department of Computer Science Engineering, Bahra Group of Institutions, Patiala Campus, Punjab, India

<sup>2</sup> Department of Computer Science Engineering, Bahra Group of Institutions, Patiala Campus, Punjab, India

side and data is transferred or remote synchronization when a user is mobile and the data is synchronized over a mobile network.

Data synchronization is enabled through specialized software that tracks data versions as they are created and utilized. The process is implemented in distributed systems where data elements are routed between several computers or systems. Each computer may modify original data versions, depending on requirements.

Data synchronization ensures that regardless of data modifications, all changes are merged with the original data source. Data synchronization is also used in data mirroring, where each data set is exactly replicated or synchronized within another device.

Examples include:

- File synchronization, such as syncing a hand-held MP3 player to a desktop computer.
- Cluster file systems, which are file systems that maintain data or indexes in a coherent fashion across a whole computing cluster.
- Cache coherency, maintaining multiple copies of data in sync across multiple caches.
- RAID (redundant array of independent disks), where data is written in a redundant fashion across multiple disks, so that the loss of any one disk does not lead to a loss of data.
- Database replication, where copies of data on a database are kept in sync, despite possible large geographical separation.
- Journaling, a technique used by many modern file systems to make sure that file metadata are updated on a disk in a coherent, consistent manner.

## 2. LITERATURE REVIEW

Zucker et al. [4] had explained that data synchronization is required for supply chain management in the B2B e-commerce environment. This case study examined the impact of the adoption of data synchronization on three large consumer product goods organizations. The study identified process and structural inadequacies that developed as the result of the implementation, as well as how these organizations recognized benefits and future opportunities after data synchronization adoption. The findings revealed the significance of internal alignment around data cleansing and accuracy, as well as opportunities for improved external alignment from a systems perspective. Synergy created between product item management, data synchronization, and internal champions existed at all three companies. The workflow re-design, process improvements and standards development imposed on these organizations by the clean data requirement of data synchronization provided the greatest benefits from the data synchronization process.

Kim et al. [5] had explained that the embedded DBMS is a lightweight DBMS for effective management of quite small databases contained in tiny mobile devices. Synchronization is a core function of the embedded DBMS to preserve the consistency of data replicated in the server and client databases. This paper presents a framework for synchronization in embedded DBMS environment. He first address key issues for realizing synchronization, and then propose solutions to them obtained from our development. The main issues touched here are (1) classifying conflicts, (2) identifying changes in a client database, (3) detecting conflicts, and (4) resolving conflicts. The proposed framework would help reduce the trial-and errors of embedded DBMS developers in implementing their synchronization server.

Yi et al.[6] had explained that Hardware/Software cosimulation is the key process to shorten the design turn around time. We have proposed a novel technique, called virtual synchronization, for fast and time accurate cosimulation that involves interacting component simulators. In this paper, we further extend the virtual synchronization technique with OS modeling for the case where multiple software tasks are executed under the supervision of a real-time operating system. The OS modeler models the RTOS overheads of context switching and tick interrupt handling as well as preemption behavior. While maintaining the timing accuracy to an acceptable level below a few percents, we could reduce the simulation time drastically compared with existent conservative approach by removing the need of time synchronization between simulators. It is confirmed with a preliminary experiment with a multimedia example that consists of four real-life tasks.

Mazzenga et al. [7] had explained that for code-division multiple-access (CDMA) cellular communications synchronous operation offers significant advantages over asynchronous operation. This feature has been already included in second-generation systems, such as the CdmaOne based on the IS-95 standard. Nonetheless, synchronous operation does not seem to be assessed as a basic system option for third-generation systems. This is mainly due to the contrasted choice of having global positioning system (GPS) receivers located at any of the base transceiver stations (BTSs) to perform station synchronization. However, the existing CDMA One interface permits time transfer between the various BTSs and a simple time difference measurement technique can provide an internal, network-based means for time synchronization of base stations clocks. This paper considers a recently proposed approach based on the pilot strength measurements message delivered by every mobile station (MS), combined with round trip delay measurements that BTSs perform on the MS transmissions. In this paper, we consider some possible algorithms to reduce time misalignments, and continuously phase-lock the clocks at different BTSs. Having developed an appropriate analytical/simulation approach, in the environment of a CDMA network, the paper evaluates and compares the performance of two suitable centralized architecture configurations, i.e., master-slave synchronization (MSS) and mutual synchronization (MUS). Both network architectures turn out to be applicable and provide suitable timing

