

GLOBAL JOURNAL

OF COMPUTER SCIENCE AND TECHNOLOGY: C

Software & Data Engineering

Mediation of Lazy Update

Decentralized P2P Architecture

Highlights

Techniques for Computing Data

Algorithm and Design Techniques

Discovering Thoughts, Inventing Future

VOLUME 19 ISSUE 3 VERSION 1.0



GLOBAL JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY: C
SOFTWARE & DATA ENGINEERING



GLOBAL JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY: C
SOFTWARE & DATA ENGINEERING

VOLUME 19 ISSUE 3 (VER. 1.0)

OPEN ASSOCIATION OF RESEARCH SOCIETY

© Global Journal of Computer Science and Technology. 2019.

All rights reserved.

This is a special issue published in version 1.0 of "Global Journal of Computer Science and Technology" By Global Journals Inc.

All articles are open access articles distributed under "Global Journal of Computer Science and Technology"

Reading License, which permits restricted use. Entire contents are copyright by of "Global Journal of Computer Science and Technology" unless otherwise noted on specific articles.

No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy, recording, or any information storage and retrieval system, without written permission.

The opinions and statements made in this book are those of the authors concerned. Ultraculture has not verified and neither confirms nor denies any of the foregoing and no warranty or fitness is implied.

Engage with the contents herein at your own risk.

The use of this journal, and the terms and conditions for our providing information, is governed by our Disclaimer, Terms and Conditions and Privacy Policy given on our website <http://globaljournals.us/terms-and-condition/menu-id-1463/>

By referring / using / reading / any type of association / referencing this journal, this signifies and you acknowledge that you have read them and that you accept and will be bound by the terms thereof.

All information, journals, this journal, activities undertaken, materials, services and our website, terms and conditions, privacy policy, and this journal is subject to change anytime without any prior notice.

Incorporation No.: 0423089
License No.: 42125/022010/1186
Registration No.: 430374
Import-Export Code: 1109007027
Employer Identification Number (EIN):
USA Tax ID: 98-0673427

Global Journals Inc.

(A Delaware USA Incorporation with "Good Standing"; Reg. Number: 0423089)

Sponsors: Open Association of Research Society

Open Scientific Standards

Publisher's Headquarters office

Global Journals® Headquarters
945th Concord Streets,
Framingham Massachusetts Pin: 01701,
United States of America

USA Toll Free: +001-888-839-7392

USA Toll Free Fax: +001-888-839-7392

Offset Typesetting

Global Journals Incorporated
2nd, Lansdowne, Lansdowne Rd., Croydon-Surrey,
Pin: CR9 2ER, United Kingdom

Packaging & Continental Dispatching

Global Journals Pvt Ltd
E-3130 Sudama Nagar, Near Gopur Square,
Indore, M.P., Pin:452009, India

Find a correspondence nodal officer near you

To find nodal officer of your country, please
email us at local@globaljournals.org

eContacts

Press Inquiries: press@globaljournals.org
Investor Inquiries: investors@globaljournals.org
Technical Support: technology@globaljournals.org
Media & Releases: media@globaljournals.org

Pricing (Excluding Air Parcel Charges):

Yearly Subscription (Personal & Institutional)
250 USD (B/W) & 350 USD (Color)

EDITORIAL BOARD

GLOBAL JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY

Dr. Corina Sas

School of Computing and Communication
Lancaster University Lancaster, UK

Dr. Sotiris Kotsiantis

Ph.D. in Computer Science, Department of Mathematics,
University of Patras, Greece

Dr. Diego Gonzalez-Aguilera

Ph.D. in Photogrammetry and Computer Vision Head of
the Cartographic and Land Engineering Department
University of Salamanca Spain

Dr. Yuanyang Zhang

Ph.D. of Computer Science, B.S. of Electrical and
Computer Engineering, University of California, Santa
Barbara, United States

Dr. Osman Balci, Professor

Department of Computer Science Virginia Tech, Virginia
University Ph.D. and M.S. Syracuse University, Syracuse,
New York M.S. and B.S. Bogazici University, Istanbul,
Turkey

Dr. Kwan Min Lee

Ph. D., Communication, MA, Telecommunication,
Nanyang Technological University, Singapore

Dr. Khalid Nazim Abdul Sattar

Ph.D, B.E., M.Tech, MBA, Majmaah University,
Saudi Arabia

Dr. Jianyuan Min

Ph.D. in Computer Science, M.S. in Computer Science, B.S.
in Computer Science, Texas A&M University, United States

Dr. Kassim Mwitondi

M.Sc., PGCLT, Ph.D. Senior Lecturer Applied Statistics/
Data Mining, Sheffield Hallam University, UK

Dr. Kurt Maly

Ph.D. in Computer Networks, New York University,
Department of Computer Science Old Dominion
University, Norfolk, Virginia

Dr. Zhengyu Yang

Ph.D. in Computer Engineering, M.Sc. in
Telecommunications, B.Sc. in Communication Engineering,
Northeastern University, Boston, United States

Dr. Don. S

Ph.D in Computer, Information and Communication
Engineering, M.Tech in Computer Cognition Technology,
B.Sc in Computer Science, Konkuk University, South
Korea

Dr. Ramadan Elaiess

Ph.D in Computer and Information Science, University of
Benghazi, Libya

Dr. Omar Ahmed Abed Alzubi

Ph.D in Computer and Network Security, Al-Balqa Applied
University, Jordan

Dr. Stefano Berretti

Ph.D. in Computer Engineering and Telecommunications, University of Firenze Professor Department of Information Engineering, University of Firenze, Italy

Dr. Lamri Sayad

Ph.d in Computer science, University of BEJAIA, Algeria

Dr. Hazra Imran

Ph.D in Computer Science (Information Retrieval), Athabasca University, Canada

Dr. Nurul Akmar Binti Emran

Ph.D in Computer Science, MSc in Computer Science, Universiti Teknikal Malaysia Melaka, Malaysia

Dr. Anis Bey

Dept. of Computer Science, Badji Mokhtar-Annaba University, Annaba, Algeria

Dr. Rajesh Kumar Rolan

Ph.D in Computer Science, MCA & BCA - IGNOU, MCTS & MCP - Microsoft, SCJP - Sun Microsystems, Singhania University, India

Dr. Aziz M. Barbar

Ph.D. IEEE Senior Member Chairperson, Department of Computer Science AUST - American University of Science & Technology Alfred Naccash Avenue Ashrafieh, Lebanon

Dr. Chutisant Kerdvibulvech

Dept. of Inf. & Commun. Technol., Rangsit University Pathum Thani, Thailand Chulalongkorn University Ph.D. Thailand Keio University, Tokyo, Japan

Dr. Abdurrahman Arslanyilmaz

Computer Science & Information Systems Department Youngstown State University Ph.D., Texas A&M University University of Missouri, Columbia Gazi University, Turkey

Dr. Tauqeer Ahmad Usmani

Ph.D in Computer Science, Oman

Dr. Magdy Shayboub Ali

Ph.D in Computer Sciences, MSc in Computer Sciences and Engineering, BSc in Electronic Engineering, Suez Canal University, Egypt

Dr. Asim Sinan Yuksel

Ph.D in Computer Engineering, M.Sc., B.Eng., Suleyman Demirel University, Turkey

Alessandra Lumini

Associate Researcher Department of Computer Science and Engineering University of Bologna Italy

Dr. Rajneesh Kumar Gujral

Ph.D in Computer Science and Engineering, M.TECH in Information Technology, B. E. in Computer Science and Engineering, CCNA Certified Network Instructor, Diploma Course in Computer Servicing and Maintenance (DCS), Maharishi Markandeshwar University Mullana, India

Dr. Federico Tramarin

Ph.D., Computer Engineering and Networks Group, Institute of Electronics, Italy Department of Information Engineering of the University of Padova, Italy

Dr. Roheet Bhatnagar

Ph.D in Computer Science, B.Tech in Computer Science, M.Tech in Remote Sensing, Sikkim Manipal University, India

CONTENTS OF THE ISSUE

- i. Copyright Notice
- ii. Editorial Board Members
- iii. Chief Author and Dean
- iv. Contents of the Issue

1. Mediation of Lazy Update Propagation in a Replicated Database over a Decentralized P2P Architecture. *1-31*
2. Study and Performance Analysis of Different Techniques for Computing Data Cubes. *33-42*
3. Algorithm and Design Techniques – A Survey. *43-52*

- v. Fellows
- vi. Auxiliary Memberships
- vii. Preferred Author Guidelines
- viii. Index



Mediation of Lazy Update Propagation in a Replicated Database over a Decentralized P2P Architecture

By Katembo Kituta Ezéchiél, Shri Kant & Ruchi Agarwal

Sharda University

Abstract- While replicating data over a decentralized Peer-to- Peer (P2P) network, transactions broadcasting updates arising from different peers run simultaneously so that a destination peer replica can be updated concurrently, that always causes transaction and data conflicts. Moreover, during data migration, connectivity interruption and network overload corrupt running transactions so that destination peers can experience duplicated data or improper data or missing data, hence replicas remain inconsistent. Different methodological approaches have been combined to solve these problems: the audit log technique to capture the changes made to data; the algorithmic method to design and analyse algorithms and the statistical method to analyse the performance of new algorithms and to design prediction models of the execution time based on other parameters. A Graphical User Interface software as prototype, have been designed with C #, to implement these new algorithms to obtain a database synchronizer-mediator. A stream of experiments, showed that the new algorithms were effective. So, the hypothesis according to which “The execution time of replication and reconciliation transactions totally depends on independent factors.” has been confirmed.

Keywords: peer-to-peer (P2P), database replication, data reconciliation, transaction serialization, synchronizer-mediator.

GJCST-C Classification: C.2.4



Strictly as per the compliance and regulations of:



Mediation of Lazy Update Propagation in a Replicated Database over a Decentralized P2P Architecture

Katembo Kituta Ezéchiel ^α, Shri Kant ^ο & Ruchi Agarwal ^ρ

Abstract- While replicating data over a decentralized Peer-to-Peer (P2P) network, transactions broadcasting updates arising from different peers run simultaneously so that a destination peer replica can be updated concurrently, that always causes transaction and data conflicts. Moreover, during data migration, connectivity interruption and network overload corrupt running transactions so that destination peers can experience duplicated data or improper data or missing data, hence replicas remain inconsistent. Different methodological approaches have been combined to solve these problems: the audit log technique to capture the changes made to data; the algorithmic method to design and analyse algorithms and the statistical method to analyse the performance of new algorithms and to design prediction models of the execution time based on other parameters. A Graphical User Interface software as prototype, have been designed with C # (C S harp), to implement these new algorithms to obtain a database synchronizer-mediator. A stream of experiments, showed that the new algorithms were effective. So, the hypothesis according to which “The execution time of replication and reconciliation transactions totally depends on independent factors.” has been confirmed.

Keywords: peer-to-peer (P2P), database replication, data reconciliation, transaction serialization, synchronizer-mediator.

I. INTRODUCTION

In computing, a Distributed Database System (DDBS) is a database whose storage devices are not necessarily all linked to a common processing unit; but rather in this approach, the database can be stored on multiple computers, located in the same physical location or can be scattered on networked computers [1], [8]. The distribution transparency is the fundamental principle of the DDBS which consists of making a distributed system to appear similar to a centralized system to the users. The distribution transparency as well as the management of a DDBS are ensured by a program called Distributed Database Management System (DDBMS)

Author α: Ph.D. scholar, Department of Computer Science and Engineering, Sharda University, Greater Noida, India. e-mail: kkitutaezechiel@yahoo.com

Author ο: Professor, Research and Technology Development Centre, Department of Computer Science and Engineering, Sharda University, Greater Noida, India. e-mail: shri.kant@sharda.ac.in

Author ρ: Associate Professor, Department of Computer Applications, JIMS Engineering Management Technic al Campus, Greater Noida, India. e-mail: dr.ruchi@outlook.com

[3]. The design of a DDBS requires that it be entirely resident on different sites of a computer network but not necessarily all. This means that at least two sites must host the database and not necessarily each site in the network, as depicted in the Fig. 1.

Thus, there are two distribution strategies: data fragmentation and data allocation on the one hand and data replication on the other hand. So, to make a good design, all these strategies are compiled [2], [3], [33]. The fragmentation consists in splitting a relation (a table of a database) into a number of sub-relations, called fragments; which can be horizontal, vertical or hybrid. Horizontal fragments are subsets of tuples (table records), vertical fragments are subsets of attributes (table columns), and hybrid fragmentation consists of mixing the two preceding ones. In turn the allocation is nothing more than the assignment of fragments to the sites in an optimal way [2]. When allocated fragments have to share data among them, they need the replication procedure.

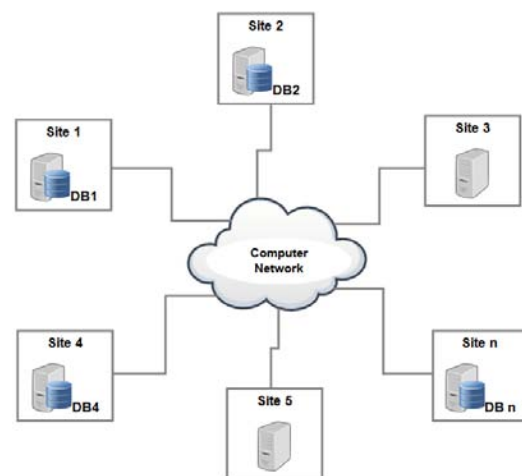


Fig. 1: Architecture of Distributed Database System.

However, this work focuses on the data replication strategy. The replication consists of duplication and storage of multiple copies or replicas (at least two) of the same fragment or the entire relation (in the case of a fully replicated database) of a DDBS in multiple different sites. The replication is the strategy used to ensure the data exchange

between fragments or relations in a fully replicated database [2], [3], [4], as illustrate in Fig. 2. In any case, the main problem of the data replication is the synchronization of replicas. Data synchronization is nothing than keeping consistent replicas in a Replicated Database System (RDBS) [5]. This means ens using the exchange of updates between replicas.

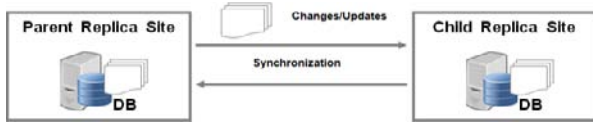


Fig. 2: Protocol of Database replication.

Nowadays P2P computer network is in full emergence. Comparatively to client/server model, in a P2P system, each client is itself a server. In this way replicating a Database over a P2P network require that all peers keep the same data copy. In the same way, the emergence of advanced applications of P2P systems, requiring general replication capabilities with different levels of granularity and multi-master mode [11], where each peer can transfer updates to all others and the same replica can be updated by several peers in a replicated databases environment [4], [10], the serialization of updates and the reconciliation of data turns out to be the particular P2P replication problems because those flows of updates (data) and refresh transactions conflict each other [8], [30], [33].

For example, the operations on an account, of a customer, opened in a bank with multiple branches can be replicated by several branches of the same bank and must be able to be updated by any branch anytime, to acquire reception of a transfer, for a deposit to the account, a withdrawal from the account, etc. Concretely, changes made by refresh transactions from different peers reach a destination site at the same time and multiple updates of the same replicas by different peers break the reliability and the consistency of replicas [2].

