International Conference on Machine Learning and Computational Intelligence-2017
International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2017 IJSRCSEIT | Volume 2 | Issue 7 | ISSN : 2456-3307

Synchronous and Asynchronous Replication

Katembo Kituta Ezechiel[1], Dr. Ruchi Agarwal[2], Dr. Baijnath Kaushik[3]
[1]CSE-SET, Sharda University, Greater Noida,Uttar Pradesh, India
[2]JIMS Engineering Management Technical Campus, Greater Noida, Uttar Pradesh, India
[3]Shri Mata Vaishno Devi University, Jammu and Kashmir, India

ABSTRACT

The objective of this study is to analyse the various technical aspects of synchronization between databases while replication is occurred. To achieve this feat, we embarked in a systematic review of the literature around this theme, an examination that revealed that synchronization is not done with more than one Peer at the time. Therefore, we have been motivated to develop an algorithm which supports the synchronization with more than one Peer.

Background and Objective: This paper presents a systematic review of the two replication approaches, namely: synchronous replication and asynchronous replication. The purpose of this study is to analyse the different technical aspects of synchronization between databases during replication. Methodology: To achieve this, we used the documentary method that allowed us to collect the necessary documents containing the literature that fits with this research and that allowed us to conduct this review. Apart from this one, algorithmic helped us to develop a model that supports synchronization with more than one peer. Result: Our long-awaited result being an algorithm, we have similarly taken care to present the fruits of the efforts of our predecessors who would have proceeded in the same way as us. Conclusion: Finally, we found that it is necessary to have a synchronization algorithm independently of the DBMSs that can be used by the designers of distributed databases.

Keywords : Algorithm, Synchronization, Replication, Peer, Database.

I. INTRODUCTION

In computer science, a distributed database is a database in which storage devices are not all attached to a common Processing Unit Such (CPU). It may be stored in multiple computers, located in the same physical location; or may be dispersed over a network of interconnected computers [1].

The design of a distributed database requires that it be entirely resident on various sites in a computer network or its portion. In this logic, there must be at least two sites hosting the database and not necessarily each site in the network. The strategies can be broadly divided into replication and fragmentation [2]. However, in most cases, a combination of the two is used. But, as far as this work is concerned, our interest is more fixed on replication of data.

Replication is a set of technologies for copying and distributing data and database objects from one database to another and then synchronizing between databases to maintain consistency [3]. It stores separate copies of the database at two or more sites. It is a popular fault tolerance technique of distributed databases [4].

From this point of view, we retain two conceptions which we will first of all clear up: synchronization and consistency.

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 [5].

In turn, the consistency of Replication models is essential to abstract away execution particulars, and to classify the functionality of a given system. Also a consistency model is a method for come to a joint considerate of each other's rights and responsibilities [6].

II. TYPES OF REPLICATION

There are two types of replication which are as follows [7]:

- Synchronous Replication: All copies of a modified relation (fragment) must be updated before commit. Here, the most up to date value of an item is guaranteed to the end user. There are two different methods of synchronous replication:
  1) Read-Any, Write-All: This method is beneficial in case well when reads are much more frequent than writes.
     1. Read-Any: when reading an item, access any of the replicas.
     2. Write-All: when writing an item, must update all of the replicas.
  2) Voting:
     1. When writing, update some fraction of the replicas.
     2. When reading, read enough copies to ensure you get at least one copy of the most recent value.

        Use a version number to determine which value is most recent the copies "vote" on the value of the item
- Asynchronous Replication: Asynchronous replication allows different copies of the same object to have different values for short periods of time. Data is updated after a predefined interval of time.
  1) Primary Site: In primary-site replication, one copy of data is assigned as the master copy. Updation of data is possible only with in the master copy. The secondary copies of data can only be read. Changes to the master are periodically propagated to the secondary copies.
  2) Peer-to-Peer: In peer-to-peer replication, more than one replica is updatable. In addition, a conflict resolution strategy must be used to deal with conflicting changes made at different sites.