performance. Finally, the paper provides a preliminary discussion on some related issues, such as GPS avoidance, synchronization of BTSs inside buildings, tunnels, and subways and hybrid MSS/MUS synchronization network architectures.

Trachtenberg et al.[8] had explained that Modern Personal Digital Assistant (PDA) architectures often use a wholesale data transfer protocol known as “slow sync” for synchronizing PDAs with Personal Computers (PCs). This approach is markedly inefficient, in terms of bandwidth usage and latency, since the PDA and PC typically share many common records. We propose, analyze, and implement a novel PDA synchronization scheme (CPIsync) based on recent information-theoretic research. The salient property of this scheme is that its communication complexity depends only on the number of differences between the PDA and PC, rather than the overall sizes of their databases. Moreover, our implementation shows that the computational complexity of CPIsync is practical, and that the overall latency is typically much smaller than that of slow sync. Thus, CPIsync has potential for significantly improving synchronization protocols for PDAs and, more generally, for heterogeneous networks of many machines.

Stasicki et al. [9] had explained that PIV became a mature technology for flow field investigations in industrial research, see for instance Kompenhans et al. (1998), it has been increasingly challenged by more complicate applications. As far as static models in wind tunnels are concerned, the experiment does not need to be synchronized with the measuring instrument. However, for the examination of dynamic systems synchronization becomes crucial. Recently, the continuous but alternating and, with regard to the measurement system, asynchronous wake flow of a helicopter’s rotor blade was interfaced with a PIV system in order to investigate defined planes in the rotor blades frame of reference, see Raffel et al. (1998). This test is the first of two examples which are described in this paper. Within the framework of the European research program WAVENC (Wake Vortex evolution and wake vortex ENCounter, Brite/EuRam project BE97-4112) PIV has been applied to a discontinuous experiment, i.e. the wake flow of a free flying air-liner model in the catapult facility of ONERA, Lille. The goal was to provide an experimental database for the far-wake evolution of the trailing vortices. The data base will be used to validate CFD calculations. First results of the PIV measurements as well as a comparison with smoke visualization tests have already been presented by Dieterle et al. (1999a). A general description of the PIV experiments is given by Dieterle et al. (1999b). Beside these two applications a technical description of the synchronization hard- and software which has been developed in cooperation DLR in Göttingen and HARDsoft and which is distributed in license by PIVTEC will be presented.

Keller et al. [10] had explained that The effect of time-domain and frequency-domain synchronization errors is quantified in the context of various coherently and noncoherently detected 1, 2, and 4 bits/symbol OFDM constellations, in order to demonstrate the wide applicability of the techniques proposed for mitigating the bit error rate (BER) performance degradations inflicted. A reference symbol is proposed and a range of correlation techniques are suggested for coarse and fine synchronization. Their performance is studied over time-dispersive Rayleigh fading channels, with the conclusion that the proposed synchronization techniques result in virtually unimpaired BERs over the range of wideband channels investigated in comparison to a perfectly synchronized system.

Wiesmann et al.[11] had explained that data replication is an increasingly important topic as databases are more and more deployed over clusters of workstations. One of the challenges in database replication is to introduce replication without severely affecting performance. Because of this difficulty, current database products use lazy replication, which is very efficient but can compromise consistency. As an alternative, eager replication guarantees consistency but most existing protocols have a prohibitive cost. In order to clarify the current state of the art and open up new avenues for research, this paper analyses existing eager techniques using three key parameters. In our analysis, we distinguish eight classes of eager replication protocols and, for each category, discuss its requirements, capabilities, and cost. The contribution lies in showing when eager replication is feasible and in spelling out the different aspects a database replication protocol must account for.