This is why this study aims to introduce an effective approach to serialize refresh transactions and to reconcile replicas in the case of inconsistency. To overcome one of DDBS homogeneity aspects, namely the same DBMS, the result of this design needs to be implemented as a synchronizer-mediator for database replication in a Graphical User Interface (GUI) using lazy decentralized sites strategy on a P2P network. To reach this purpose, the structure of this paper is organized as follow: the first section introduced by presenting the context of this research as well as the status of the problem, the second section will review the related works, the third will present the methodology, fourth section will show the simulation environment for experimentation, the fifth section will offer the result and finally the sixth section will conclude this study.

II. RELATED WORKS

This section will rapidly review certain research works already realized to attempt to solve these two aforementioned problems.

a) Data replication

Designing a RDBS pursue four majeure objectives, namely : improving data availability, improving performance, ensuring scalability and users applications requirements. These purposes can be summarized as "improving consistency and/or reliability" [2], [3]. To ensure consistency between replicas, the synchronization procedure uses the transaction running technique. A transaction is a collection of operations that transforms the database from a consistent state to another consistent state [6], as illustrated in Fig. 3.

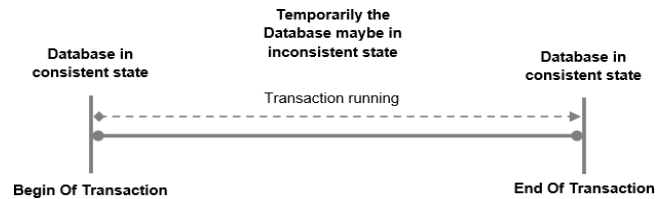


Fig. 3: Protocol of Transaction running.

A transaction has a Begin Of Transaction (BOT) and an End Of Transaction (EOT). This End is managed by three different functions: either a "commit" to validate, a "rollback" to cancel, or an "abort" to interrupt the execution of operations inside the transaction. The consistency and/or reliability of a transaction are guaranteed by 4 properties: Atomicity, Coherence, Isolation, Durability (ACID) that make the "acidity" of a transaction [2], [7]. As we are dealing with data flow, our focus remains on the Structured Query Language (SQL) operators, especially the Data Manipulation Language operators in most of DDBMSs, which contains [9]: The write operators (Insert, Update and Delete SQL commands) and the read operator (Select SQL command). Typically, like the structuring of instructions of a procedural language, a transaction "T" can have the following structure:

```

Begin_Of_Transaction T
Insert operator
Update operator
Delete operator
Select operator
End_Of_Transaction T
    
```

However, to solve the aforementioned main problem of data replication, i.e. the synchronization of replicas, there already exists four replication strategies, resulting from the combination of two factors: "when" and "where". The "when" factor specifies when

updates are broadcasted (synchronously/eagerly or asynchronously/lazily), while the "where" factor indicate where updates occur on a centralized site (primary copy/mono- master) or on decentralized sites (everywhere/multi-master) before being propagated. So when we take the factor "where" in "when", it emerges [1], [2], [3], [4], [30], [33], [34]:

A. *Synchronous or Eager Replication*: All replicas must be updated before the transaction commit i.e. in real-time. Here, the most up to date value of an item is guaranteed to the end user. There are two different strategies in synchronous replication:

- 1) *Eager centralized site*: This method is beneficial in case where reads are much more frequent than writes. It works under the principle "Read-One, Write-All (ROWA)". After transaction commitment, any one of replicas can be read; so the write process must update all replicas.
- 2) *Eager decentralized sites*: The principle is "update everywhere"; in this logic every site is allowed to propagate updates to all sites in the same transaction, at the same time so that on the end of the transaction updates become available on all sites.

B. *Asynchronous or Lazy Replication*: Allows different replicas of the same object to have different values for a short periods of time i.e. in near real-time. They are updated after a predefined interval of time. There are two different strategies in asynchronous replication:

- 1) *Lazy centralized site*: It works with the principle such that one copy of replicas is assigned as the "primary copy or mono-master" so that changes of data or writes are possible only on it. These changes are periodically propagated to the secondary copies. The secondary copies of data can only be read.
- 2) *Lazy decentralized sites*: Here the principle is so that changes can be performed "everywhere or multi - master", on each site. So these changes are propagated independently to other sites sporadically.

These replication strategies, have already been implemented in most of modern DDBMSs [9]. It is largely the centralized strategy that is much more wrapped in the replication models offered by almost all DBMSs. But, although these modelling are done, there remains a problem to emphasize in eager centralized site approach such that if there is a site unavailable during updates propagation by the master site, the transaction cannot commit. So, some researches are already attempting to design an optimal algorithm that can allow the update

transaction commitment on the available sites and to update unavailable sites as soon as they become available again; hence the approach "Read-One, Write-All Available (ROWA-A)" [2], [30], [33]. In addition, one could expect the problem related to the momentary interpolation of the line of communication between the master site and the slave sites, because it is enough for example that the master site overlord or be inaccessible so that the slaves no more access to updates [8]. Well, there is only the decentralized strategy that can clear this concern.

Nevertheless, eager decentralized sites experience the same problem as eager centralized site, whereby update transactions that arise from all sites, if they find at least one site unavailable they abort. But to overcome this problem, such kind of systems should be able first of all to commit transactions on only available sites and so update unavailable sites as soon as they become available again; hence the approach "Update Everywhere Available" [17]. So nowadays, some researches attempt to improve these algorithms by distributed voting algorithm [4]. Thus, if the sites number quorum is reached the transaction commit on them; so afterwards, when writing, update all fraction of the replicas and when reading, read enough replicas to ensure you get at least one copy of the most recent value.

In view of the above, it seems that the lazy strategy is appropriate for P2P topology, especially since it allows replicas of various sites to diverge for a given moment. So as in a P2P network, the participants (Peers) are present or absent momentarily, updates propagation can be applicable on the present Peers while the absent Peers will remain with non - updated replicas in order to receive their updates when they become available again [10], [33]. Thus, lazy centralized sites approach is appropriate for the centralized P2P topology because updates are performed only on the central site and then forwarded to slave sites in near real-time while lazy decentralized sites approach is the most appropriate for the materialization of replication on a decentralized P2P topology because in near real-time, like centralized approach, updates can be performed everywhere, i.e. on each peer and then be broadcasted to all others.

Referring on our problem concerning replication over a decentralized P2P architecture, the observation has been that only a few of DDBMSs have already tried to implement the lazy decentralized strategy in order to formalize the P2P replication; let us quote for instance SQL Server [13] and Oracle [14]. Unfortunately, the particular problems of P2P replication still exist and will be developed in following lines:

- *Transaction conflicts:* Several updates carried by refreshing transactions, from different sites reach a destination site at the same time but they cannot be performed on the same time, then reliability and consistency will be lost and there will be the risk of transaction conflicts [2], [30], [33], [35]. DDBMSs must ensure that transaction execution meets a set of properties that lead to the consistency of distributed databases and conveniently summarized by the ACID, since when the execution is always concurrent [6], [7]. Thus, several researches have already been undertaken to solve the transaction concurrency control problem. Concurrent execution without harmonization constraints poses a number of problems, the most important of which is the loss of operations and incorrect readings. Therefore, it is necessary to set the serializability, a property determining a correct execution of the completion of transactions [3].
- *Data conflicts:* P2P replication allows to perform changes on each peer in the topology and then forward them to other peers. However, as changes are performed at different peers, probable data conflicts are to be pointing out when modifications are being broadcasted [2], [30], [33]. Thus, in all DDBMSs which have already succeed to implement the lazy decentralized sites approach to make it P2P replication, one can distinguish three types of data conflicts [13], [14], [20], [21]:
 - a) *Primary key or uniqueness conflict:* Occurs when a record with the same primary key has been created and inserted at more than one peer in the topology. So when those peers need to exchange updates, it is then impossible to violate the criterion of entity integrity;
 - b) *Foreign key conflict:* Can occurs if in any case the refresh transaction forward updates which contains a record with a foreign key column but whose primary key is not yet forwarded to the destination peer. So it is then impossible to violate the criterion of referential integrity;
 - c) *Data modifications conflicts:*
 - ✓ Update conflict: occurs when the same record has been updated on more than one peer;
 - ✓ Insertion/Update conflict: occurs when a record has been updated on a peer and the same record has been deleted and re-inserted on another peer;
 - ✓ Insert/Delete conflict: occurs when a record has been deleted on a peer and the same record has been deleted and re-inserted on another peer;
 - ✓ Update/Delete conflict occurs when a record has been updated on one peer and the same record has been deleted on another peer;

- ✓ Deletion conflict: occurs when a record has been deleted on more than one peer.

Thus it is necessary to think about a certain number of rules to warranty the conflict policy avoidance in the decentralized P2P replicated environment. Apart from the inconsistency of data caused by transaction conflicts and data conflicts, there are other phenomena which make the replicated data inconsistent. Thus, although the transaction that propagates the updates is successfully committed, the data remains inconsistent. Hence, there is the need of an automatic data reconciler.

b) *Data reconciliation*

Database reconciliation is a process of verifying data when there has been a migration or transfer of data from a source database to a destination. The purpose of this process is to ensure that the migration has been done accurately [22]. In this logic, in a global manner, the data is the set of tables of a given database and in a basic way, the set of records of definite tables which can be accessed by a certain selection criterion. In a replicated Databases environment, updates broadcasting as well consists to migrate or to transfer data changes from a Primary site toward Secondary sites [23].

However, during data migration, errors may have occurred [12]. Most are like execution failures due to network interruptions as well as network overload those end up corrupting transactions and causing data to be lost or remain in an invalid state at the destination [8], [34]. These phenomena lead to a series of problems such as: missing records, duplicate records, incorrect values, missing values, incorrectly formatted values, broken relationships between tables in case of forced redundancy, etc. [22]. But, some researches have already been undertaken to find solutions in several ways and some algorithms are already implemented in DDBMS and particular software to reconcile data after migration process.

Oracle Corporation [24], possesses some databases reconciliation tools for their DDBMSs: Upgrade Reconciliation Toolkit is used to compare the data on the Oracle DB source and Oracle DB destinations after data migration and after running the parallel End Of Day (EOD) activities mostly for different branches of a bank. This tool generates also the reconciliation report at the end of the process. Another tool is mysqlbcompare especially for MySQL, this tool compares two databases by identifying differences between databases objects; changed or missing rows of tables are shown in standard formats like grid, table, etc. It is going beyond the data comparison; this utility compare also objects data definition of two databases [25].

H. Jonathan [26], implemented a PHP script, to produce MySQL_Diff tool, a Web application running in a browser, to reconcile two MySQL databases schema difference by visualizing databases tools and resolving differences. ApexSQL LLC [29], a Microsoft Gold Certified Partner, provide ApexSQL Data Diff, a Windows application to compare tables in the databases and visualize the difference in a grid before synchronizing two different remote sources SQL Server databases. Slotix s.r.o. [27], provide DBConvert, a Windows application to migrate data (1 Million records in 5-10 minutes) between multiple databases and DBSync, a customized Windows application as well, to compare (missing and additional records) and synchronize data between two different databases.

Pragmatic Works Inc. [28], a Microsoft Gold Certified Partner, provide another Data Reconciliation Tool LegiTest's, which can be connected to a variety of data sources, mostly for Microsoft so that data verification can be perform cross-platform. It supports SQL Server, Oracle, SSAS, OLE DB sources, and ODBC sources. Experian Ltd. [22], provides Experian Aperture Data Studio, a Windows application for data migration and data reconciliation between a source and a destination database.

Nevertheless, all these tools run reconciliation between one source and one destination. The only one which can reconcile one source and multiple destinations is Upgrade Reconciliation Toolkit for Oracle. Unfortunately, it is only limited to Oracle DB. The tools mysqldbcompare and MySQL_Diff are also limited to MySQL and they are not taking in to account multiple destinations. The Tool LegiTest's should be more interesting because it is able to reconcile multi-DBMS databases, but it is also one source, one destination; and all others which have been listed in this review present such kind of limitation.

Moreover, these data reconciliation tools rely on simple counting of records to keep track if the expected number of records has been migrated. It can be esteemed that this was mainly due to the importance of the processing of essential data to carry out field validation of a given data. Nowadays, for more accuracy, the data migration algorithm should provide data reconciliation capabilities that allow the reconciliation of each data or each field, i.e. at the intersection of each row and each column (attributes by record) of each database table [12].

To preserve data inconsistency and to maintain acidity, all instructions of the replication procedure must be wrapped in transactions [2], [7]. The instructions of a transaction are the commands or operators of the data manipulation language. But, when an operator of the data modification language is executed on a site, some time passes while waiting

for the response. While a transaction may have more than one operator and the factors are likely to be varied in a P2P environment, this phenomenon should greatly influence the temporal complexity in the event of variation of different factors. So it is necessary to design a prediction model of replication and reconciliation execution time.

The assumption of this study is formulated as follows: "it seems that P2P replication systems experience the weak performance, especially since the time to replicate and to reconcile data from a Master Peer to Slave Peers dependent, if not totally, partially of certain factors, such as: the number of records in each table, the number of tables whose data has changed, the number of peers connected during the propagation of updates and other factors (number of columns per table, data types columns, etc.)".

However, these problems deserve a special attention; that is why there is a reason to wonder about setting up "a synchronizer-mediator for lazy replicated databases over a decentralized P2P architecture". This system should be able to serialize updates performed simultaneously on different replicas of the same database and to reconcile this replicas, effectively, over a decentralized P2P network.

III. METHODOLOGY

To ensure strong replica consistency in a distributed database, traditionally the implementation of a synchronous or eager refresh algorithm which is specially Two-Phase-Commit (2PC) based technique is the unique gateway to avoid discrepancies between replicas [2]. However, this solution is inapplicable in a P2P architecture because does not guarantee the updates delivery to all peers as they are not all always available at the same time [15]. Thus, asynchronous or lazy replication is more appropriate for P2P systems because it allows replicas to be updated independently and to remain divergent until a refresh transaction takes place [16]. Modifications which have been done to the local replica, by local transactions are captured and the refresh transaction propagates them to remote replicas asynchronously i.e. in near real-time. The technique used in this work to capture modifications is audit-log.

a) *Audit-log technique*

Almost all DDBMSs support this technique by running triggers belonging to a specific table in order to capture data modifications. A trigger is attached to an event produced by an Insert or Update or Delete operator so that it captures changes before or after the event has taken place in the database [5], [33]. So, in this work the interest is carried on after trigger. To achieve this, for each data-table the creation of one

audit-table is necessary. The audit-table is composed by the data-table primary key column, other data-table columns (apart from the primary key), the updated column name, the audit action, the timestamp and the synchronization ID. These

elements are required for a record to do the comparison between data. Each table in the database would need three triggers to run after Insert, after Update and after Delete. The flow chart, Fig. 4 here below illustrates the audit-log creation.

