2019 IEEE Conference on Information and Communication Technology (CICT)

Efficient Method for Data Synchronization in Mobile Database

Nidhi Singh
Dept. of Computer Science
Madan Mohan Malaviya, Gorakhpur, India
e-mail: sinidhi302@gmail.com

Mr. Muzammil Hasan
Dept. of Computer Science
Madan Mohan Malaviya, Gorakhpur, India
e-mail: muzammil_hasan@yahoo.com

Abstract— **In the present era, the use of mobile gadgets is increasing, the process of such mobile gadgets has a limited processing, bandwidth, memory and transfer speed, for which proper resolution is necessary. In a versatile Processing situation, there are many issues like synchronization in the database, the security of the information and portable exchanges. One of the major challenges in the Mobile database is Synchronization.**

Multiple analyzes are conducted over time to minimize these problems in order to maintain better accuracy. While synchronization requires the accuracy of data to expect a high volume of traffic and high time complexity and to develop appropriate algorithms to synchronize such problems.

This paper we consist the current situation of data synchronization and give a new and better concepts for understanding the synchronization. It also helps to control problems of data deployment and maintains the possibilities of accurate data using a new solution for data synchronization.

***Index Terms*— Mobile Database, Synchronization, Mobile-side Database, Server-side Database, Mobile Computing, Message Digest.**

I. INTRODUCTION

In the field of mobile devices, database synchronization is a process that establishes consistency between the data source and master.

Over time, mobile technologies advanced through their equipped devices with a lightweight database. Due to these assets mobile devices provide a great role in the exchange of information for the processing of business, organizations and many other fields through their own applications which are based on mobility, as their features supported them.

Occasionally, mobile devices have less processing and insufficient availability of memory is reliable at a better source of battery, although mobile devices have limited transmission capacity, yet they always use the network. Typically, this network is sometimes disconnected from the source of the network, this may be due to several reasons such as network failure, failure of device connectivity, loss of battery and the possibility of losing the network of the device due to several events.

It has been demonstrated by this paper that the synchronization algorithm for mobile databases and server databases relies on message digest, which we believe to be true of research and creates new challenges, enabling future researchers to conduct research. Thereafter, they create something new and can get solutions in the field of data synchronization of mobile devices. Synchronization algorithm occur data interchange through a procedural way in one end to another end.

II. Background Knowledge

A. <u>**Literature Review:-**</u>

The purpose of this process is to construct and compare the basis of the entire existing work. In this section, systematic literature performance evaluation is executable. Kitchenam and Charters [13] state that first, we will dispute the search strategy to recover the research paper. After that, we describe the subtypes as observers of this research paper. Finally, as presented in the review strategy, we will fill gaps in this research as to the insertion and extraction processes that will make it a new one according to my view.

We have used a source from published material IEEE Xplore from 1998 to 2019. In computer science and engineering, short critical journals have already been added. IEEE's prestigious publishers are showcasing international journals and research journals in terms of innovation in engineering and management, computer science and mobile computing. Sources among management, engineering and technology, signal processing and integrated networks, artificial intelligence, and industrial engineering sections are distinguished in their fields.

B. <u>**Synchronization:-**</u>

The archetypal (ideal) model of synchronization 7 in Database Management System (DBMS) works with the number of clients that are connected to the server through the system, as standard clients in servers and compact gadgets. Through the DBMS server, the servers are connected to the database shared by the client. A client can build a user database from a client device. Despite this fact, it is usually created by downloading information from a remote server. Since the data is kept in the database of both the client device and the server, thus there is some replication of the information. As usual, after the exchange of information, the client is displaced from the server. Once the association (connection) is cut, the client's database can be changed by the DBMS implanted on the client device; despite the fact that when the client is associated with it, this change must be connected to the server and this information is completed for consistency. For a similar reason, changes

978-1-7281-5398-8/19/$31.00 © 2019 IEEE

between ends of associations that occur in a database server must be associated with the client gadget in the same way when both are delivering messages, so the gadget (device) may receive a response from the server. The synchronizing administrator (i.e. AnySync) is an essential module of the DBMS. The reason is that it is preserving data replication compatibility.