III. LITERATURE REVIEW

We are not the first to direct our thoughts on the problem of data replication and its techniques. It is therefore essential for us to review the literature that fits in with this theme in order to justify our research.

Chaturvedi & Prof. Jain [8] focuses on replication in distributed file systems. They present replication as a strategic key in distributed systems for enhancing accuracy, availability, and performance. They gave also a brief introduction to replication and various algorithms have been discussed and a detailed study has been performed. Although they concluded that optimistic replication systems have been in use for some time. Client - Server model is generally used

but when dealing with mobile, peer-to-peer model is used; this allows direct communication and synchronization between all the peers but has scalability problems.

Gudakesa et al. tackled the data synchronization method usable in the two-ways of data synchronization (Master-Slave / Slave-Master) [9]. They indicate that synchronization of data is an integral part of replication in that it ensures the reliability and similarity between copies of the database and thus the objects and data. They added that to synchronize the data several methods are applicable; so they presented the one that uses the audit log that records all the activities that occur to the database. In this context, they have shown that this technique can be applied to almost all database management systems (DBMS). In the end, they also targeted possible problems that can occur when synchronization is running and how to solve this problem.

Kaushik B. et al [23-25] developed a neural network based model to evaluate the performance of reliability evaluation in complex and high performance network model. Agrawal et al. proposed taxonomy for partitioned replicated database systems that reside in the cloud [10]. Taxonomy is a practice derived from the need for scientific classification of species. Indeed, the authors of this article have focused on databases that support both transactions and replication. They used several advanced systems to illustrate specific instances of taxonomy. Thus, they quickly realized that taxonomy is distorted, in the sense that there are several subcategories of replicated transaction systems, but only one category of replicated object systems. Finally, they concluded that taxonomy represents a first attempt at principle to understand and classify the transaction mechanisms of many proposed systems of the state of the art.

Krishna et al. advocated producing a cloud Operating System in which the user can upload file from mobile or Personal Computer to the cloud storage [11]. They have implemented a Windows platform on which a user's files can be automatically synchronized with his devices. Thus the user can view his files anywhere; in analogy with the existing systems, they were able to demonstrate that here the files were to be downloaded manually. On finish, automatic data synchronization between devices of the user was studied.

Ranjan & Agarwal [22] analysed the customer transactional database and try to understand the true value of the customer.

Agarwal.et al [19] reviewed parallel apriori algorithm on mad reduce framework for performance enhancement by dividing the transactional database into data chunks and distributes them among different machines. Ranjan and Agarwal [21] also analysed the database of a retail firm and found the behaviour of purchased products using classification and Regression Modelling.

Kelemu & Prof. Patil [12] defined a strategy which combines together horizontal fragmentation and vertical fragmentation of the relational database table in distributed system. This strategy is hybrid fragmentation. The subspace grouping algorithm on which hybrid based fragmentation is an incomparable advance in the application of partitioning data with the tuples (rows) and attributes (columns) of a database relation (table). They referred to project clustering as an advance that uses the sub-space grouping algorithm. Thus, it has been developed a new algorithm and it has been implement and also analysis with the random algorithm. This approach has improved the distribution of relations of databases containing large tuples and attributes.

Agarwal & Ranjan [20] developed a model to measure and manage the performance of a retail sector using multiple databases. Salunke & Potdar presented some static and dynamic database partitioning techniques to support mission critical databases [13].

Following techniques have been discussed:

- Partitioning based on selectivity;
- A greedy algorithm to select the "best" dimension tables of a star schema;
- Reloading in a main memory database system supported by database partitioning techniques;
- The distribution of a database based on a grouping approach and a genetic algorithm;
- A dynamic vertical partitioning approach for the distributed database system.