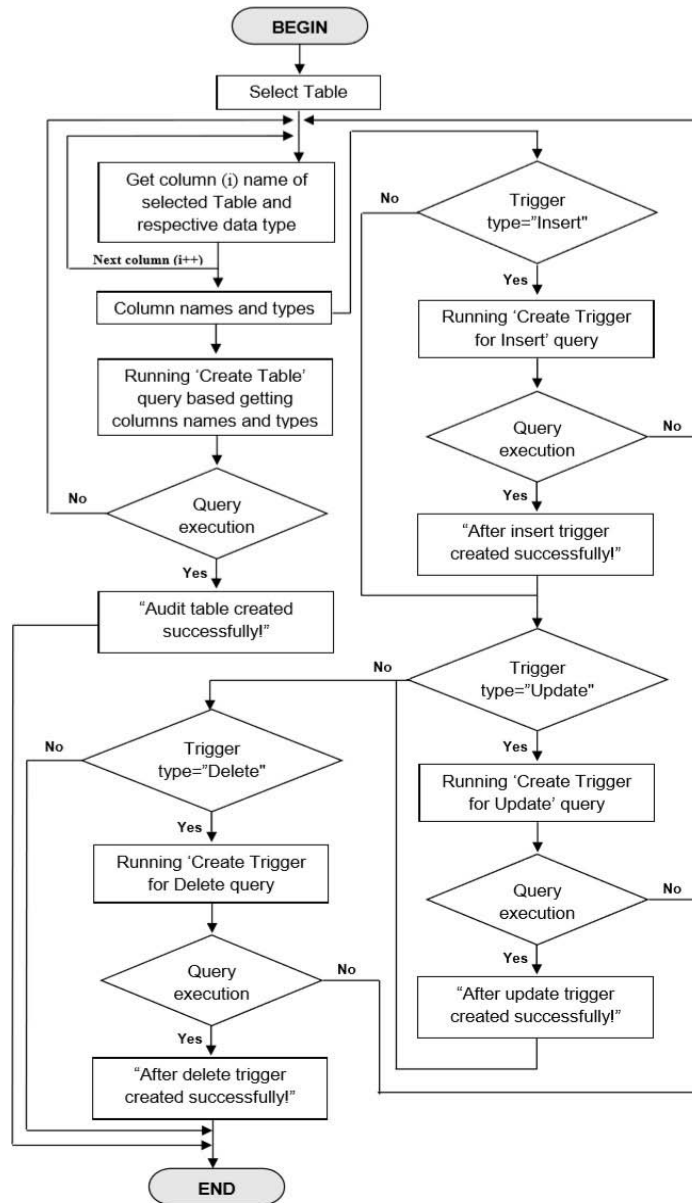


Fig. 4: Audit -log (Audit table and triggers) creation.

Suppose that the database is homogenous and full replicated, as soon as the audit log creation of each data table completed, on each peer, for each SQL data modification operation, the DDBMS performs following action accordingly:

- After each Insert operation in the data table, the “insert trigger” captures the newly added record and inserts it in the audit table, as shown in Fig. 6, row 1 to 5 in Slave Peer Audit-table;
- After each Update operation of a column of data table, the “update trigger” captures the

concerned record, with the new data that has just been set, and inserts it in the audit table, as shown in Fig. 6, row 6 to 8 in Slave Peer Audit-table;

- After each Delete operation from the data table, the “delete trigger” captures the deleted record and inserts it in the audit table, as shown in Fig. 6, row 9 and 10 in Slave Peer Audit-table.

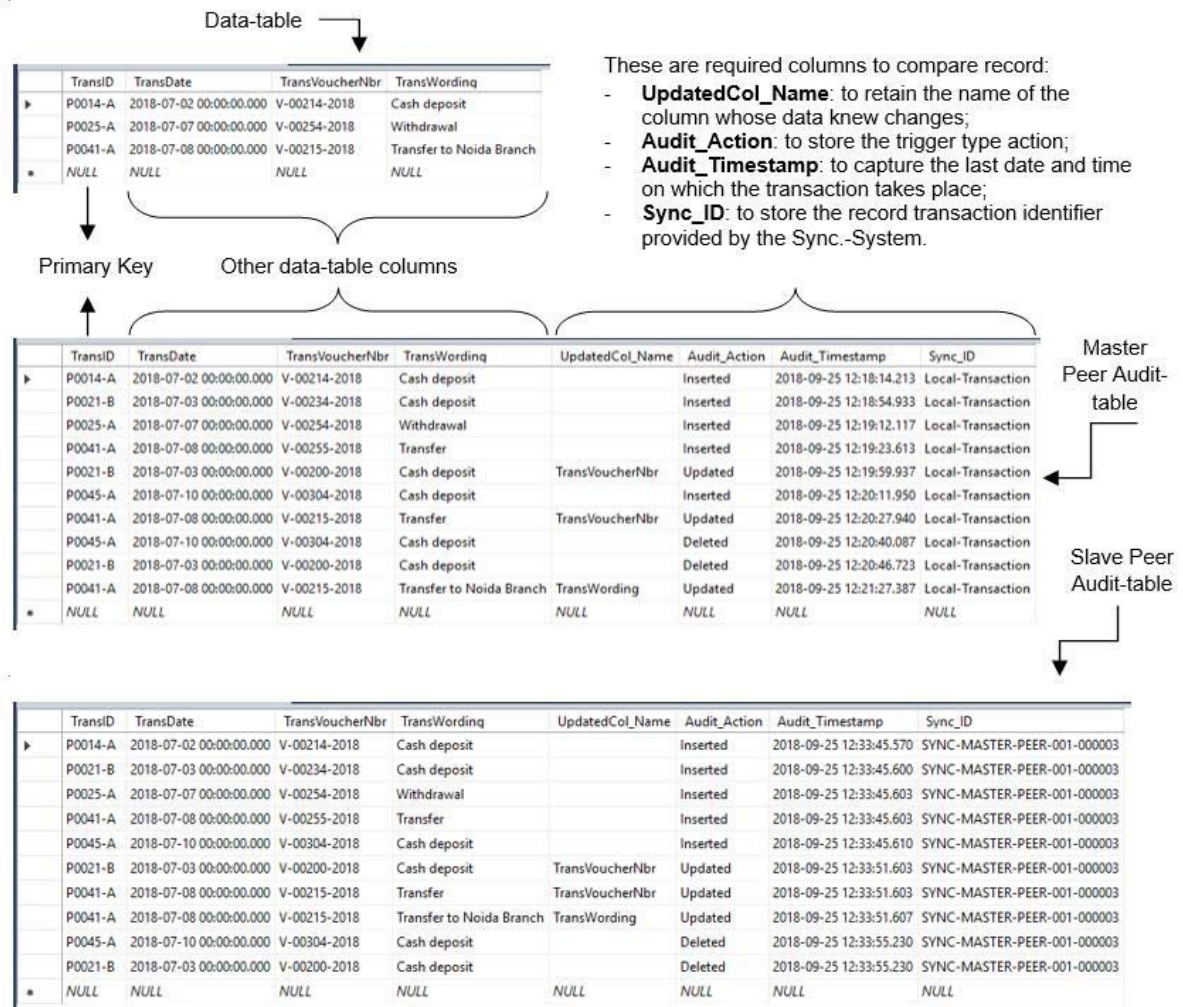


Fig. 5: Data table triggered and audit tables Master and Slave.

The column synchronisation ID (Sync_ID) in Audit -tables don't have same value; for a Master Peer Audit-table its content is "Local-Transaction", value automatically provided by the trigger procedure when the transaction is initiated locally by the user application whereas for a remote transaction the synchronization procedure update automatically this column by the sync. ID provided by the Sync. Mediator-System. So, the synchronization procedure select only data whose Sync_ID is equal to "Local-Transaction" and whose Audit_Timestamp is in the interval of begging date and time to ending date and time and apply them to Slave Peers according to the Audit_Action value. This technique permits us to resolve the problem of the endless loop in the sync. procedure used two -ways or symmetrical replication which was knowing old synchronizers [5].

b) Algorithmic method

The Algorithmic method will be used to design and to analyse instructions of algorithms and steps of a Peer-to-Peer Synchronizer. This method will take in account the Circulating Token Ring Algorithm, the Decentralized Peer-to-Peer Replication

Algorithm and the Decentralized Peer-to-Peer Data Reconciliation Algorithm.

i. Network Topology and Algorithm

When a peer needs to broadcast its captured updates toward other peers, it needs a token which gives it the state of a Master i.e. the permission to forward its updates and other peers become automatically Slaves. A fully replicated P2P database system includes *p* peers and each peer has a complete copy of the database. Peers communicate with each other by exchanging messages and forwarding updates or accessing peer data by performing transactions [17]. In this way, updates will be applied according to a circulating token, as depicted in Fig. 6, which determine transactions serialization order or one can give the privilege to updates from certain sites considered to be more important or privileged.

Suppose a network consisting of four peers A, B, C and D all networked. The Fig. 6 below presents the decentralized topology of peer-to-peer token ring network.

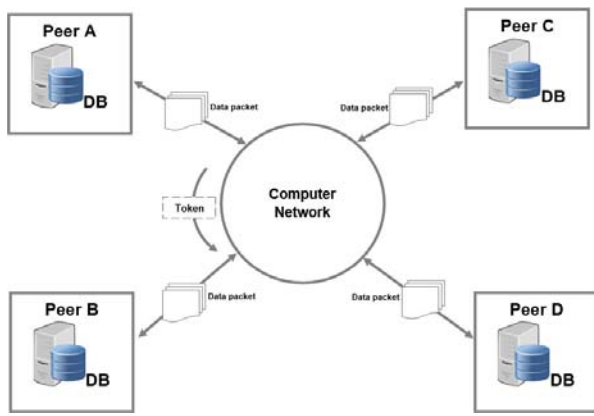


Fig. 6: Topology of Decentralized P2P circulating Token Ring.

A predefined order of releasing or getting the token, since we are in a P2P network where a peer p may or may not be available, is not needed. The optimization policy here is to give the token directly to a peer which needs it instead of going through a list of peers that we are not sure of their availability at the time of the token release. So, transaction serialization is managed by the new circulating token algorithms 1 and 2, successively for getting the token and releasing the token.

Algorithm 1: P2P getting the circulating token

```

Input: A set  $p$  of slave peers
Output: A peer (Master/Slave) owning the token
begin getTokenFunction()
1:  select all SlavePeers
2:  for ( $p \leftarrow 0$  to NumberOfSlavePeers - 1) do
3:    if (SlavePeer( $p$ ).ConnexionState = "True") then
4:      SlavePeer( $p$ ).Availability ← "True"
5:    else
6:      SlavePeer( $p$ ).Availability ← "False"
7:    end if
8:  end for  $p$ 
9:  select AvailableSlavePeers
10: for ( $p \leftarrow 0$  to NumberOfAvailableSlavePeers - 1) do
11:   select Token from SlavePeer( $p$ )
12:   for ( $j \leftarrow 0$  to NumberOfSyncIDInSlavePeer( $p$ ) - 1)
13:     if SyncID( $j$ ).TokenPossession = "True" then
14:       SlavePeer.TokenPossState ← "True"
15:       SlavePeer.SyncID ← SlavePeer( $p$ ).SyncID( $j$ )
16:     end if
17:   end for  $j$ 
18:   if (SlavePeer( $p$ ).TokenPossState = "True") then
19:     TokenAvailability ← "false"
20:     select Token from MasterPeer
21:     if (MasterPeer.TokenPossState = "True") then
22:       Set MasterPeer.TokenPossState ← "False"
23:       Set MasterPeer.TokenReleaseDateTime ← now()
24:     end if
25:     if (MasterPeer.Privilege = "True") then
26:       for ( $p \leftarrow 0$  to NumberOfAvailableSlavePeers - 1) do
27:         if (SlavePeer( $p$ ).Privilege = "False") then
28:           Set SlavePeer( $p$ ).TokenPrevention ← "True"
29:         end if
30:       end for  $p$ 
31:       Send TokenRequest to SlavePeer( $p$ )
32:     else
33:       Send TokenRequest to SlavePeer( $p$ )

```




```

34:   end if
35:   else
36:     if( $p = \text{NumberOfAvailableSlavePeers} - 1$ )then
37:       Set MasterPeer.TokenPossState = "True"
38:       Set MasterPeer.TokenReceptionDateTime = now()
39:     end if
40:   end if
41: end for p
42: return a peer (Master/Slave) owning the token
end getTokenFunction

```

Since when a peer (p), which can be "A" or "B" or "C" or "D" gets the token, it executes the transactions according to the algorithm 3, 4 and 5 for data replication and 6 for data reconciliation. Consequently, all transactions performed are accepted and none rejection because only a peer which possess the token can perform a transaction of its updates broadcasting and

reconcile other peers' data with its updates. As soon as peer "A" finishes to perform updates and reconciliations with peers "B, C and D", it releases the token and other peers like "B" or "C" or "D" can randomly take it, but according to the token request minimum date and time, and do the same, unless a privileged peer requests it.

Algorithm 2: P2P releasing the circulating token

```

Input: A set  $p$  of slave peers
Output: A slave peer receiving the token
begin releaseTokenFunction()
1:   select AvailableSlavePeers
2:   NonPrilegedPeerNber ← 0
3:   TokenRequestNber ← 0
4:   for ( $p \leftarrow 0$  to  $\text{NumberOfAvailableSlavePeers} - 1$ ) do
5:     if SlavePeer( $p$ ).ConnexionState = "True" then
6:       select SlavePeer( $p$ ).TokenRequest
7:       if (SlavePeer( $p$ ).TokenRequest = "True") then
8:         TokenRequestNber ++
9:         if (NonPrilegedPeerNber = 0) then
10:          if (SlavePeer( $p$ ).Privilege = "True") then
11:            if (SlavePeer( $p$ ).TokenRequestDateTime =  $\text{Min}(\text{DateTimeOfPrivilegedSlavePeers})$ ) then
12:              Set MasterPeer.TokenPossState ← "False"
13:              Set MasterPeer.TokenReleaseDateTime ← now()
14:              Set SlavePeer( $p$ ).TokenPossState ← "True"
15:              Set SlavePeer( $p$ ).TokenPrevention ← "False"
16:              Set SlavePeer( $p$ ).TokenRequest ← "False"
17:              return SlavePeer( $p$ ).TokenReceived (end for  $p$ )
18:            else
19:              Continue( $p$  ++ )
20:          endif
21:        else
22:          if ( $p = \text{NumberOfAvailableSlavePeer} - 1$ ) then
23:            if (SlavePeer( $p$ ).TokenRequestDateTime =  $\text{Min}(\text{DateTimeOfNonPrivilegedSlavePeers})$ ) then
24:              Set MasterPeer.TokenPossState ← "False"
25:              Set MasterPeer.TokenReleaseDateTime ← now()
26:              Set SlavePeer( $p$ ).TokenPossState ← "True"
27:              Set SlavePeer( $p$ ).TokenPrevention ← "False"
28:              Set SlavePeer( $p$ ).TokenRequest ← "False"
29:              return SlavePeer( $p$ ).TokenReceived (end for  $p$ )
30:            else
31:              NonPrilegedPeerNber ++
32:               $p \leftarrow -1$ 
33:            end if
34:          else if ( $p < \text{NumberOfAvailableSlavePeer} - 1$ ) then
35:            Continue( $p$  ++ )
36:          end if
37:        end if
38:      else
39:        if (SlavePeer( $p$ ).TokenRequestDateTime =  $\text{Min}(\text{DateTimeOfNonPrivilegedSlavePeers})$ ) then

```



```

40: Set MasterPeer.TokenPossState←"False"
41: Set MasterPeer.TokenReleaseDateTime←now()
42: Set SlavePeer(p).TokenPossState←"True"
43: Set SlavePeer(p).TokenPrevention←"False"
44: Set SlavePeer(p).TokenRequest←"False"
45: return SlavePeer(p).TokenReceived (end for p)
46: end if
47: end if
48: else
49: if (p = NumberOfAvailableSlavePeer - 1) then
50: Set MasterPeer.TokenPossState←"False"
51: Set MasterPeer.TokenReleaseDateTime←now()
52: end if
53: end if
54: end if
55: end for p
56: return a slave peer receiving the token
end releaseTokenFunction
    
```

ii. Replication Protocol and Algorithm

Assuming that the database is homogenous, full replicated and each Peer work under a Two-Phase-Locking (2PL) concurrency control technique. The model of the lazy replication over a decentralized Peer-to-Peer Architecture is presented as follows: let

$W(x)$ be a write transaction where x is a replicated data item at Peers A, B, C and D. The Fig. 7, here below depicts how transactions update different copies at all Peers and after commit the refresh transaction, wrapped in the Sync. Mediator-System, forward updates to all peers.