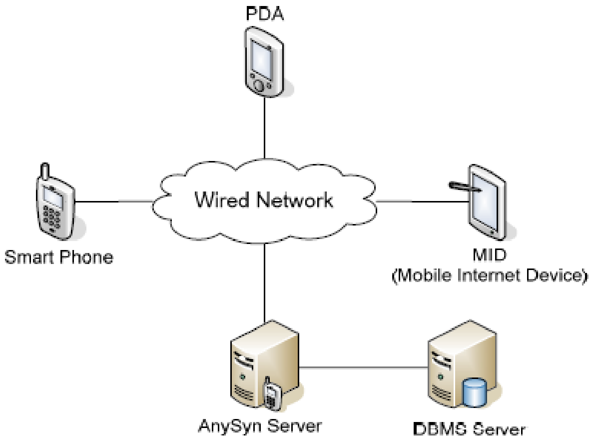

Fig 1: SAMD Synchronization Framework [4]

C. ***Contention:-***

If multiple clients are to transfer (exchange) information from the server, at that point these can be changed in a various manner, thus the data become differ. In this scenario, in the databases of every client cannot be applied to the changes appropriately into the database server. And this dubious situation is known as the contention [8]. Conventionally, AnySyn recognizes and settle potential contentions [7]. Between various tasks, the primary kind of conflict happens which must be performed for similar data records and it might happen amid client devices synchronization. Contention there are 3 types of operations in which contention can be divided: insertion, elimination, and conflicts in the effect of data updating. Demonstrate every activity that can be performed on the online database server to a lone record that makes inherent clashes in Table I [1].

Exactly when the synchronization starts within a customer and a server, The synchronizing supervisor (AnySyn) perceives that what kind of possible clash has occurred and settle every dispute in a predefined way, in light of the fact that each contention can be settled after the criteria of goals characterized by the client of the client device itself or by the overseer administration, selecting the concurrence of information you have to store them into the database of the outlying server.

C	DB Mobile	DB Server	C	DB Mobile	DB Server	C	DB Mobile	DB Server	C	DB Mobile	DB Server
1	UC	UC	5	ADD	UC	9	UC	MOD	13	DEL	DEL
2	UC	ADD	6	DEL	UC	10	ADD	DEL	14	ADD	MOD
3	MOD	UC	7	MOD	ADD	11	MOD	MOD	15	MOD	DEL
4	ADD	ADD	8	DEL	MOD	12	DEL	ADD	16	UC	DEL

Table 1: Feasible Contention

(UC: Unchanged, C: Case, ADD: Added, DEL: Deleted, MOD: Modified)

In Fig. 2 is shown for generating the message digest value (MDV) and TS fig shows the manner by which can be applied this mechanism in multiple columns of a table.

D. ***Message Digest:-***

For the solution of contention synchronization techniques commonly utilized via AnySync in the DBMSs are automated and 2 components are used. That is Message Digest and the Alteration Time. The one-way hash function is the message digest [3] so as to maps, a message with an unpredictable length hash a motivator to a set-up length; in different terms, the hash function denoted with H creates a message which is denoted as h that can be represented as,

Plain Text h = message digest with fixed length M (H (M))

Regardless of whether a solitary piece is replaced in the message which can make every one of the estimations of any change in the message digest.

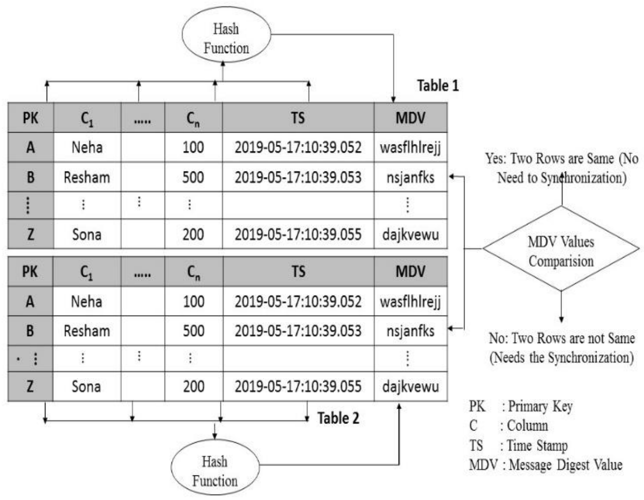

Figure. 2. Time Stamp and Message Digest in Data Tables

III. SAMD Synchronization Algorithm

The synchronization algorithm about SAMD implemented and fig: 3 demonstrates that table constructed between both server and device databases. Each of them possess a data table and a message digest table. A mobile database having MCDT data table, MCMDT message digest table and server database contains data table and message digest table. The message digest table contains value of the message digest and time stamp from the data table. The data table having the business data with time stamp. Time stamp is calculate using the Logical Clock.

