An Efficient Method to Synchronize Data in Identified Client and Server

Shahnawaz Khurshid
Computer Science Engineering
Desh Bhagat University, Mandi Gobindgarh, Punjab, India

Jyoti Arora
Department of Computer Science Engineering
Desh Bhagat University, Mandi Gobindgarh, Punjab, India

Abstract- Data synchronization refers to the idea of keeping multiple copies of a dataset in coherence with one another, or to maintain data integrity. In this paper, system developers have to take into consideration the use of synchronization methods in order to efficiently implement their systems. The application developers have worked on serial of data sink which is in order between offline and online application. Earlier when two different images from two different clients uploaded on a server then first image replaced with another. In this researcher solved the problem of data synchronization with image uploading.

Keywords – Database, Synchronization, client, Server, FTP.

I. INTRODUCTION

1.1 Database

A database [2] 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.

1.2 Database Management 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.

1.3 The Architecture of Database Systems

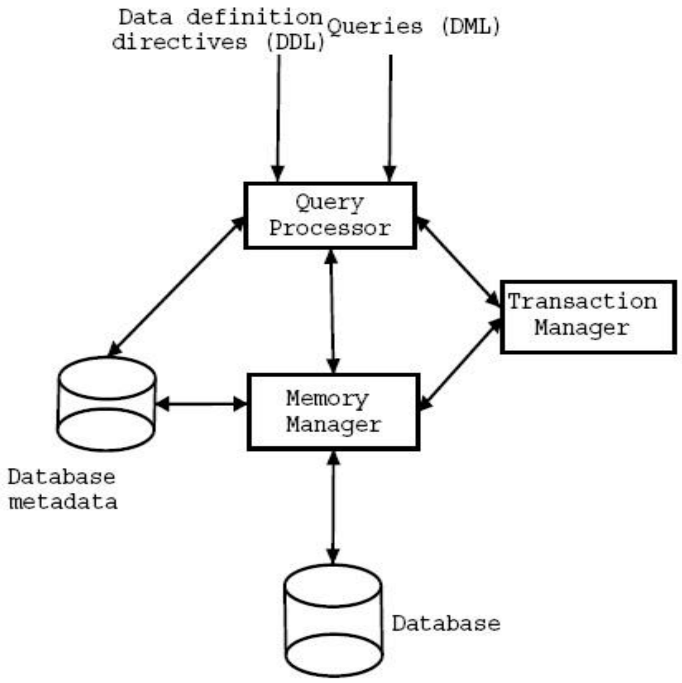

Figure 1. Functional Architecture of DBMSs

The architecture of a DBMS can be examined from several angles: the functional architecture that identifies the main components of a DBMS, the application architecture that focuses on application uses of DBMSs, and the logical architecture that describes various levels of data abstractions.

Functionally, a DBMS contains several main components shown in Figure 1:

1. The memory manager;
2. The query processor;
3. The transaction manager.

The query processor converts a user query into instructions the DBMS can process efficiently, taking into account the current structure of the database (also referred as metadata--which means data about data).

The memory manager obtains data from the database that satisfies queries compiled by the query processor and manages the structures that contain data, according to the DDL directives.

Finally, the transaction manager ensures that the execution of possibly many transactions on the DBMS satisfies the ACID properties mentioned above and, also, provides facilities for recovery from system and media failures.

The standard application architecture of DBMSs is based on a client/server model. The client, which can be a user or an application, generates a query that is conveyed to the server. The server processes the query and returns an answer to the client. This architecture is known as two-tier architecture. In general, the number of clients may vary over time.

In large organizations, it is often necessary to create more layers of processing, with, say, a layer of software to concentrate the data activities of a branch office and organize the communication between the branch and the main data repository. This leads to what is called a multi-tier architecture. In this setting data are scattered among various data sources that could be DBMSs, file systems, etc. These constitute the lowest tier of the architecture, that is, the tier that is closest to the data. The highest tier consists of users that act through user interfaces and applications to obtain answers to queries. The intermediate tiers constitute the middleware, and their role is, in general, to serve as mediators between the highest and the lowest tiers. Middleware may be consist of web servers, data warehouses, and may be considerably complex. Multi-tier architecture is virtually a requirement for World Wide Web applications.

The logical architecture, also known as the ANSI/SPARC architecture, was elaborated at the beginning of the 1970s. It distinguishes three layers of data abstraction:

1. The physical layer contains specific and detailed information that describes how data are stored: addresses of various data components, lengths in bytes, etc. DBMSs aim to achieve data independence, which means that the database organization at the physical level should be in different to application programs.
2. The logical layer describes data in a manner that is similar to, say, definitions of structures in C. This layer has a conceptual character; it shields the user from the tedium of details contained by the physical layer, but is essential in formulating queries for the DMBS.
3. The user layer contains each user's perspective of the content of the database.

1.4 Synchronization

In computer science [3], synchronization refers to one of two distinct but related concepts: synchronization of processes, and synchronization of data. Process synchronization refers to the idea that multiple processes are to join up or handshake at a certain point, in order to reach an agreement or commit to a certain sequence of action. Data synchronization refers to the idea of keeping multiple copies of a dataset in coherence with one another, or to maintain data integrity. Process synchronization primitives are commonly used to implement data synchronization.