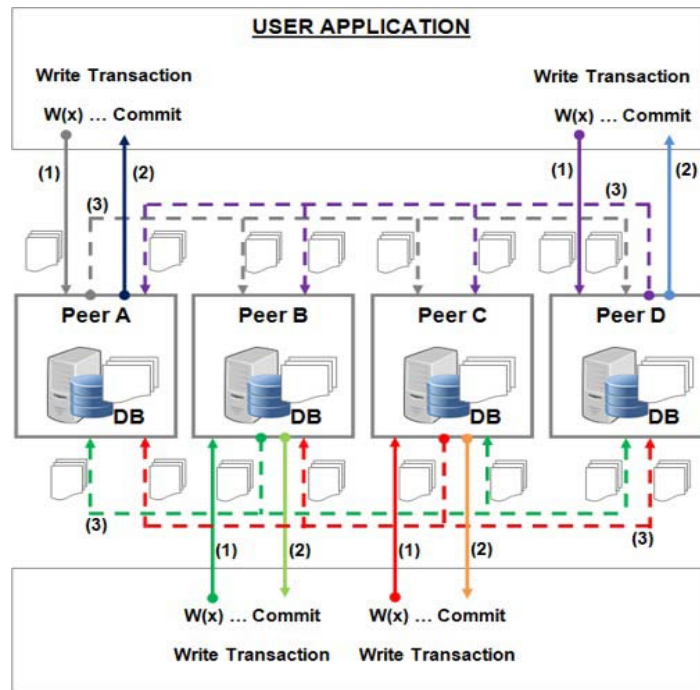


Fig. 7: Protocol of Lazy Decentralized P2P Data Replication.

Figure legend

1. Modifications are performed to all replicas by users;
2. The commitment of a transaction makes the modifications stable;
3. The modifications (Inserted, Updated and Deleted data) are independently transmitted to the other data copies or replicas.

According to the Fig. 7, arrows (1) and (2) deal with the user application i.e. for each local write transaction (1), the user application must receive the commitment (or abort) message (2). Changes carried by (3) are another set of transactions, wrapped in the Sync. Mediator-System, routed from each Master Peer to Slave Peers. The algorithm 3 here below establishes instructions in transactions of the Insert operator function.

Algorithm 3: P2P Replication Algorithm for Data Insertion

Input: Master peer inserted records

Output: Transaction Commitments or Abortions

```

begininsertFunction()
1:  begininsertMainTransaction
2:    selectall Available Slave Peers
3:    for(p ← 0 toNumberOfAvailableSlavePeers – 1)do
4:      begininsertSubTransactionPeer(p)
5:        selectall Audit Table Names in Mater Peer Database
6:        selectall Data Table Names in Slave Peer(p) Database
7:        for(ts ← 0 toNumberOfDataTableNamesInSlavePeer(p)Database – 1)do
8:          selectall Rows in Audit Table(ts) of Master Peer Database where AuditAction = 'Inserted'
          and AuditTimeStamp ≥ BeginningDateAndTime and
          AuditTimeStamp ≤ EndingDateAndTime
9:          for(rtm ← 0 toRowsInAuditTable(ts)OfMasterPeerDatabase – 1)do
10:           selectall Column Names in Data Table(ts) of Slave Peer(p) Database
11:           for(cts ← 0 to NumberOfColumnNamesInDataTable(ts)OfSlavePeer(p)Database – 1)do
12:             ColumnNames ← ColumnNames & ColumnNames[cts]
13:             Values ← Values & Row[rtm]Column[cts]
14:           end for cts
15:           insert in toDataTableNames(ts)InSlavePeer(p)Database (ColumnNames)values(Values)
16:         end for rtm
17:       end for ts
20:     endinsertSubTransaction(Commit or Abort)
21:   end for p
22: endinsertMainTransaction(Commit or Abort)
23: returnTransaction Commitments or Abortions
endinsertFunction
    
```

After records which have been inserted be which has the instructions in transactions of the replicated to slave peers, the algorithm 4 here below, update function, also runs in turn.

Algorithm 4: P2P Replication Algorithm for Data Update

Input: Master peer updated records

Output: Transaction Commitments or Abortions

```

beginupdateFunction()
1:  beginupdateMainTransaction
2:    selectall Available Slave Peers
3:    for(p ← 0 toNumberOfAvailableSlavePeers – 1)do
4:      beginupdateSubTransactionPeer(p)
5:        selectall Audit Table Names in Mater Peer Database
6:        selectall Data Table Names in Slave Peer(p) Database
7:        for(ts ← 0 toNumberOfDataTableNamesInSlavePeer(p)Database – 1)do
8:          selectall Rows in Audit Table(ts) of Master Peer Database where AuditAction = 'Updated'
          and AuditTimeStamp ≥ BeginningDateAndTime and
          AuditTimeStamp ≤ EndingDateAndTime
9:          for(rtm ← 0 toRowsInAuditTable(ts)OfMasterPeerDatabase -1)do
10:           selectall Column Names in Data Table(ts) of Slave Peer(p) Database
11:           for(cts ← 0 toNumberOfColumnNamesInDataTable(ts)OfSlavePeer(p)Database -1) do
12:             if(ColumnName(cts)InDataTable(ts)OfSlavePeer(p)Database =
              UpdatedColumnName)then
13:               updateDataTable(ts)OfSlavePeer(p)DatabasesetColumnName(cts)InDataTable(ts)Of
              SlavePeer(p)Database ← 'Row[rtm]Column[cts]'
              whereColumnName(0)InDataTable(ts)OfSlavePeer(p)Database =
              'Row[rtm]Column[0]'
14:             end if
15:           end for cts
16:         end for rtm
17:       end for ts
20:     endupdateSubTransaction(Commit or Abort)
21:   end for p
22: endupdateMainTransaction(Commit or Abort)
23: returnTransaction Commitments or Abortions
endupdateFunction
    
```

Finally, all deleted records are replicated by the algorithm 5 here below, by instructions in transactions of the delete function.

Algorithm 5: P2P Replication Algorithm for Data Delete

```

Input: Master peer deleted records
Output: Transaction Commitments or Abortions
begin deleteFunction()
1: begin deleteMainTransaction
2: select all Available Slave Peers
3: for (p ← 0 to NumberOfAvailableSlavePeers - 1) do
4:   begin deleteSubTransactionPeer(p)
5:     select all Audit Table Names in Mater Peer Database
6:     select all Data Table Names in Slave Peer(p) Database
7:     for (ts ← 0 to NumberOfDataTableNamesInSlavePeer(p)Database - 1) do
8:       select all Rows in Audit Table(ts) of Master Peer Database where AuditAction = 'Deleted'
       and AuditTimeStamp ≥ BeginningDateAndTime and
       AuditTimeStamp ≤ EndingDateAndTime
9:       for (rtm ← 0 to RowsInAuditTable(ts)OfMasterPeerDatabase - 1) do
10:        select all Column Names in Data Table(ts) of Slave Peer(p) Database
11:        delete from Data Table(ts) Of Slave Peer(p) Database where ColumnName(0) In Data Table(ts)
        Of Slave Peer(p) Database = 'Row[rtm]Column[0]'
12:      end for rtm
13:    end for ts
14:  end deleteSubTransaction(Commit or Abort)
15: end for p
16: end deleteMainTransaction(Commit or Abort)
17: return Transaction Commitments or Abortions
end deleteFunction
    
```

iii. *Reconciliation Protocol and Algorithm*

After a large data transmission, to overcome the problem of data inconsistency due to untimely interruptions of connectivity, network overload and other technical hazards, updates forwarded to each peer in the replication procedure must be reconciled.

The model of the Decentralized Peer-to-Peer Data Reconciliation is presented as follows: let R(x) be a read transaction where x is a replicated data item at Peers A, B, C and D. The Fig. 8, here below depicts how reconciliation is performed on different copies of all peers.

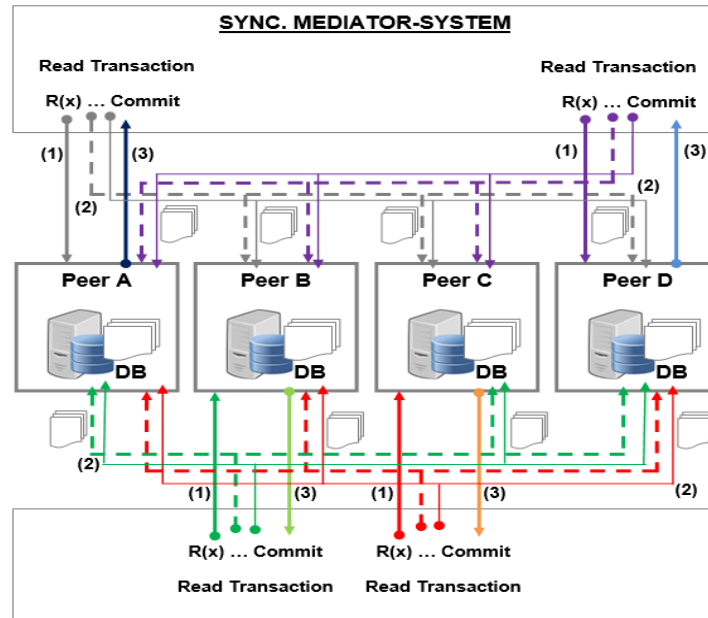


Fig. 8: Protocol of Decentralized P2P Data Reconciliation.

Figure legend

1. After the refresh transactions (Deleted, Updated and Inserted data) commit, then the reconciliation

procedure starts. A read transaction is performed to the Master peer to retrieve updates lastly broadcasting toward Slaves.

2. Processing:
 - Reading of updates independently forwarded to other replicas by the refresh transactions (in (2) dashed arrows);
 - Comparison with the Master data read in (1) undashed arrows;
 - Reconciliation is written to the Slave peers (in (2) undashed arrows).
3. The commitment (or cancelation) of a reconciliation transaction makes replicas consistent.

transactions to Slave peers. Dashed arrows (2) read as well data received by Slave peers. The comparison is done on the Master peer, by the Sync. - System in order to detect conflicts between data. Undashed arrows (2) update respective Slave peers fixing incoherency detected. So when the reconciliation transaction ends, the message, represented by arrows (3), is sending back to the Sync. Mediator-System and the process terminates. The whole reconciliation process is established by instructions in transactions of the reconciliation function of the algorithm 6 here below:

According to the Fig. 8, arrows (1) read data forwarded by the Master peer in the last update

Algorithm 6: P2P Algorithm for Data Reconciliation

Input: Master peer replicated (inserted, updated and deleted) records

Output: Transaction Commitments or Abortions

```

beginreconcileFunction()
1: beginreconcileMainTransaction
2: selectall Available Slave Peers
3: for(p ← 0 toNumberOfAvailableSlavePeers - 1)do
4:   beginreconcileSubTransactionPeer(p)
5:   selectall Audit Table Names in Mater Peer Database
6:   selectall Audit Table Names in Slave Peer(p) Database
7:   selectall Data Table Names in Slave Peer(p) Database
8:   for(ts ← 0 toNumberOfDataTableNamesInSlavePeer(p)Database - 1)do
9:     selectall Rows in Audit Table(ts) of Master Peer Database where TransactionType =
       'Local' and AuditTimeStamp ≥ BeginningDateAndTime and
       AuditTimeStamp ≤ EndingDateAndTime
10:    selectall Rows in Audit Table(ts) of Slave Peer(p) Database where TransactionType =
       'Remote'
11:    selectall Column Names in Data Table(ts) of Slave Peer(p) Database
12:    sortRows of Audit Table(ts) of Master Peer Database
13:    sortRows of Audit Table(ts) of Slave Peer(p) Database
14:    if(NumberOfRowsInAudit Table(ts)OfSlavePeer(p)Database - 1 < NumberOfRowsInAudit
       Table(ts)OfMasterPeerDatabase - 1)then
                                     //Reconcile missing records process start
15:      rts ← 0
16:      for(rtm ← 0 toNumberOfRowsInAuditTable(ts)OfMasterPeerDatabase - 1)do
17:        repeat
18:          if(rts ≤ NumberOfRowsInAudit Table(ts)OfSlavePeerDatabase - 1)then
19:            if(Row[rtm]Column[0]InAudit Table(ts)OfMasterPeerDatabase =
              Row[rts]Column[0]InAuditTable(ts)OfSlavePeer(p)Database)then
20:              Continue(rts++)
21:            end repeat
22:          else
23:            //Call function to insert missing records
              insertMissingRecordFunction(arguments)
24:            end if
25:          else
26:            //Call function to insert missing records
              insertMissingRecordFunction(arguments)
27:            end repeat
28:          end if
29:          until(Row[rtm]Column[0]InAudit Table(ts)OfMasterPeerDatabase =
              Row[rts]Column[0]InAuditTable(ts)OfSlavePeer(p)Database)
30:        end for rtm
31:      else if(NumberOfRowsInAudit Table(ts)OfSlavePeer(p)Database - 1
              > NumberOfRowsInAuditTable(ts)OfMasterPeerDatabase - 1)then
                                     //Reconcile duplicated records process start

```



```

32:      startSlaveLoop←0
33:      for(rtm ← 0 toNumberOfRowsInAudit Table(ts)OfMasterPeerDatabase – 1)do
34:          for(rts ← startSlaveLoop toNumberOfRowsInAudit Table(ts)OfSlavePeer(p)Database –
35:              1)do
36:              if(Row[rtm]Column[0]InAudit Table(ts)OfMasterPeerDatabase =
37:                  Row[rts]Column[0]InAuditTable(ts)OfSlavePeer(p)Database)then
38:                  if(rtm<NumberOfRowsInAudit Table(ts)OfMasterPeerDatabase – 1)then
39:                      startSlaveLoop←rts + 1
40:                  end for rts
41:              else
42:                  startSlaveLoop←rts + 1
43:              end if
44:          else
45:              //Call function to delete duplicated records
46:              deleteDuplicatedRecordFunction(arguments)
47:          end if
48:      end for rts
49:      end for rtm
50:      else
51:          //Reconcile incorrect, missing and incorrectly formatted values process start
52:          for (rtm = 0 to NumberOfRowsInAudit Table(ts)OfMasterPeerDatabase – 1)do
53:              for (cts = 0 to NumberOfColumnNamesInDataTable(ts)OfSlavePeer(p)Database –
54:                  1)do
55:                  if (Row[rtm]Column[cts]InAudit Table(ts)OfMasterPeerDatabase ≠
56:                      Row[rtm]Column[cts]InAuditTable(ts)OfSlavePeer(p)Database) then
57:                      //Call function to update Incorrect values, missing values, incorrectly formatted
58:                      values
59:                      updateIncorrectValuesFunction(arguments)
60:                  end if
61:              end for cts
62:          end for rtm
63:      end if
64:      end for ts
65:      end reconcileSubTransaction(Commit or Abort)
66:      end for p
67:      end reconcileMainTransaction(Commit or Abort)
68:      return Transaction Commitments or Abortions
69:  end reconcileFunction

```

To insert missing records, the algorithm 7 here is called.