The server database message digest table containing following columns: PK column of data table, time stamp column (TS), MDV column, flag column (f) and mobile id column (Mid). The flag column (f) denotes the variability of data which happened in comparable column; thus, to recognize the row which required to synchronization, the f column (flag column) is consumed as an identifier. The mobile id segment is the mobile device ID that have a separate digit of mobile devices, although its utilization only for distinguish to those devices in which synchronization is required. The TS column (Time Stamp column) is the time stamp column which indicates the time of data insertion or modification of data by the client in the database.

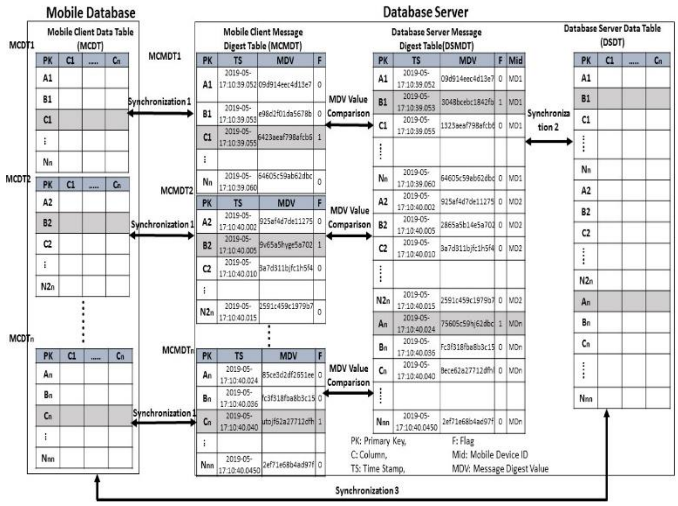

Figure 3: Efficient Method for Synchronization in Mobile Database Data Table.

In above fig, if PK value of a row is A1 which indicate the value of two message digest without any synchronization. Sometime, if the PK value of a row is B1, the MDV value in MCDT and the MDV value of the DSMDT is different and the value of MCMDT flag is 1. Accordingly, a process of synchronization is needed here. For each row, the synchronization procedure is performed to determine the majority of the irregularities referenced in Section 2/C. For example, in the event that there is an irregularity in row B1, synchronization happens to database of server from device database and PK B1 row of DSDT is supplanted with the PK B1 row of MCDT. The synchronization algorithm involves the Synchronizations (i.e. synchronization 1, synchronization 2, and synchronization 3) as displayed in the above figure. The data table and the message digest table is synchronized through synchronizations 1 and 2.

Hence, the two are indistinguishable synchronization algorithms enforced to various tables. Value which are makes by each table row are matched or not with the value of same table of message digest. If the values are equivalent, at that point the process of synchronization is not required and data are also same. Other hand, values are contrasting, it indicates the correction about value of data table. For this situation, resuscitation of message digest table accompanied by new value and the value of flag lay as 1. The flag value indicates that a row needs to synchronization or not. The server-side database contains a DSMDT for each DSDT. In spite of the fact that the MCMDT estimate is smaller than that of the DSMDT, for each mobile device, there is an MCMDT it contains a unique ID. For each row of the DSDT each time it is extremely wasteful to perform Synchronization 2, here synchronization order are needed for mobile device.

Consequently, at whatever point demands of synchronization is needed for device, the value of device ID dispatched above server-side database. After that a specific row from DSMDT which belongs to mid column value of device ID and synchronization 2 are adopted by the SAMD which are implemented to specified rows. For example, a mobile device demands synchronization and its value of mobile device ID are 'md1', the row whose Mid-section value is 'md1' is chosen and afterward just utilize the Synchronization 2.

Succeeding the SAMD algorithm dissect the sort of unpredictability utilizing the flag estimations of one and the other messages digest tables, the primary key, which is utilized to distinguish the row. Since Synchronization 3 is executed among two data tables for respectively inconsistent type. The row flag of synchronization is attached with 0 in the table of messages digest after completion of synchronization.

Portable devices have constrained assets, and amid the synchronization procedure, the heap on the device ought to be limited. Therefore, as appeared in Fig. 3, all the tables of message digest are put in the database which belongs to server to accumulate depot place of the mobile device, while there is the heap brought about by getting to the system in Synchronization 1 however the MCDT data dimension are shorter as compared to server. Moreover, for shifting of synchronization 1 to the server-side database in a solitary conveyance through cabled system utilizing a SQL query fit for batch processing, the MCDT data essential. After this point in which lessens are loaded and brought about by accessing of network in the process of synchronization 1 arrangements, where mobile devices have no load.