Malhotra et al. [12] provided an algorithm to solve the problem that when all clients are relying on a single server. If that database becomes unavailable due to planned server downtime or from server failures, all of the remote workers will be disconnected from their data. Data is stored on their system (user system). When the user connected to the internet data automatically sink from their client system to the server in serial order. It also works on file handling. When the system is disconnected from network all the files(images) uploaded by user, saved on client machine folder when it is again connected with the server, automatically files(images) transferred from client to server.

Kaur et al. discussed [3] that a distributed database system allows applications to access data from local and remote databases. The objective of the research is to transfer data from client to server when the mode is online, stored as a backup data on the client machine when it is offline, backup data automatically sink from client to server in order when again it is online and also displayed the data available on server.

### 3. CONCLUSION

This paper proposes synchronization of client database with Server database. In this paper, we have managed to conduct review on synchronization between client and server-side databases effectively. Relevant literatures have been chosen for reviews and several aspects have been identified to be either directly or indirectly involved in synchronization.

### 4. REFERENCE

- [1] [www.cs.umb.edu/cs630/hd1.pdf](http://www.cs.umb.edu/cs630/hd1.pdf)
- [2] [en.wikipedia.org/wiki/Synchronization\\_\(computer\\_science\)](http://en.wikipedia.org/wiki/Synchronization_(computer_science))
- [3] Anit Kaur and Aashima(2014), "Synchronized Algorithm for Database and Image Processing Between Client and Server", IJCSIT, Vol. 1, Issue 6.
- [4] Susan G. Zucker & Shouhong Wang, "the impact of Data synchronization adoption on organizations: a Case study" Journal of Electronic Commerce in Organizations", Volume 7, Issue 3.
- [5] Sang-Wook Kim, "Synchronization in an Embedded DBMS Environment" IJCSNS International Journal of Computer Science and Network Security, VOL.6 No.7A, July 2006.
- [6] Youngmin Yi, Dohyung Kim, Soonhoi Ha," Virtual Synchronization Technique with OS Modeling for Fast and Time-accurate Cosimulation" CODES+ISSS '03, October 1-3, 2003, Newport Beach, California, USA.
- [7] Franco Mazzenga, Francesco Vatalaro and Charles E. Wheatley," Performance Evaluation of a Network Synchronization Technique for CDMA Cellular Communications" IEEE Transactions on Wireless Communications, Vol. 1, No. 2, April 2002.
- [8] Ari Trachtenberg, David Starobinski, and Sachin Agarwal," Fast PDA Synchronization Using Characteristic Polynomial Interpolation" INFOCOM 2002.
- [9] B. Stasicki, K. Ehrenfried, L. Dieterle; K. Ludwikowski, M. Raffel," Advanced synchronization techniques for complex flow field investigations by means of PIV" 4th International Symposium on Particle Image Velocimetry Göttingen, Germany, September 17-19, 2001.
- [10] Thomas Keller, Lorenzo Piazza, Paolo Mandarini, and Lajos Hanzo "Orthogonal Frequency Division Multiplex Synchronization Techniques for Frequency-Selective Fading Channels" IEEE Journal on Selected Areas in Communications, Vol. 19, No. 6, June 2001.
- [11] Matthias Wiesmann, Fernando Pedone, Andre Schiper, Bettina Kemme and Gustavo Alonso,"Database Replication Techniques: a Three Parameter Classification "Proceedings of 19th IEEE Symposium on Reliable Distributed Systems (SRDS2000), pages 206-215, Nürnberg, Germany, October 2000. IEEE Computer Society.
- [12] Naveen Malhotra, Anjali Chaudhary(2014), "Implementation of Database Synchronization Technique between Client and Server", International Journal Of Engineering And Computer Science, Volume 03 Issue 07, Page No. 7070-7073.