Algorithm 7: Function to insert missing records

Input: DataTable(ts)OfSlavePeer(p)Database, cts, rtm

Output: Nothing

begininsertMissingRecordFunction(args)

```

1:  for(cts←0 to NumberOfColumnNamesInDataTable(ts)OfSlavePeer(p)Database – 1)do
2:      ColumnNames←ColumnNames&ColumnNames[cts]
3:      Values ←Values & Row[rtm]Column[cts]
4:  end for cts
5:  insert in toDataTableNames(ts)InSlavePeer(p) Database (ColumnNames) values (Values)

```

endinsertMissRecordFunction

To delete duplicated records, the algorithm 8 here is called.

Algorithm 8: Function to delete duplicated records

Input: DataTable(ts)OfSlavePeer(p)Database, rtm

Output: Nothing

begindeleteDuplicatedRecordFunction(args)

```

1:  deletefromDataTable(ts)OfSlavePeer(p)DatabasewhereColumnName(0)InDataTable(ts)OfSlave
Peer(p)Database = 'Row[rtm]Column[0]'

```

enddeleteDuplicatedRecordFunction

To update incorrect values, the algorithm 9 is called.

Algorithm 9: Function to update incorrect values, missing values, incorrectly formatted values

Input: DataTableName(ts)OfSlavePeer(p)Database, UpdatedColumnName, cts, rtm

Output: Nothing

beginupdateIncorrectValuesFunction(args)

- 1: **if**(ColumnName(cts)InDataTable(ts)OfSlave Peer(p)Database=UpdatedColumnName)**then**
- 2: **update**Data Table(ts)OfSlavePeer(p)Database**set**ColumnName(cts)InData Table(ts)OfSlave Peer(p)Database = 'Row[rtm]Column[cts]' **where**ColumnName(0)InData Table(ts)OfSlave Peer(p)Database = 'Row[rtm]Column[0]'
- 3: **end if**

endupdateIncorrectValuesFunction

After the implementation of these algorithms presented above, the main goal, according to which setting up a synchronizer-mediator for database replication being able to serialize the propagation of updates and their reconciliation in a replicated databases system over a decentralized P2P network is achieved. Although this goal be achieved, it is appropriate to know here that in computing the performance of an algorithm is assessed on the basis of its complexity [18]. The analysis of the theoretical complexity of this algorithm will be more concerned the time complexity than the space complexity especially as the data will be momentarily transit through the buffer to the destination. Nevertheless, the practical time that the execution of this algorithm takes will result from the simulation and will be calculated by the statistical method.

c) *Statistical method*

The performance of a system depends on a certain number of factors. We have to determine the practical time, that makes our system to execute successively transactions of updates propagation or replication (insert, update and delete) and transactions of data reconciliation. To analyse this performance, we will use the linear regression test with the random sampling technique. The linear regression test is a statistical analysis method that describes the variations of an endogenous variable associated with the variations of one or more exogenous variables i.e. the relation between an endogenous variable and one or more exogenous variables. In the case where the study concerns an endogenous variable with one exogenous variable, it's a simple regression and when it's an endogenous variable with more than one exogenous variable, it is a multiple regression [19].

This test will be used not only to determine the execution time based on a certain sample, but also to make a linear regression model that will be used to predict the execution time, which is the dependant factor or endogenous variable, based on other independent factors or exogenous variables, namely the number of records, the number of tables in the database and the number of Slave Peers. The following variables are selected:

- Y_i : is a random variable to explain "the time the synchronization algorithm takes to broadcast

updates and to reconcile replicas for an execution i ";

- X_{i1} : is an explanatory variable "the number of records the synchronization algorithm broadcast from a Master Peer to Slaves and reconcile between the Master and Slaves for an execution i ";
- X_{i2} : is an explanatory variable "the number of tables in the database whose records knew updates which need to be broadcasted and reconciled with Slaves for an execution i "
- X_{i3} : is an explanatory variable "the number of Slave Peers available to receive updates and to be reconciled for an execution i ".

Given a sample $(Y_i, X_{i1}, X_{i2}, X_{i3})$ whose $i \in [1, n]$, we will try to explain, as precisely as possible, the values taken by Y_i , the so-called endogenous variable from a series of explanatory variables X_{i1}, X_{i2}, X_{i3} . The model formulated in terms of random variables, takes the form: $Y_i = b_0 + b_1 X_{i1} + b_2 X_{i2} + b_3 X_{i3} + \epsilon_i$

Where:

- $i = 1, 2, \dots, n$
- b_0 is the constant term;
- b_1, b_2 and b_3 are coefficients of the regression to be estimated;
- ϵ_i : is the model error that expresses or summarizes the missing information in the linear explanation of the values of Y_i from X_{i1}, X_{i2}, X_{i3} (a random variable of zero mathematical expectation in this model i.e. problem of specifications, variables not taken into account, etc.).

The intensity of the relation between the independent variables and the dependent variable will be expressed by the correlation coefficient "R", which is the square root of the "R²", the determination coefficient of a linear regression model. The coefficient of correlation, will be used to determine the degree of linkage between the independent variables and the dependent variable while the coefficient of determination will help to measure the proportion of dependence of the dependent variable explained by independent variables. Thus, two sets of hypotheses are evoked as follow:

1. Test of the significance of each independent variable (X_{i1}, X_{i2}, X_{i3})



- ✓ Null hypothesis (H_0): X_{ik} is not a significant predictor of Y_i .
 - ✓ Alternative hypothesis (H_1): X_{ik} is a significant predictor of Y_i .
2. Test of significance of the overall regression model
- ✓ Null hypothesis (H_0): The overall regression model is not significant.
 - ✓ Alternative hypothesis (H_1): The overall regression model is significant.

These hypotheses will be verified at the end of the results which will be produced by a series of experiments perpetrated on a simulation environment which will be described in the following section.

IV. SIMULATION ENVIRONMENT

The implementation and experimentations will be run on a P2P network consisting of 4 traditional computers depicted in the Fig. 9, with the following properties: Processor: Intel Core i5, CPU 2.40GHz, Memory (RAM): 8.00GB and Storage: 1TB. The network will be based on a desktop switch of 100 Mbps of transmission speed, to establish a simple LAN using twisted - pair cables connection and RJ45 connectors. These computers will run under Windows 10 Professional 64 bits and SQL Server Management Studio 2012 Express as DDBMS, to manage databases and establish the connectivity between them.

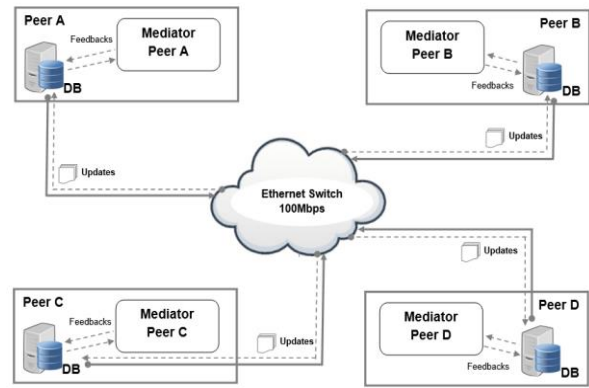


Fig. 9: Protocol of P2P Mediator-Synchronizer.

According to this Fig. 9 above, a node is composed by hardware and software as required previously. But in this same figure one can point out the presence of a “Mediator” for each peer. The mediator is nothing else than the synchronization system, "Sync. Mediator-System", a C# software which has been designed and in which it has been implemented algorithms, already described in the methodology, to lead to a windows application running under a graphical user interface, as presented in the Multiple-Document Interface (MDI) window here below in the Fig. 10.



Fig. 10: Sync. Mediator-System MDI window.

Thus this mediator must be installed on each node to manage the replication transactions and the reconciliation of replicas. For the execution to be effective, there are prerequisites to fulfil.

a) Prerequisites

When designing the global schema of the database, each table must have:

- The name such as “Data_tbTableName” and the first column as its primary-key to identify data and

to make the difference between records. The creation of primary keys by automatic incremental system provided by the DBMSs is disadvised, it is preferable to program an automatic primary key combined with the site number to avoid redundancy;

- Bear in mind that the database is homogeneous i.e. the data structure of the replicated database must be uniform on all peers.

b) Processing phases

Before the actual processing phase begins, under expected replication, "Sync. Mediator-System" provides two procedures that must be performed automatically in advance for each table, as showed in the window, Fig. 11:

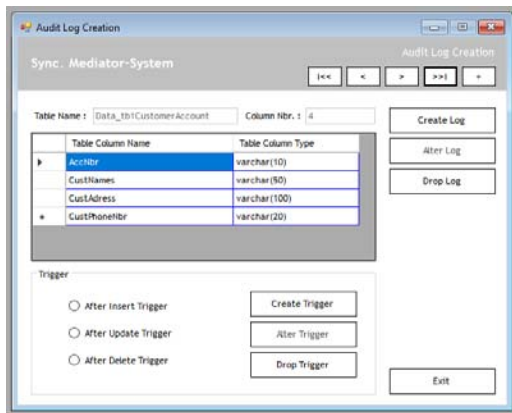


Fig. 11: Audit log creation window.

- To create one audit table named "Audit_tbTableName", to store changes captured by 3 triggers belonging to each table. Each audit table must have its next four last columns to store respectively the updated column name, the audit action, the audit timestamp and the last column to store the synchronization ID;
- To create three triggers to run after Insert, after Update and after Delete, to capture data changes and store them in the specific audit table.

The new circulating token algorithm has two phases:

i. Data replication

Update transaction serialization: All update transactions must be executed in serial order. Before initiating a refresh transaction, each peer must first receive a single token of a sequential series, to get the order in which the transaction will be executed. Once a token has been assigned to a peer *p*, this last becomes directly a Master so it performs update transactions to all connected Slave peers, as showed in the window, Fig. 12.

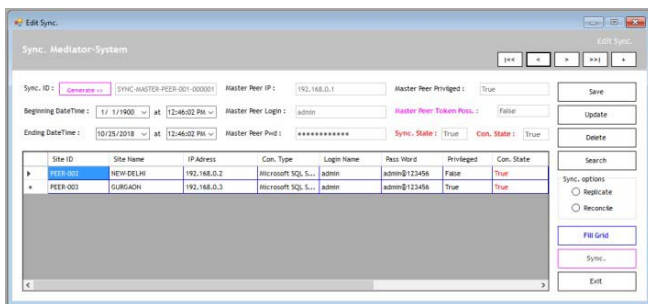


Fig. 12: Synchronization editor window.

Update transaction performing: When a Slave peer receives an executing transaction, it places it

according to its Master peer's token as well as its number (Sync_ID, in Fig. 5) and updates are performed to the Slave peer database. As soon as the transaction ends on each Slave peer, it sends an appropriate message to the Master peer to certify the transaction commitment. The peers connected during the initiation of the transaction and whose transaction has been aborting during transaction performing, due to any kind of issue to the site which host the peer, must be mentioned in the pending list in order to be updated later in a new procedure re-using the same Sync_ID. Then the main transaction, initiated on the Master peer, ends when it has been executed on all peers and give immediately the relay to the reconciliation procedure.

ii. Data reconciliation

Reconciliation transaction serialization: Reconciliation in turn will benefit from the serial order of their "Mather" update transactions. This phase must begin on the Master peer once the replication is complete. The reconciliation procedure must also initiate transactions to read updates received by Slave peers. These readings consist of a comparison between the data sent by the Master peer and the data received by the Slave peers. The comparison operation is performed according to data carrying the token of the same Master initiator of the replication transactions, as revealed in the window, Fig. 12. All errors like missing records, duplicate records, incorrect values, missing values, incorrectly formatted values are retained in order to be fixed.

Reconciliation transaction execution: This phase consists of fixing all retained errors so that missing records are inserted, duplicate records are deleted, missing values are added to their respective fields, incorrectly formatted values are replaced by correct values. Data reconciliation process can be however restarted if the first one done didn't put replicas in consistent state. So procedure can be repeated until all replicas become consistent, then the Master peer can release the token. In the case where the inconsistency persists among data, probably it can be caused by conflicts.

c) Conflicts avoidance rules

To avoid potential conflicts among data in the P2P replicated database environment, some rules must be respected:

- When using the database, it is inadvisable not to update the value of the primary key; instead, it is better to delete the entire record and re-insert it;
- When designing an application which communicate with the database, create procedures which cannot allow from a peer to update or to delete a record whose insertion was not performed on that same peer i.e. the modification of a data must be done only and

respectively on the peer that created it or inserted it.

After the configuration be performed as indicated in this section to simulate the replication process on a P2P network, the test and/or experiment sets yielded the results which are presented in the next section.

V. RESULT

This section is dedicated to testing this new synchronizer of databases, presenting the results and evaluating the performance of the newly proposed algorithm. To achieve this, it is necessary to analyse the performance in order to justify the effectiveness of the algorithm.

a) Performance analysis

Suppose that this algorithm has to broadcast updates emerging from the replicated database over 4 peers A, B, C, and D, local servers of a bank branches. Being fully replicated and homogeneous, the physical schema of this database consists of 3 tables, as presented in Fig. 13.

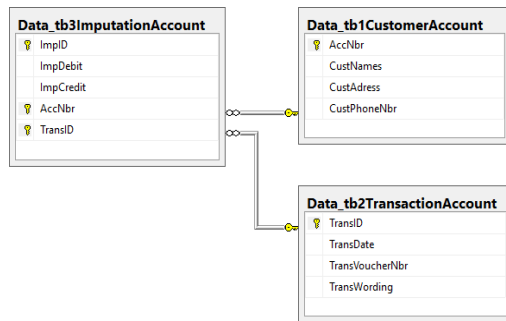


Fig. 13: Physical schema of a Banking Database.

So, for all cases, consider the sample of 12 executions, to operate randomly and based on the reality of the replicated data manipulation in the distributed environment of banking database. However, in all cases, insertions are greater than or equal to updates and deletes. But updates can be more or less than deletions.