The SAMD synchronization algorithm necessity to have the following limitations.

A. All database tables must contain a primary key.

B. Message digest table and data table primary keys contain indistinguishable incentive for a liable row.

C. A new row inserted into the database of both server-side and mobile devices; there are no similarities between both primary key values.

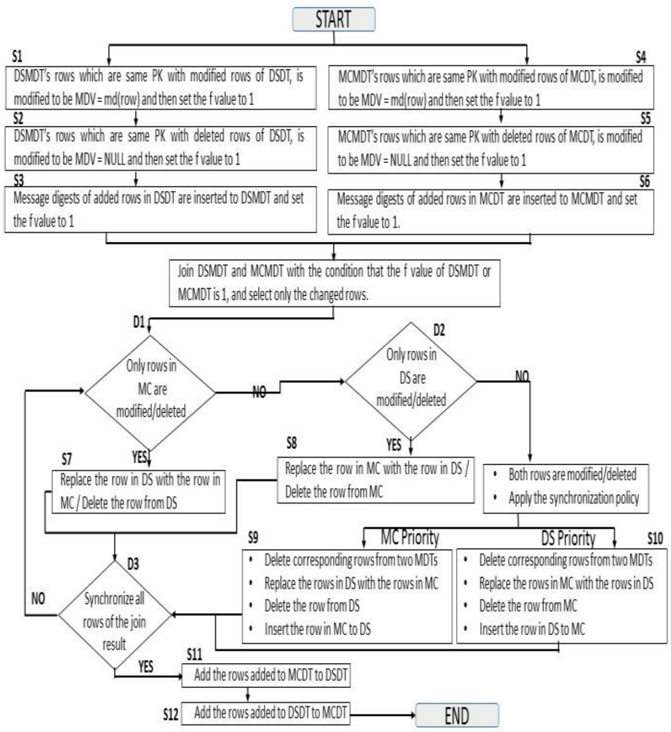

Figure 4: stream outline for the SAMD synchronization algorithm.

Restrictions A and B having in every table of the relational database model are the basic conditions. Amid the synchronization procedure between both device and server databases, Restriction C infers that there is no honesty impact for the primary key. Despite the fact that indistinguishable primary key value exist embedded at the two points, this issue is not thought about since it can without much of a stretch be settled by application-level processing or the synchronization approach.

The Synchronization 2 stage is represented in Fig 3 of steps S1 to S3. DSDT rows (for which Steps S1, S2, and S3 ought to be enforced) occur recognized by drooping rows when the DSDT and DSMDT are Full-Outer-Joined.

To makes alteration of DSDT row, Step S1 do the synchronization process with the DSMDT.

The message digest value is determined for the DSDT row. For estimation of DSMDT of message digest value and MDV column contrasted. On the off chance that the both values are indistinguishable, not required of changed in DSDT row, and the vice versa. On the off chance that a conversion is distinguished, message digest value of DSDT's row and value is set as 1 of flag column override with DSMDTs MDV column value.

In forms of position or erased row from the DSDT, the synchronization process is in step S2 - S3. Fitted of a row, its message digest and primary key estimations of the new column are included the DSMDT. And for erase row, the value of MDV column lay as NULL in the DSMDT. In the event that absence of positioning the column value an incentive to NULL the row is erased, be indistinct from placing a row into the DSDT.

Steps S4 - S6 shows that the Synchronization 1 stage of Fig. 3. In this stage The MCDT and MCMDT synchronization are involved; the essential algorithm is indistinguishable from that utilized in Steps S1~S3. However, different vendor databases have internal data tables, the MCDT, and MDMDT which are actually differ, so Full-Outer-Join is not feasible in Steps S1~S3. For this situation, for the MCDT table to be duplicated to the server side an impermanent table must be formulated, and Synchronization 1 procedure is executed, succeeding, the replicated data is erased. By the batch processing, this individual transaction ensures the autonomy of the SAMD algorithm for the trafficker of database.

The Synchronization 3 phase of Fig. 3, is shown in Steps S7~S12. At the point when the DSMDT and MCMDT are Full-Outer-Joined, the rows that are liable to synchronization and utilizing the dangling rows and the DSMDT and MDCMDT flags, the conflicting sorts are distinguished and at that point, the synchronization between the DSDT and MCDT is accomplished.