1.4.1 Thread or process synchronization

Thread synchronization [4] or serialization, strictly defined, is the application of particular mechanisms to ensure that two concurrently-executing threads or processes do not execute specific portions of a program at the same time. If one thread has begun to execute a serialized portion of the program, any other thread trying to execute this portion must wait until the first thread finishes. Synchronization is used to control access to state both in small-scale multiprocessing systems, in multithreaded environments, multiprocessor computers and in distributed computers consisting of thousands of units, in banking and database systems, in web servers, and so on.

1.4.2 Data synchronization

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

II. LITERATURE REVIEW

Sudha S et al. [5] had explained that cloud computing usually consists of front-end user devices and back-end cloud servers. This gives users to access a large volume of storage on cloud. In this paper, the user can upload file from mobile or PC to the cloud storage. These files will be automatically synchronized to the user's devices when they are connected to internet. So, user files can be viewed from anywhere by any device. In the existing system, we need to download files manually. This paradigm provides the user to synchronize data automatically between devices. They had implemented this paradigm for windows platform.

Isak Shabani et al. [6] had presented an algorithm for data synchronization based on Web Services (WS), which allows software applications to work well on both configurations "Online" and “Offline”, in the absence of the network. For this purpose is in use Electronic Student Management System (ESMS) at University of Prishtina (UP) with the appropriate module. Since the use of ESMS, because of a uncertain supply of electricity, disconnecting the network and for other reasons which are not under the control of professional staff that manages the performance of this system, has interruption to the online work. In order to continue work in such conditions, are founded adequate solutions to work in offline mode and later data synchronization in normal conditions.

Zeljko Medenica et al. [7] had explained that analyzing the effects of driver distraction and inattention on cognitive load has become a very important issue given the substantial increase in the number of electronic devices which are finding their way into vehicles. Typically separate equipment is used for collecting different variables sensitive to cognitive load changes. In order to be able to draw reliable conclusions it is important to possess dependable ways of synchronizing data collections between different equipment. This paper offers one low-cost solution which enables synchronizing three types of devices often used in driving research: driving simulator, eye tracker and physiological monitor.

Arun Kumar Yadad et al. [8] presented a Distributed Transaction Processing Model and an approach for concurrency control in distributed database systems. The analysis of their approach is a decomposition of the concurrency control problem into two major sub-problems: read-write and write-write synchronization. They describe a series of synchronization techniques for solving each sub-problem and will show how to combine these techniques into algorithms for solving the entire concurrency control problem. Such algorithms are called "concurrency control methods". Their approach concentrates on the structure and correctness of concurrency control methods and also the performance of such methods up to some extent.

Susan G. Zucker et al. [1] 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.

III PROBLEM STATEMENT

Non-regular power supply, connection drops, server drops, presents a big problem in software applications. Such problems cause activity interruption at work and inability to do the service on time, which reflects the service to the students. Considering those aspects and for the continuity of this project, a solution to minimize the problems should be found. For this purpose an offline mode is used, realized with the design of the algorithm used for data synchronization based on Web Services. Through FTP we can also transfer the images file to the server in synchronized mode.

IV RESULTS

For the implementation of algorithm we have used C#.net as Frontend and SQL server & MS-Access Database as backend.

1. In the login form, the client enter the required username and password. By clicking on the login button, the enquiry form is displayed.
2. In the enquiry form, the client enter the required information and upload the image. This image will be saved in the database.
3. If the network connection is not available, details of the client along with image remains stored on the client database. Whenever network connection becomes available, the details of the client along with image will transfer to the server.
4. When we make changes on any client system in online mode, the results of data changes on server synchronized only of the concerned client machine. We have tested this algorithm by implementing the customer's information application. For example, when we update the information like contact no. of customer of Chandigarh from the client machine of Delhi. The data updated on the server when the client machine of Delhi is in

the network. The same up-gradation is synchronized only in the customer's database available on the Chandigarh client machine.

V.CONCLUSION

In this research paper, we have investigated the WSs oriented approach for data synchronization of shared data systems, enabling software applications to work in offline mode and thus increasing the confidence in work. We used synchronization to avoid problems that have with the interruption of electricity and the network failures. By the use of synchronization, the officials in the institutions have no more problems with the network failures, and so the system functions without any interruption. The synchronization created has shown positive results in the reliability of the users of the software applications.

This synchronized algorithm is working perfectly on all the columns of database but in future, researcher will also work on images synchronization in all the developed applications. Researcher worked on the synchronization between client and server as the images uploaded from client to server. But through this research, we find that the synchronization is also important from server to client.

VI. REFERENCE

[1] 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.

[2] Ramez Elmastri and Shamkant B. Navathe(2009), "Fundamentals of Database Systems" 5th edition.

[3] A. Bruce Carlson(2012), "E-Study Guide for: Communication Systems".

[4] Cornelia Nussbaum(2012), "Data Synchronization Techniques".

[5] Sudha S and Brindha K(2012), "Data Synchronization Using Cloud Storage" International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 2, Issue 11.

[6] Isak Shabani, Betim Cico and Agni Dika (2012),"Solving Problems in Software Applications through Data Synchronization in Case of Absence of the Network", IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 1, No 3.

[7] Zeljko Medenica, Andrew L. Kun(2012), "Data Synchronization for Cognitive Load Estimation in Driving Simulator-based Experiments", AutomotiveUI'12, Portsmouth, NH, USA.

[8] Arun Kumar Yadav and Dr. Ajay Aggarwal(2010), "A Distributed Architecture for Transactions" International Journal on Computer Science and Engineering (IJCSE), Vol. 2, No. 6, , pp. 1984-1991.