After the replication transaction has completed, if there has been an overload or interruption of the network corrupting the replication transaction, then assume that the data that the

destination peers have received has experienced some inconsistencies with respect to those of the master peer. From the total replicated data (inserts, updates, and deletes), consider that 25% are missing records that require re-insertion, incorrect values, missing values, and incorrectly formatted values which need to be updated and duplicate records that require deletion, as typically data to be reconciled does not exceed 1/4 of that of replication [2], [22]. Thus, it resorts the data presented in the table 1 hereafter:

Table 1: Records Number Sample data

Nbr. Obs.	Number of rows to replicate	Number of rows to reconcile
1.	723	181
2.	900	225
3.	120	30
4.	2500	625
5.	1253	313
6.	80	20
7.	3000	750
8.	5000	1250
9.	450	113
10.	4860	1215
11.	600	150
12.	235	59
Mean	1643.42	410.92
Total	19721	4931

For analysing the effectiveness of our algorithm, the experimentation will be realized in four scenarios, namely:

1. Experimentation based one table stored on a master peer with two slave peers ;
2. Experimentation based two tables stored on a master peer with two slave peers ;
3. Experimentation based one table stored on a master peer with three slavepeers;
4. Experimentation based two tables stored on a master peer with three slavepeers.

To carry out the analysis of the performance, based on the prediction of the execution time according to the data of the sample presented in the Table 1 above, it results the execution times obtained after experimentation and presented successively in the tables and charts below:

Table 2: Result of the experimentation based one table stored on a master peer with two slavepeers

Sample numbering		Insert execution time (in Sec.)		Update execution time (in Sec.)		Delete execution time (in Sec.)	
Nbr. Obs.	Master Peer	Repliation	Reconci liation	Repliation	Reconci liation	Repliation	Reconci liation
1.	B	19	2	19	3	20	2
2.	A	24	2	24	4	24	2
3.	C	3	0	3	1	4	0
4.	C	67	5	68	12	69	8
5.	A	35	3	35	5	36	4

6.	A	3	0	2	0	3	0
7.	B	84	7	84	15	87	11
8.	B	144	11	148	25	152	18
9.	A	15	1	14	2	14	1
10.	C	161	12	173	26	189	18
11.	C	24	2	24	3	25	3
12.	B	10	0	9	1	10	1
Mean		49.08	3.75	50.25	8.08	52.75	5.67
Total		589	45	603	97	633	68

All basic factors remaining unchanged i.e. one table stored on a master peer with two slave peers, replication and reconciliation models are successively presented as follow : insert operator, Fig. 14(a) $y = 0.0302x - 0.5595 + \epsilon$ for data replication and Fig. 15(a) $y = 0.0093x - 0.0777 + \epsilon$ for data reconciliation,

update operator, Fig. 14(b) $y = 0.0318x - 2.0714 + \epsilon$ for data replication and Fig. 15(b) $y = 0.0208x - 0.4639 + \epsilon$ for data reconciliation and delete operator, Fig. 14(c) $y = 0.0336x - 2.528 + \epsilon$ for data replication and Fig. 15(c) $y = 0.0148x - 0.4124 + \epsilon$ for data reconciliation.

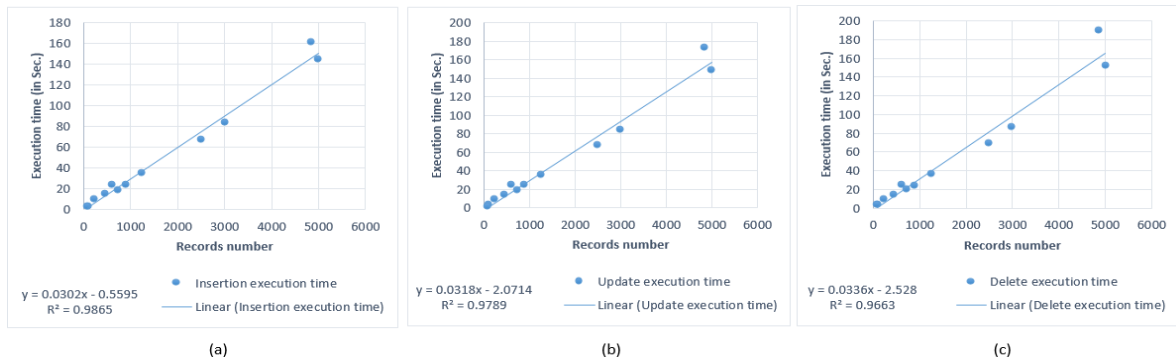


Fig. 14: Replication execution time: (a) Insertion, (b) Update and (c) Delete results from the experimentation based one table stored on a master peer with two slave peers.

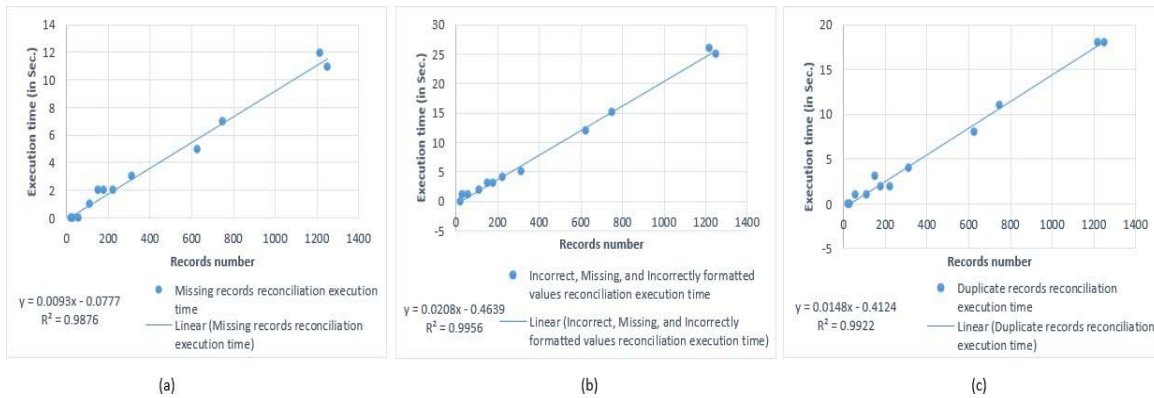


Fig. 15: Reconciliation execution time: (a) missing records, (b) incorrect values, missing values, and incorrectly formatted values and (c) duplicate records results from the experimentation based one table stored on a master peer with two slave peers.

Keeping unchanged basic factors, in 1 second (y) we predict that this algorithm can successively replicate and reconcile following number of records (x):

- For insert operator
 - ✓ In replication procedure (Fig. 14(a)): $1 = 0.0302x - 0.5595 \Rightarrow -0.0302x = -1.5595 \Rightarrow x = 51.63 \Rightarrow x \approx 52$ inserted records to be replicate in 1 second. So, as the coefficient of determination $R^2 = 0.9865$ then the insertion execution time depend on 98.65% of the number of records

and as the coefficient of correlation $R = \sqrt{R^2} \Rightarrow R = \sqrt{0.9865} \Rightarrow R = 0.9934$ then the degree of linkage between the insertion execution time and the number of records is 99.34%.

- ✓ In reconciliation procedure (Fig. 15(a)): $1 = 0.0093x - 0.0777 \Rightarrow -0.0093x = -1.0777 \Rightarrow x = 115.88 \Rightarrow x \approx 116$ missing records to be reconcile in 1 second. So, as the coefficient of determination $R^2 = 0.9876$ then the missing records reconciliation execution time depend on

98.76% of the number of records and as the coefficient of correlation $R = \sqrt{R^2} \Rightarrow R = \sqrt{0.9876} \Rightarrow R = 0.9938$ then the degree of relationship between the missing records reconciliation execution time and the number of records is 99.38%.

- For update operator
 - ✓ In replication procedure (Fig. 14(b)): $1 = 0.0318x - 2.0714 \Rightarrow -0.0318x = -3.0714 \Rightarrow x = 96.58 \Rightarrow x \approx 97$ updated records to be replicate in 1 second. Thus as the coefficient of determination $R^2 = 0.9789$ then the update execution time depend on 97.89% of the number of records and as the coefficient of correlation $R = \sqrt{R^2} \Rightarrow R = \sqrt{0.9789} \Rightarrow R = 0.9894$ then the degree of linkage between the update execution time and the number of records is 98.94%.
 - ✓ In reconciliation procedure (Fig. 15(b)): $1 = 0.0208x - 0.4639 \Rightarrow -0.0208x = -1.4639 \Rightarrow x = 70.37 \Rightarrow x \approx 70$ incorrect values, missing values, and incorrectly formatted values to be reconcile in 1 second. So, as the determination coefficient $R^2 = 0.9956$ then incorrect values, missing values, and incorrectly formatted values reconciliation execution time depend on 99.56% of the number of records and as the coefficient of correlation $R = \sqrt{R^2} \Rightarrow R = \sqrt{0.9956} \Rightarrow R = 0.9978$ then the linkage degree between the incorrect values, missing values, and incorrectly formatted values

reconciliation execution time and the number of records is 99.78%.

- For delete operator
 - ✓ In replication procedure (Fig. 14(c)) : $1 = 0.0336x - 2.528 \Rightarrow -0.0336x = -3.528 \Rightarrow x = 105$ deleted records to be replicate in 1 second. So, as the coefficient of determination $R^2 = 0.9663$ then the delete execution time depend on 96.63% of the number of records and as the coefficient of correlation $R = \sqrt{R^2} \Rightarrow R = \sqrt{0.9663} \Rightarrow R = 0.9830$ then the correlation between the insertion execution time and the number of records is 96.63%.
 - ✓ In reconciliation procedure (Fig. 15(c)): $1 = 0.0148x - 0.4124 \Rightarrow -0.0148x = -1.4124 \Rightarrow x = 95.43 \Rightarrow x \approx 95$ duplicated records to be reconcile in 1 second. Thus, as the coefficient of determination $R^2 = 0.9922$ then the duplicated records reconciliation execution time depend on 99.22% of the number of records and as the coefficient of correlation $R = \sqrt{R^2} \Rightarrow R = \sqrt{0.9922} \Rightarrow R = 0.9961$ then the degree of relation between the duplicated records reconciliation execution time and the number of records is 99.61%.

Table 2, Figs. 14 and 15 here above presented successively the result of the replication and reconciliation of records of one (1) table stored on a master peer toward two (2) slave peers, in terms of the execution time. Now, let's vary the number of tables and still observe the result.

Table 3: Result of the experimentation based two tables stored on a master peer with two slavepeers

Sample numbering		Insert execution time (in Sec.)		Update execution time (in Sec.)		Delete execution time (in Sec.)	
Nbr. Obs.	Master Peer	Repliation	Reconci liation	Repliation	Reconci liation	Repliation	Reconci liation
1.	B	12	2	12	3	11	2
2.	A	15	2	15	4	16	3
3.	C	2	0	2	1	3	1
4.	C	45	5	47	11	46	7
5.	A	24	3	24	5	25	4
6.	A	1	0	2	0	2	1
7.	B	61	7	61	9	63	10
8.	B	104	12	110	24	116	18
9.	A	12	1	12	2	12	1
10.	C	115	11	121	23	125	16
11.	C	16	1	16	2	16	1
12.	B	7	1	6	1	7	1
Mean		34.50	3.75	35.67	7.08	36.83	5.42
Total		414	45	428	85	442	65

By varying the factor number of tables, from one to two tables stored on a master peer, dividing the number of records equitably between two tables and maintaining unchanged the factor number of

slave peers in "two (2) peers", the replication and the reconciliation models are successively given as follow:

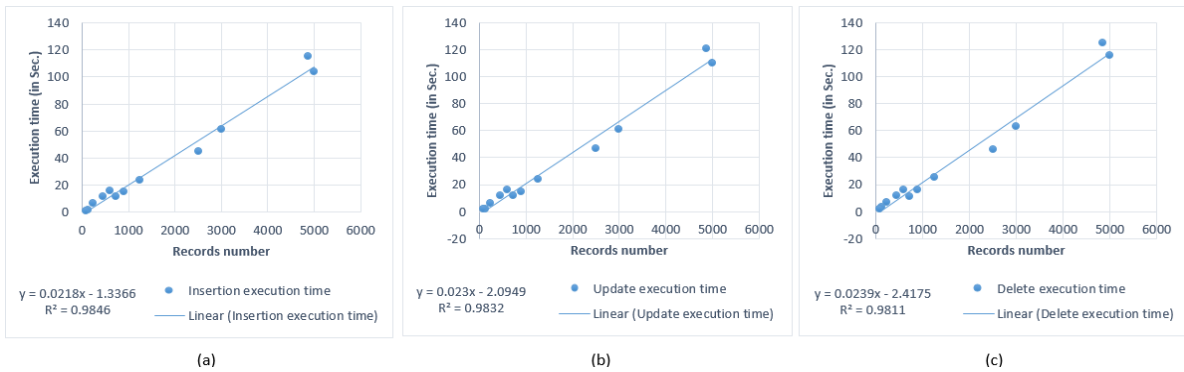


Fig. 16: Replication execution time: (a) Insertion, (b) Update and (c) Delete results from the experimentation based two tables stored on a master peer with two slave peers.

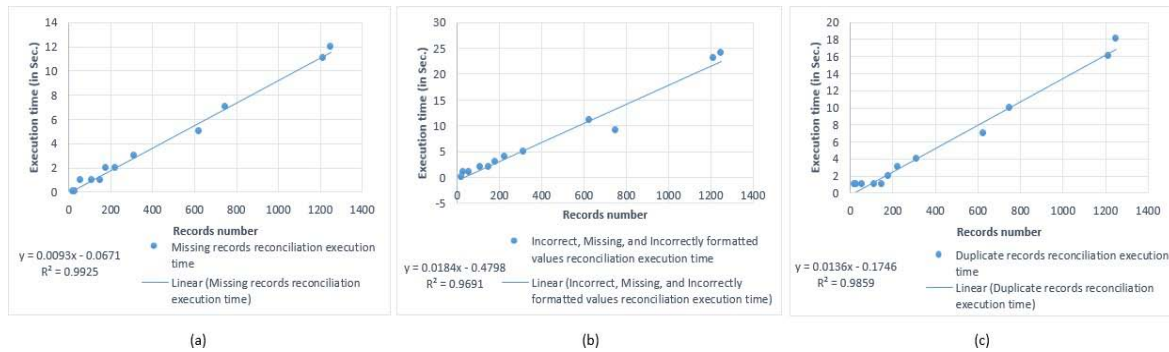


Fig. 17: Reconciliation execution time: (a) missing records, (b) incorrect values, missing values, and incorrectly formatted values and (c) duplicate records results from the experimentation based two tables stored on a master peer with two slave peers.

Insert operator, Fig. 16(a) $y = 0.0218x - 1.3366 + \epsilon$ for data replication and Fig. 17(a) $y = 0.0093x - 0.0671 + \epsilon$ for data reconciliation, update operator, Fig. 16(b) $y = 0.023x - 2.0949 + \epsilon$ for data replication and Fig. 17(b) $y = 0.0184x - 0.4798 + \epsilon$ for data reconciliation and delete operator, Fig. 16(c) $y = 0.0239x - 2.4175 + \epsilon$ for data replication and Fig. 17(c) $y = 0.0136x - 0.1746 + \epsilon$ for data reconciliation.