Hiremath & Dr. Kishor [14] studied the various problems areas and advantages and disadvantages of Distributed Database . They showed that Distributed database would allow end user to create and store data anywhere in the network where database is situated. In this approach, while storing and accessing data from distributed database through computer network, there are various problems that occurs such deadlock, concurrency and data allocation using fragmentation and replication. To manage these problems, it is necessary to design the distributed database carefully manner. Apart that they have presented some distributed database architecture strategies:

- Top-Down Approach: mostly used when we have to implement distributed database from beginning;
- Bottom-up Approach: This type of Approach is used when distributed database already exists and we have to add another database in existing environment.

Akshay &Yogesh [15] reviewed the security issues and concurrency control in distributed database system and data security aspects of client/server architecture. Here it was pointed out that the most important problem is the security that could occur and potentially compromises access control and system integrity. For example, a solution has been proposed for certain security aspects such as multilevel access control, privacy, reliability, integrity, and recovery for a distributed database system. In addition, some competitive control algorithms were examined, for example: the basic timestamp algorithm, the distributed two-phase lock protocol (2PL), the distributed optimized protocol and the wound-wait algorithm.

Imam et al. proposed systematic literature review on Data synchronization between mobile devices and server-side databases [16]. The objective of this study was to examine the state of the art in various aspects of mobile devices with regard to data sharing and synchronization between mobile device databases and server-side databases, opposing to server-server synchronization. Relevant literature was chosen for examinations and several aspects were identified as being directly or indirectly involved in synchronization. The issues that have been discussed are the synchronization of data between mobile device and server databases, cloud-based solutions, data inconsistency, conflict resolution strategies, data processing, dependency Suppliers, summary of messages and algorithms used and common tools adopted.

Shabani et al. have designed 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 [17]. In this study, it has been shown that the use of synchronization is a major challenge for the institutions in general because the agents or personnel (users) are less aware of the network failures and the system works without any involuntary interruption. In particular, the

confidence of the administrative and academic staff of the University of Prishtina is increasing as there is no waiting for documents because the network fails. The results of this synchronization set up are judged to be positive because there is a reliability which is observed in the user software applications.

Fadoua & Amel established a standard synchronization process between different sites of a distributed database architecture including database heterogeneity, variable synchronization delays, network capability restrictions and fault management ability [18]. To go straight to the goal, apart from the introduction and the conclusion, it has been:

- Described the different approaches and their limitations;
- Presented the suggested synchronization protocol;
- Detailed the data serialization and the deserialization mechanism regarding network bandwidth consumption and packet preparation time impact;
- Studied the protocol performance on the most optimistic and pessimistic scenarios;
- Reported the experimental tests of the implemented example.

IV. METHODOLOGY

After the course of the literature above offered by our predecessors in this field, we realized however that all of these possibilities for synchronization are being making over a centralized Peer-to-Peer Architecture because they require a central server. Thus, we notice that existing synchronizers are limited because either they require a synchronization of not more than two copies at a time.

It would be interesting to set up another collaborative environment to remove such limitations in such a way that synchronization is now done with several copies. We approach in this direction because visibly centralized Peer-to-Peer architecture also presents a major problem that is such that it only offers a single gateway, its central server, which is the Achilles heel of everything over the network. It would be enough that this server knew a breakdown to block or to disconnect all the users and to stop the operation of the whole system because no peer will have the data no longer updated.

Here we present the methodology of the work we used to achieve our goals. So, for our case, our gaze remains focused on:

- The documentary method, a set of steps to search, identify and find documents related to a subject by developing a research strategy. It helped us to collect and analyse the documents that we used to identify the shortcomings of existing synchronizers.
- The Algorithmic as method or technique helped us to design the algorithm of instructions and steps of a Pure Peer-to-Peer Synchronizer.

V. RESULT

This section presents the models collected from the documents of our predecessors in relation to the objectives that this work has set itself. Then it analyses the defined methodology and then presents the discussion of the results obtained.

Review of the literature around this theme revealed that synchronization is not done with more than one peer at a time. Some synchronization models, results of some works, are presented in Figure 1. and Figure 2.