Step S7 includes synchronization of alter or erased row from the MCDT with the DSDT. Step S7 scans for a row under the D1 condition, with an MCMDT flag estimation of 1 and a DSMDT flag estimation of 0. The flag values designate that the row was altered or destroyed from the MCDT. A null MDV column of the MDMDT means removal from the MCMDT. Something else, there has been an alteration. On account of deletion, pushes that compare to the rows erased from the MCDT ought to be erased from DSDT, DSMDT, and MDMDT. In the modification case, the DSDT row esteem is supplanted with the MCDT row esteem, and DSMDT row esteem is supplanted with the MDMDT row esteem. The flag estimations of the synchronized row of the DSMDT and MCMDT are set to 0 over the completion of synchronization. This Procedure determine the inconsistency cases C3 and C4.

Synchronization of the altered/erased rows with the MCDT from the DSDT is included in Step8. The previous progression is indistinguishable from the Step S7's algorithm, however synchronization is happened to MCDT from DSDT. To accomplish the irregularity cases case C9 and Case C13 this procedure settles.

At the point when in the two tables DSDT and MCDT change or erasure happens, as indicated by the synchronization strategy steps step S9 and step S10 carry out synchronization toward the path to MCDT from DSDT or in the inverted flank. The rows which necessity the synchronization are those and the flag estimations of 1 for both the DSMDT and MCMDT. In Steps S9 and S10 four cases ought to be considered, as in table 5.1.

Case	Mobile Database	Server-side Database
1	Modification	Deletion
2	Modification	Modification
3	Deletion	Modification
4	Deletion	Deletion

Table 2: Case Analysis

Synchronization is carried out form the MCDT in the direction of the DSDT in step S9. Indistinguishable DSDT and MCDT rows are erased, therefore the relating DSMDT and MCMDT rows ought to be erased too, in the Case 1. After finishing of Case 1, the rows that compare to those erased from the MCDT ought to be erased from the DSDT for Case 3.The irregularity cases C12 and C16 are settled, when the two cases are finished. Case 2 and Case 4, with the altered row of the MCMDT renovate the row in DSMDT. For Case 2, into the DSDT, the MCDT row esteem is embedded in light of the fact that the row in the DSDT is erased.

For the sake Case 4, the row, which is comparing, in the DSDT is supplanted with MCDT row esteem on grounds that the row which is in the DSDT is converted or improved. The irregularity cases C11 and C15 are settled one time the two cases are finished.

For Step S10, synchronization happens towards the MCDT from the DSDT. Which is utilized in Step S9 with an alternate synchronization bearing, the algorithm is indistinguishable from that. After Step S10 is finished the cases with irregularity (i.e., C16, C12, C15, and C11) are settled.

Step S11 includes a reflecting row embedded into the MCDT to the DSDT. In order to which the flag estimation at MCMDT is set to be 1 and this row is an impending row, Step 11 is enforced to that push. The rows which are embedded into the MCDT and MCMDT that likewise embedded into the DSDT and DSMDT. Settles the irregularity case C2 in finishing the procedure. Step S12 indicates the row embedded into the DSDT on the MCDT. The algorithm is indistinguishable from that utilized in Step S11 yet with an alternate synchronization direction. The culmination of this progression settles the irregularity case C5.

SAMD algorithm executes to settle through the synchronization procedure the majority of the irregularities recorded in TABLE I. As a result, it tends to be inferred that the SAMD algorithm synchronizes each conceivable type of irregularity.

IV. Conclusion

This paper is based on message digest and time stamp synchronization that combines both databases as well as the server-side and the mobile side with the SAMD synchronization algorithm. The SAMD algorithm is convenient to use in specific

978-1-7281-5398-8/19/$31.00 © 2019 IEEE

venders and to avoid relational databases operating from both server-side and mobile side database SQL results. In this case, the profession is approved for a better undertaking of eligibility, compliance, and limitation. This quality of the mobile profession is effective because it takes into account the future mobile market segment which has diverse characteristics and it prioritizes RDBMS, mobile databases and various mobile devices.

References

[1] M.-Y. Choi, E.-A. Cho, D.-H. Park, C.-J. Moon, and D.-K. Baik, "A database synchronization algorithm for mobile devices," IEEE Transactions on Consumer Electronics, vol. 56, no. 2, pp. 392–398, 2010.