When we increase the number of tables from one to two, in 1 second, the prediction of the execution time (y), during which this algorithm can successively replicate and reconcile the number of records (x), is calculated from the following way:

- For insert operator
 - ✓ In replication procedure (Fig. 16(a)) : $1 = 0.021x - 1.3366 \Rightarrow -0.021x = -1.3366 \Rightarrow x = 111.26 \Rightarrow x \approx 111$ inserted records to be replicate in 1 second. So, as the coefficient of determination $R^2 = 0.9846$ then the dependence degree of insertion execution time compared to the number of records is 98.46% and as the coefficient of correlation $R = \sqrt{R^2} \Rightarrow R = \sqrt{0.9846} \Rightarrow R = 0.9923$ then the degree of linking between the insertion execution time and the number of records is 99.23%.

- ✓ In reconciliation procedure (Fig. 17(a)): $1 = 0.0093x - 0.0671 \Rightarrow -0.0093x = -1.0671 \Rightarrow x = 114.74 \Rightarrow x \approx 115$ missing records to be reconcile in 1 second. As the coefficient of determination $R^2 = 0.9925$ then the missing records reconciliation execution time depend on 99.25% of the number of records and as the coefficient of correlation $R = \sqrt{R^2} \Rightarrow R = \sqrt{0.9925} \Rightarrow R = 0.9963$ then the degree of relation between the missing records reconciliation execution time and the number of records is 99.63%.
- For update operator
 - ✓ In replication procedure (Fig. 16(b)): $1 = 0.023x - 2.0949 \Rightarrow -0.023x = -3.0949 \Rightarrow x = 134.56 \Rightarrow x \approx 135$ updated records to be replicate in 1 second. Thus as the coefficient of determination $R^2 = 0.9832$ then the update execution time depend on 98.32% of the number of records and as the coefficient of correlation $R = \sqrt{R^2} \Rightarrow R = \sqrt{0.9832} \Rightarrow R = 0.9916$ then the degree of the relation between the update execution time and the number of records is 99.16%.
 - ✓ In reconciliation procedure (Fig. 17(b)): $1 = 0.0184x - 0.4798 \Rightarrow -0.0184x = -1.4798 \Rightarrow x = 80.42 \Rightarrow x \approx 80$ incorrect values, missing

values, and incorrectly formatted values to be reconcile in 1 second. As the coefficient of determination $R^2 = 0.9691$ then incorrect values, missing values, and incorrectly formatted values reconciliation execution time depend on 96.91% of the number of records and as the coefficient of correlation $R = \sqrt{R^2} \Rightarrow R = \sqrt{0.9691} \Rightarrow R = 0.9844$ then the relation degree between the incorrect values, missing values, and incorrectly formatted values reconciliation execution time and the number of records is 98.44%.

- For delete operator
 - ✓ In replication procedure (Fig. 16(c)) : $1 = 0.0239x - 2.4175 \Rightarrow -0.0239x = -3.4175 \Rightarrow x = 142.99 \Rightarrow x \approx 143$ deleted records to be replicate in 1 second. So, as the coefficient of determination $R^2 = 0.9832$ then the delete execution time depend on 98.32% of the number of records and as the coefficient of correlation $R = \sqrt{R^2} \Rightarrow R = \sqrt{0.9832} \Rightarrow R = 0.9916$ then the correlation between the insertion execution time and the number of records is 99.16%.
 - ✓ In reconciliation procedure (Fig. 17(c)): $1 = 0.0136x - 0.1746 \Rightarrow -0.0136x = -1.1746 \Rightarrow x = 86.36 \Rightarrow x \approx 86$ duplicated records to be reconcile in 1 second. Thus, as the coefficient of determination $R^2 = 0.9859$ then the duplicated records reconciliation execution time

depend on 98.59% of the number of records and as the coefficient of correlation $R = \sqrt{R^2} \Rightarrow R = \sqrt{0.9859} \Rightarrow R = 0.9929$ then the degree of relation between the duplicated records reconciliation execution time and the number of records is 99.29%.

The experimentation of this algorithm on a topology consisting of two (2) slave peers proves that the variation of the number of tables containing data to replicate and reconcile in a P2P replication system has a significant impact only for the replication transaction as illustrated in Fig. 18. For all data modification operators, illustrated by graphs of Fig. 18(a), Fig. 18(b) and Fig. 18(c), successively, taken into account in the replication process, the execution time, when records originate from one (1) table, is greater than the execution time when the same number of records emerge from two (2) different tables while for reconciliation the impact is not too great.

Hence this variation has no significant effect on the execution time of data reconciliation because the number of records to reconcile from one (1) table and average of execution time, calculated in Table 2, are not far different from those to reconcile from two (2) tables and whose average of execution time is calculated in Table 3. This is why the curves of the graphs depicted in Fig. 18(d), Fig. 18(e) and Fig. 18(f) are almost similar.

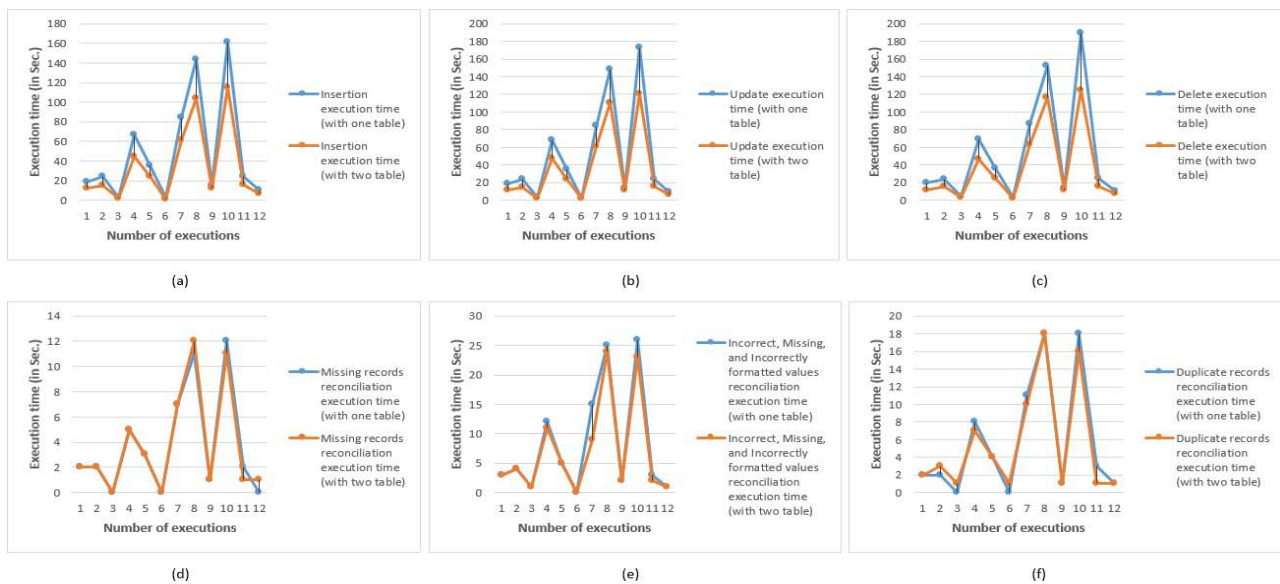


Fig. 18: Effectiveness of replication and reconciliation based one table stored on a master peer with two slave peers vs. two tables stored on a master peer with two slave peers.

So, partially we can conclude that this algorithm is efficient for the replication of databases because generally a database does not have one table i.e. data to replicate are scattered in several tables. As for reconciliation, since it takes place only

when it is necessary and mostly data to be reconciled do not exceed one quarter of that of replication, little importance should be attached to the computational time of this phenomenon.

This conclusion was obtained after varying the factor number of tables. However, by keeping unchanged all other factors, except the number of slave peers that vary from two (2) to three (3) peers,

using the same sample in Table 1, let us observe the execution time results from experimentation n, presented successively in the tables and graphs below:

Table 4: Result of the experimentation based one table stored on a master peer with three slave peers

Sample numbering		Insert execution time (in Sec.)		Update execution time (in Sec.)		Delete execution time (in Sec.)	
Nbr. Obs.	Master Peer	Repliation	Reconci liation	Repliation	Reconci liation	Repliation	Reconci liation
1.	B	22	2	23	3	23	2
2.	A	28	2	28	5	28	2
3.	C	3	0	3	1	5	0
4.	C	78	6	79	14	80	11
5.	D	41	3	41	6	42	5
6.	A	3	0	2	0	3	0
7.	B	97	8	97	17	101	15
8.	D	185	14	200	30	218	21
9.	A	17	1	16	2	16	1
10.	C	165	12	170	27	172	20
11.	D	28	2	28	3	29	3
12.	B	12	1	11	1	12	1
Mean		56.58	4.25	58.17	9.08	60.75	6.75
Total		679	51	698	109	729	81

Keeping the factor number of table unchanged, one table stored on a master peer with three slave peers, the replication and reconciliation models are successively presented as follow: insert operator, Fig. 19(a) $y = 0.0348x - 0.5762 + \epsilon$ for data replication and Fig. 20(a) $y = 0.0106x - 0.0883 + \epsilon$ for

data reconciliation, update operator, Fig. 19(b) $y = 0.0368x - 2.3047 + \epsilon$ for data replication and Fig. 20(b) $y = 0.0235x - 0.5576 + \epsilon$ for data reconciliation and delete operator, Fig. 19(c) $y = 0.0387x - 2.8053 + \epsilon$ for data replication and Fig. 20(c) $y = 0.0176x - 0.4611 + \epsilon$ for data reconciliation.

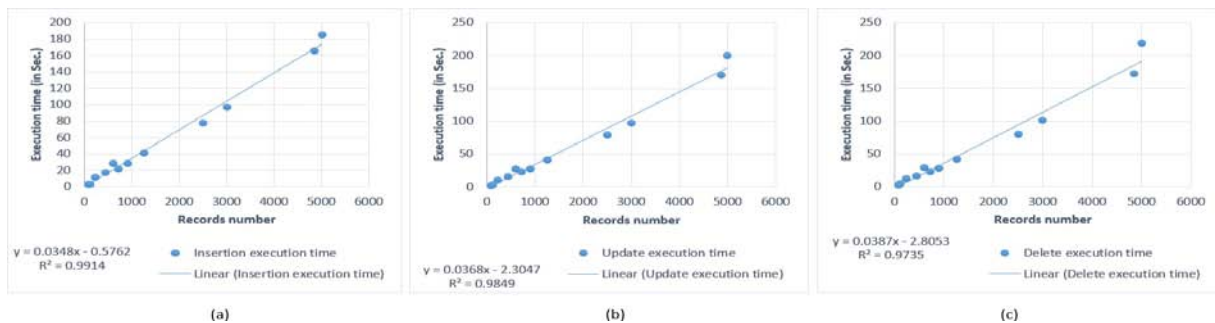


Fig. 19: Replication execution time: (a) Insertion, (b) Update and (c) Delete results from the experimentation based one table stored on a master peer with three slave peers.

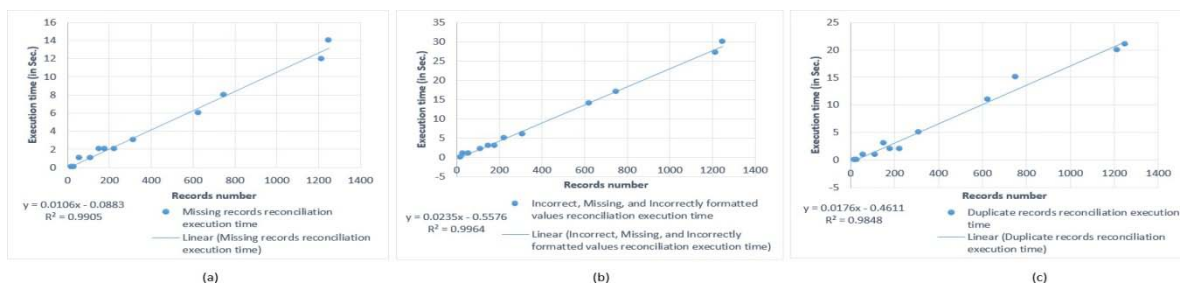


Fig. 20: Reconciliation execution time: (a) missing records, (b) incorrect values, missing values, and incorrectly formatted values and (c) duplicate records results from the experimentation based one table stored on a master peer with three slave peers.

In 1 second (y) we predict that this algorithm can successively replicate and reconcile following number of records (x):

- For insert operator
 - ✓ In replication procedure (Fig. 19(a)) : $1 = 0.0348x - 0.5762 \Rightarrow -0.0348x = -1.5762 \Rightarrow x = 45.29 \Rightarrow x \approx 45$ inserted records to be replicate in 1 second. So, as the coefficient of determination $R^2 = 0.9914$ then the insertion execution time depend on 99.14% of the number of records and as the coefficient of correlation $R = \sqrt{R^2} \Rightarrow R = \sqrt{0.9914} \Rightarrow R = 0.9957$ then the degree of linkage between the insertion execution time and the number of records is 99.57%.
 - ✓ In reconciliation procedure (Fig. 20(a)): $1 = 0.0106x - 0.0883 \Rightarrow -0.0106x = -1.0883 \Rightarrow x = 102.67 \Rightarrow x \approx 103$ missing records to be reconcile in 1 second. Thus, as the coefficient of determination $R^2 = 0.9905$ then the missing records reconciliation execution time depend on 99.05% of the number of records and as the coefficient of correlation $R = \sqrt{R^2} \Rightarrow R = \sqrt{0.9905} \Rightarrow R = 0.9952$ then the degree of relationship between the missing records reconciliation execution time and the number of records is 99.52%.
- For update operator
 - ✓ In replication procedure (Fig. 19(b)): $1 = 0.0386x - 2.3047 \Rightarrow -0.0638x = -3.3047 \Rightarrow x = 51.79 \Rightarrow x \approx 52$ updated records to be replicate in 1 second. Thus as the coefficient of determination $R^2 = 0.9849$ then the update execution time depend on 98.49% of the number of records and as the coefficient of correlation $R = \sqrt{R^2} \Rightarrow R = \sqrt{0.9849} \Rightarrow R = 0.9924$ then the degree of linkage between the update execution time and the number of records is 99.24%.
 - ✓ In reconciliation procedure (Fig. 20(b)): $1 = 0.0235x - 0.5576 \Rightarrow -0.0235x = -1.5576 \Rightarrow x = 66.28 \Rightarrow x \approx 66$ incorrect values, missing values, and incorrectly formatted values to be

reconcile in 1 second. Thus, as the determination coefficient $R^2 = 0.9964$ then incorrect values, missing values, and incorrectly formatted values reconciliation execution time depend on 99.64% of the number of records and as the correlation coefficient $R = \sqrt{R^2} \Rightarrow R = \sqrt{0.9964} \Rightarrow R = 0.9982$ then the relationship degree between the incorrect values, missing values, and incorrectly formatted values reconciliation execution time and the number of records is 99.82%.