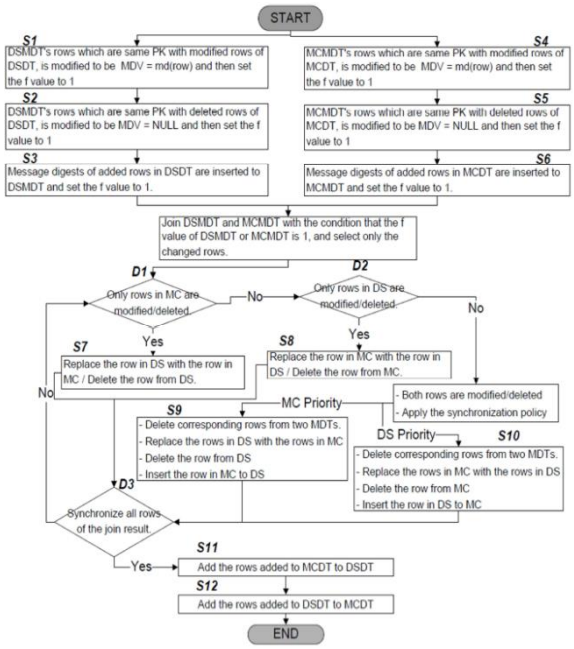

Figure 1. Synchronization Algorithm [16]

DS: Database Server	D1 to D3: Decision 1 to Decision 3
DSMDT: Database Server Mobile Digest Table	DSDT: Database Server Data Table
MDV: Message Digest Value	MC: Mobile Client
PK: Primary Key	MCDT: Mobile Client Data Table
S1 to S12: Step 1 to Step12	f: Flag

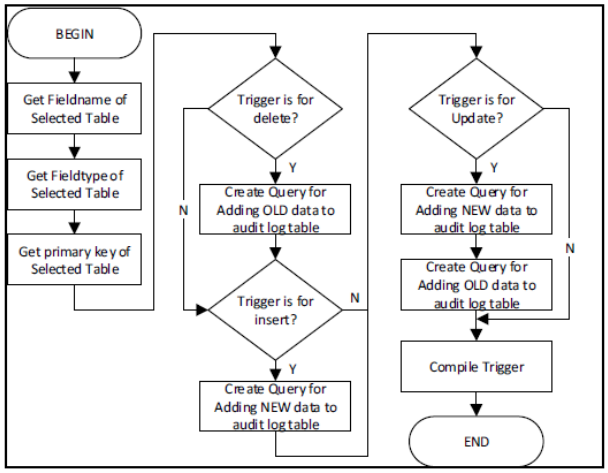

Figure 2. How Audit Log Trigger Created [9]

This is used when it's mandatory to keep the same copy of the data in two data storage node on the network. It is Two-ways synchronization used Audit log mechanisms [9].