[2] M. Eric, S. Mitrovic, M. Stefanovic, and A. Djordjevic, "Tribology database as platform for mobile database and use of mobile technologies," Tribology in Industry, vol. 38, pp. 268–276, 06 2016.

[3] J. Singh and E. A. Kaur, "Review for image and database synchronization between client and server," in International Journal of Innovations and Advancement in Computer Science IJIACS, 2016.

[4] M. B. Kekgathetse and K. J. Letsholo, "A survey on database synchronization algorithms for mobile device." Journal of Theoretical and Applied Information Technology, vol. 86, no. 1, 2016.

[5] T. A. Alhaj, M. M. Taha, and F. M. Alim, "Synchronization wireless algorithm based on message digest (swamd) for mobile device database," in 2013 INTERNATIONAL CONFERENCE ON COMPUTING, ELECTRICAL AND ELECTRONIC ENGINEERING (ICCEEE), pp. 259–262, IEEE, 2013.

[6] V. Balakumar and I. Sakthidevi, "An efficient database synchronization algorithm for mobile devices based on secured message digest," in 2012 International Conference on Computing, Electronics and Electrical Technologies (ICCEET), pp. 937–942, IEEE, 2012.

[7] D. Barbar´a, "Mobile computing and databases-a survey," IEEE transactions on Knowledge and Data Engineering, vol. 11, no. 1, pp. 108–117, 1999.

[8] D. Becker-Bessudo, G. Fernandez-Anaya, and J.-J. Flores-Godoy, "Preserving synchronization under characteristic polynomial modifications," IFAC Proceedings Volumes (IFAC-PapersOnline), vol. 2, pp. 187–192, 01 2009.

[9] B. A. R and P. B. VB, "Mobile database review and security aspects," International Journal of Computer Science and Mobile Computing, vol. 3, no. 3, pp. 1174–1182, 2014.

[10] J. Domingos, N. Sim˜oes, P. Pereira, C. Silva, and L. Marcelino, "Database synchronization model for mobile devices," in 2014 9th Iberian Conference on Information Systems and Technologies (CISTI), pp. 1–7, IEEE, 2014.

[11] M. Faiz and U. Shanker, "Data synchronization in distributed client-server applications," in 2016 IEEE International Conference on Engineering and Technology (ICETECH), pp. 611–616, March 2016.

[12] U. Grenoble, U. Montpellier, U. Paris, and U. Versailles, "Mobile databases: a selection of open issues and research directions," SIGMOD RECORD, vol. 33, no. 2, pp. 78–83, 2004.

[13] P. Kalyanakumar and A. Sangeetha, "A synchronization algorithm for mobile databases using samd," 2015.

[14] Mopati B.Kekgathetse and Keletso J.Letsholo,"A Survey on Database Synchronization Algorithms For Mobile Devices " in JATIT & LLS 2016 Vol. 86. No.1.

[15] Ranjeet Singh and Chiranjit Dutta "A Synchronization Algorithm of Mobile Database for Cloud Computing" International Journal of Innovation in Engineering and Management, Vol. 2, No.3, March 2013

[16] Bhagat, A.R and Prof. Bhagat, V.B, "Mobile Database Review and Security Aspects", International Journal of Computer Science and Mobile Computing, Vol. 3, No. 3, March 2014.

[17] D. Sethia, S. Mehta, A. Chowdhary, K. Bhatt and S. Bhatnagar, "MRDMS-mobile replicated database management synchronization," *2014 International Conference on Signal Processing and Integrated Networks (SPIN)*, Noida, 2014, pp. 624-631.

[18] Becker-Bessudo, D & Fernandez-Anaya, G & Flores-Godoy, Jose-Job. (2009). Preserving synchronization under characteristic polynomial modifications. IFAC Proceedings Volumes (IFAC-PapersOnline). 2. 187-192.

[19] Lu, Z. Y., & Guo, Z. B. (2015, July). A method of data synchronization based on message oriented middleware and xml in distributed heterogeneous environments. In *2015 International Conference on Artificial Intelligence and Industrial Engineering*. Atlantis Press.

[20] B. A. Kitchenham and S. Charters, "Guidelines for performing Systematic Literature Reviews in Software Engineering," Version 2.3. EBSE Technical Report, 2007.

[21] Nidhi Singh, Muzammil Hasan ," A Survey on Synchronization Algorithms for Mobile Computing", International Journal of Scientific Research and Review (IJSRR) , Vol. No. 08, Issue- 04, April 2019, ISSN NO: 2279-543X.