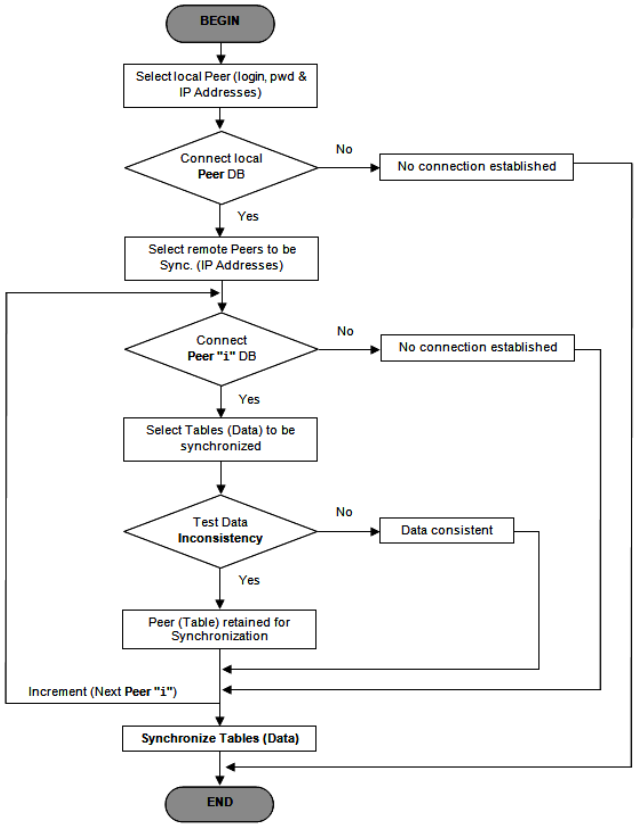

Figure 3. Algorithm for a Pure Peer-to-Peer synchronization

Therefore, it has been necessary to design a synchronization algorithm that can update several copies of the databases at the same time. Each machine in its roles is identical to another, and then we will call this type of system Pure Peer-to-Peer Synchronizer.

As shown in Figure 3 the steps of the algorithm for a Pure Peer-to-Peer synchronization are given below:

Step1. Select the local Peer and connect on it by providing the login, password and the IP address (facultative): if these provided parameters are incorrect then no connection established else next step;

Step2. Select remote Peers, by indicating theirs IP addresses, to be Sync and test connection with them one to one: if Peer non-jointed then no connection established, next Peer else next step;

Step3. Select Tables (Data) to be synchronised and test Data inconsistency: if Data consistent then next Peer else Peer (Table) retained for Synchronization, next step;

Step4. Synchronize Tables (Data) of all retained Peers at the same time.

VI. CONCLUSION

In this paper we have managed to conduct literature study on synchronization while replication is occurred. Although in relevant literature, we have realized that synchronization is not done with more than one Peer at the time; this has been our motivation to develop an algorithm which support the synchronization with more than one Peer.

This algorithm may be used by Distributed Databases and applications designers since they will need interaction between different DBMSs.

As future work we will write a complete algorithm in which it will be presented step by step scenarios to Synchronize Tables (Data) i.e. the internal reality of a Pure Peer-to-Peer synchronization algorithm.

VII. REFERENCES

[1]. Kaur, K., & Singh, H. (2016). Distributed database system on web server: A Review. International Journal of Computer Techniques, 3(6), 12-16.

[2]. Ozsu, M. T., & Valduriez, P. (2011). Principles of Distributed Database Systems (3rd ed.). New York, USA: © Springer Science+Business Media, LLC.

[3]. Microsoft. (2017). SQL Server Replication. Retrieved May 2017, from Microsoft Documentation: https://docs.microsoft.com/en-us/sql/relational-databases/replication/sql-server-replication

[4]. Truica, C., & Boicea, A. (2013). "Asynchronous Replication in Microsoft SQL Server, PostgreSQL and MySQL". International conference on cyber science and engineering.

[5]. Malhotra N., Chaudhary A., (2014), Implementation of Database Synchronization Technique between Client and Server, International Journal of Engineering and Computer Science, 3(7): 7070-7073.

[6]. Souri, A., Pashazadeh S. & Navin, A., H. (2014). "Consistency of data replication protocols in database systems: A review", International Journal on Information Theory (IJIT),3(4), 19-32.

[7]. Tomar, P., & Megha. (2014). "An Overview of Distributed Databases". International Journal of Information and Computation Technology, 4(2), 207-214.

[8]. Chaturvedi, N., & Prof. Jain, D., C., (2012), Analysis of Replication and Replication Algorithms in Distributed System, International Journal of Advanced Research in Computer Science and Software Engineerin, 2(5), 2277 128X.

[9]. Gudakesa, R., Sukarsa, I., & Sasmita, I., M., A., (2014). Two-ways database synchronization in homogeneous dbms using audit log approach, Journal of Theoretical and Applied Information Technology, 65(3), 1992-8645.

[10]. Agrawal, D., El Abbadi, A., & Salem,K., (2015). A Taxonomy of Partitioned Replicated Cloud-based Database Systems, Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, 38(1), 4-9.

[11]. Krishna, S., V., Gokul, K.& Kumar, S., (2012). Data Synchronization Using Cloud Storage, International Journal of Advanced Research in Computer Science and Software Engineering, 2(6), 2277 128X.

[12]. Kelemu, Y., & Prof. Patil, S., (2016). Hybridized Fragmentation of Distributed Databases using Clustering, International Journal of Engineering Trends and Technology (IJETT), 37(1), 2231-5381.

[13]. Salunke, D., T., & Potdar, G., P., (2014). A Survey Paper on Database Partitioning, International Journal of Advanced Research in Computer Science & Technology (IJARCST), 2(3), 2347 – 8446.

[14]. Hiremath, D., S., & Dr. Kishor, S., B., (2016). Distributed Database Problem areas and Approaches, Journal of Computer Engineering : National Conference on Recent Trends in Computer Science and Information Technology, 2278-8727.

[15]. Akshay, M., G., &Yogesh, R., G., (2016). Concurrency Control and Security Issue in Distributed Database System, International Journal of Engineering Development and Research, 4(2), 2321-9939.

[16]. Imam, A., A., Basri, S., Ahmad, R., (2015). Data synchronization between mobile devices and server-side databases: a systematic literature review, Journal of Theoretical and Applied Information Technology, 81(2), 1992-8645.

[17]. Shabani, I., Çiço B.,& Dika, A., (2012), Solving Problems in Software Applications through Data Synchronization in Case of Absence of the Network, International Journal of Computer Science Issues, 9(1): 1694-0814.

[18]. Fadoua, H., & Amel, G., T., (2015), Near Real-time Synchronization Approach for Heterogeneous Distributed Databases, The Seventh International Conference on Advances in Databases, Knowledge, and Data Applications, 978-1-61208-408-4.

[19]. Ruchi Agarwal, Sunny Singh, Satvik Vats, 'Review of Parallel Apriori Algorithm on Map Reduce Framework for Performance Enhancement', organized by Computer society of India(CSI) Delhi and NCR chapter at Bharti vidyapeeth educational complex, New Delhi during 2nd -5thDecember, 2015. Published in Springer Nature Singapore, Advances in Intelligent and Soft Computing (AISC) series, Scopus Indexed pp 403-411.

[20]. Ruchi Agarwal, Jayanthi Ranjan, 'An empirical investigation on association rule mining in Indian Retail Industry', International Journal of Intercultural Information Management, Inderscience Publishers, Vol. 3, No. 3, pp.226–241, November 2013.

[21]. Jayanthi Ranjan, Ruchi Agarwal, 'Advantages of Decision Trees Using Data Mining In Indian Retail Industry', Journal of Knowledge Management Practice, Vol. 11, Special Issue 1, January 2010.

[22]. Jayanthi Ranjan, Ruchi Agarwal, 'Application of segmentation in customer relationship management: a Data Mining perspective', International Journal of Electronic Customer Relationship Management, Inderscience Publishers, Scopus Indexed, Vol. 3, No 4, pp.402-414, 2009.

[23]. Kaushik B. et al., (2015), 'Performance evaluation of approximated artificial neural network (AANN) algorithm for reliability improvement', Journal of Applied Soft Computing, Elsevier, Vol. 26, pp.303-314.

[24]. Kaushik B. et al., (2013), 'Achieving Maximum Reliability in Fault Tolerant Network Design for Variable Networks' Journal of Applied Soft Computing, Elsevier, pp. 3211-3223.

[25]. Kaushik B. et al., (2012), 'Improved Approach for Maximizing Reliability in Fault Tolerant Networks', Journal of Advanced Computational Intelligence and Intelligent Informatics, Fuji Press, Japan, Vol. 17, 1, pp. 27-41.