- For delete operator
 - ✓ In replication procedure (Fig. 19(c)) : $1 = 0.0387x - 2.8053 \Rightarrow -0.0387x = -2.8053 \Rightarrow x = 98.32 \Rightarrow x \approx 98$ deleted records to be replicate in 1 second. So, as the coefficient of determination $R^2 = 0.9735$ then the delete execution time depend on 97.35% of the number of records and as the coefficient of correlation $R = \sqrt{R^2} \Rightarrow R = \sqrt{0.9735} \Rightarrow R = 0.9867$ then the correlation between the insertion execution time and the number of records is 98.67%.
 - ✓ In reconciliation procedure (Fig. 20(c)): $1 = 0.0176x - 0.4611 \Rightarrow -0.0176x = -1.4611 \Rightarrow x = 83.02 \Rightarrow x \approx 83$ duplicated records to be reconcile in 1 second. Thus as the determination coefficient $R^2 = 0.9848$ then the duplicated records reconciliation execution time depend on 98.48% of the number of records and as the coefficient of correlation $R = \sqrt{R^2} \Rightarrow R = \sqrt{0.9848} \Rightarrow R = 0.9924$ then the degree of relation between the duplicated records reconciliation execution time and the number of records is 99.24%.

Table 4, Figs. 19 and 20 here above presented successively the execution time results from the replication and reconciliation of records of one (1) table stored on a master peer toward three (3) slave peers. So, after the variation of the number of tables from one (1) and spreading proportionally records in two (2) tables, let us observe the result.

Table 5: Result of the experimentation based two tables stored on a master peer with three slavepeers

Sample numbering		Insert execution time (in Sec.)		Update execution time (in Sec.)		Delete execution time (in Sec.)	
Nbr. Obs.	Master Peer	Repliation	Reconci liation	Repliation	Reconci liation	Repliation	Reconci liation
1.	B	22	3	19	5	18	3
2.	A	26	3	28	6	28	5
3.	C	6	0	7	1	6	2
4.	C	90	8	93	18	92	15
5.	D	58	5	51	8	76	6
6.	A	6	0	6	0	6	0
7.	B	188	12	181	13	180	23

8.	D	242	28	266	38	288	32
9.	A	39	2	27	3	26	2
10.	C	291	24	250	37	272	31
11.	D	42	2	42	3	41	1
12.	B	17	2	17	2	17	2
Mean		85.58	7.42	82.25	11.17	87.50	10.17
Total		1027	89	987	134	1050	122

Varying the factor number of table stored on a master peer with three slave peers, the replication and reconciliation models are successively presented as follow: insert operator, Fig. 21(a) $y = 0.0539x - 2.9424 + \epsilon$ for data replication and Fig. 22(a) $y = 0.0206x - 1.0387 + \epsilon$ for data reconciliation, update operator, Fig. 21(b)

$y = 0.0527x - 4.3298 + \epsilon$ for data replication and Fig. 22(b) $y = 0.0293x - 0.8713 + \epsilon$ for data reconciliation and delete operator, Fig. 21(c) $y = 0.0566x - 5.5273 + \epsilon$ for data replication and Fig. 22(c) $y = 0.0266x - 0.7763 + \epsilon$ for data reconciliation.

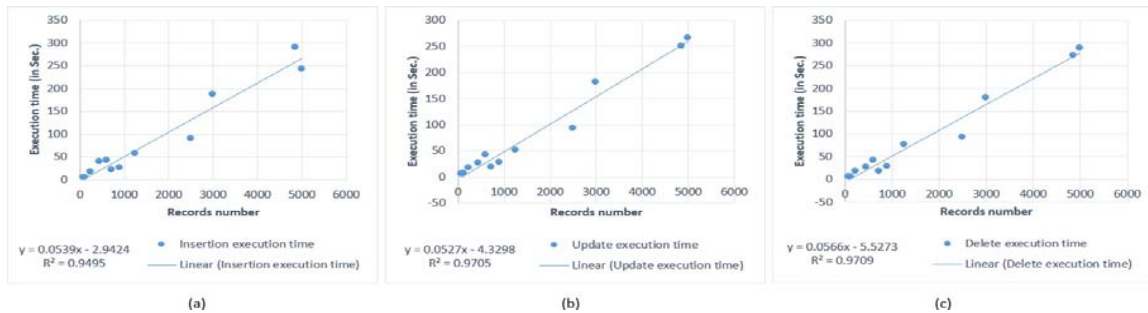


Fig. 21: Replication execution time: (a) Insertion, (b) Update and (c) Delete results from the experimentation based two tables stored on a master peer with three slave peers.

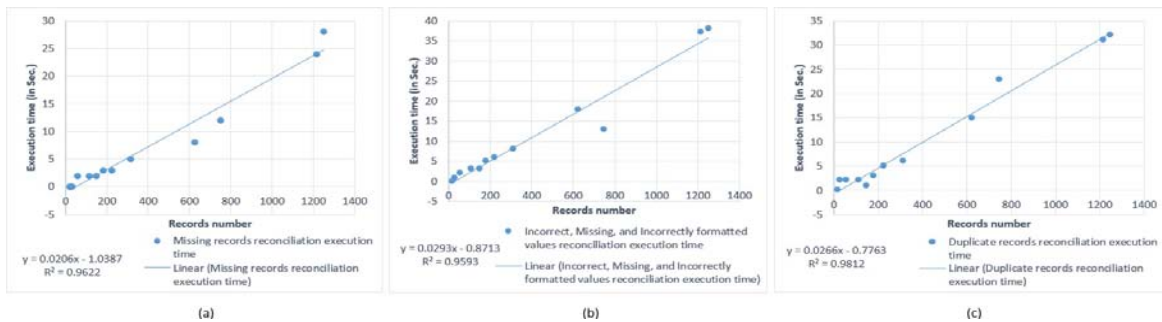


Fig. 22: Reconciliation execution time: (a) missing records, (b) incorrect values, missing values, and incorrectly formatted values and (c) duplicate records results from the experimentation based two tables stored on a master peer with three slave peers.

After increasing the number of tables from one to two, in 1 second, the prediction of the execution time (y), during which this algorithm can successively replicate and reconcile the number of records (x), is established as follows:

- For insert operator
 - ✓ In replication procedure (Fig. 21(a)) : $1 = 0.0539x - 2.9424 \Rightarrow -0.0539x = -2.9424 \Rightarrow x = 73.17 \Rightarrow x \approx 73$ inserted records to be replicate in 1 second. So, as the determination coefficient $R^2 = 0.9495$ then the dependence degree of insertion execution time compared to the number of records is 94.95% and as the coefficient of correlation $R = \sqrt{R^2} \Rightarrow R = \sqrt{0.9495} \Rightarrow R = 0.9744$ then the degree of

linking between the insertion execution time and the number of records is 97.44%.

- ✓ In reconciliation procedure (Fig. 22(a)): $1 = 0.0206x - 1.0387 \Rightarrow -0.0206x = -2.0387 \Rightarrow x = 98.88 \Rightarrow x \approx 99$ missing records to be reconcile in 1 second. As the coefficient of determination $R^2 = 0.9622$ then the missing records reconciliation execution time depend on 96.22% of the number of records and as the correlation coefficient $R = \sqrt{R^2} \Rightarrow R = \sqrt{0.9622} \Rightarrow R = 0.9809$ then the degree of relation between the missing records reconciliation execution time and the number of records is 98.09%.

- For update operator
 - ✓ In replication procedure (Fig. 21(b)): $1 = 0.0527x - 4.3298 \Rightarrow -0.0527x = -5.3298 \Rightarrow x = 101.13 \Rightarrow x \approx 101$ updated records to be replicate in 1 second. Thus as the coefficient of determination $R^2 = 0.9705$ then the update execution time depend on 97.05% of the number of records and as the coefficient of correlation $R = \sqrt{R^2} \Rightarrow R = \sqrt{0.9705} \Rightarrow R = 0.9851$ then the degree of the relation between the update execution time and the number of records is 98.51%.
 - ✓ In reconciliation procedure (Fig. 22(b)): $1 = 0.0293x - 0.8713 \Rightarrow -0.0293x = -1.8713 \Rightarrow x = 63.86 \Rightarrow x \approx 64$ incorrect values, missing values, and incorrectly formatted values to be reconcile in 1 second. As the determination coefficient $R^2 = 0.9593$ then incorrect values, missing values, and incorrectly formatted values reconciliation execution time depend on 95.93% of the number of records and as the coefficient of correlation $R = \sqrt{R^2} \Rightarrow R = \sqrt{0.9593} \Rightarrow R = 0.9794$ then the degree of relation between the incorrect values, missing values, and incorrectly formatted values reconciliation execution time and the number of records is 97.94%.
- For delete operator
 - ✓ In replication procedure (Fig. 21(c)) : $1 = 0.0566x - 5.5273 \Rightarrow -0.0566x = -6.5273 \Rightarrow$

$x = 115.32 \Rightarrow x \approx 115$ deleted records to be replicate in 1 second. So, as the coefficient of determination $R^2 = 0.9709$ then the delete execution time depend on 97.09% of the number of records and as the coefficient of correlation $R = \sqrt{R^2} \Rightarrow R = \sqrt{0.9709} \Rightarrow R = 0.9853$ then the correlation between the insertion execution time and the number of records is 98.53%.

- ✓ In reconciliation procedure (Fig. 22(c)): $1 = 0.0266x - 0.7763 \Rightarrow -0.0266x = -1.7763 \Rightarrow x = 66.78 \Rightarrow x \approx 67$ duplicated records to be reconcile in 1 second. As the coefficient of determination $R^2 = 0.9812$ then the duplicated records reconciliation execution time depend on 98.12% of the number of records and as the coefficient of correlation $R = \sqrt{R^2} \Rightarrow R = \sqrt{0.9812} \Rightarrow R = 0.9905$ then the degree of relation between the duplicated records reconciliation execution time and the number of records is 99.05%.

When running this algorithm on a topology consisting of three (3) slave peers, the experimentation result proves that the variation in the number of tables containing data to replicate and to reconcile in a P2P replication system has a significant impact on the execution time of replication and reconciliation transactions, as shown in Fig. 23.

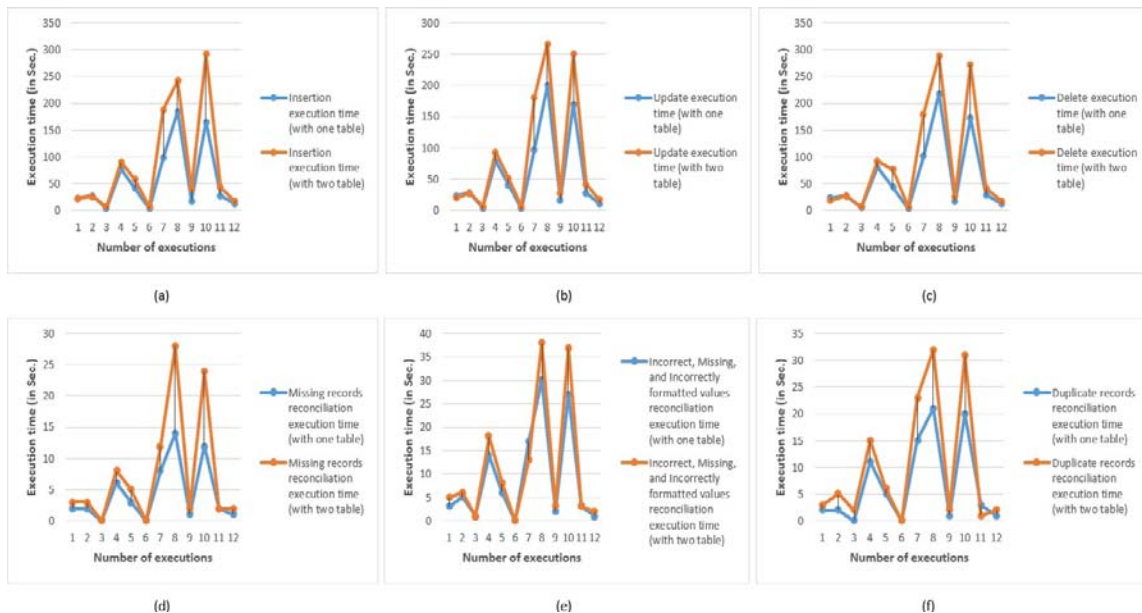


Fig. 23: Effectiveness of replication and reconciliation based one table stored on a master peer with three slave peers vs two tables stored on a master peer with three slavepeers.

However, this impact is explained only by the comparison of averages, in Table 4 and 5, which make successively curves, of execution time with two

tables, of graphs shown in Figs. 23(a), 23(b) and 23(c) for data replication and Figs. 23(d), 23(e) and 23(f) for data reconciliation to be high than those of

execution time with one table. But, in terms of predictive models, we found that, when the records come from one table, the execution time is greater than the execution time when the same number of records is split and comes from two different tables. This phenomenon is clarified by the successive resolution of the prediction equations of the replication and reconciliation models which proved that the number of records to replicate and reconcile

to 1 second, with two tables of origin is greater than those when there is only one table.

Thus, partially we can conclude that this algorithm is effective for the replication of databases, its performance increases with the increase of the tables for a certain number of records. So, since the data to replicate is usually scattered across multiple tables, we can count on its effectiveness.

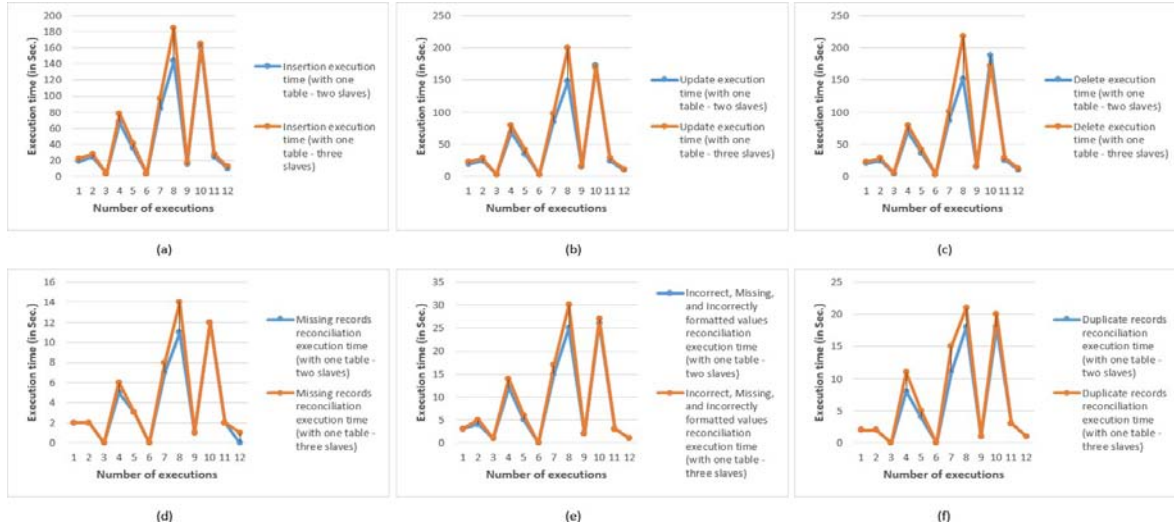


Fig. 24: Effectiveness of replication and reconciliation based one table stored on a master peer with two slave peers vs one table stored on a master peer with three slave peers.

The result we have achieved so far comes from the analysis of performance by varying the numbers of tables in which the data to be replicated and reconciled originate. Nevertheless, later on, we have to analyse the performance of this algorithm

starting from the variation of the slave peers. Thus, Fig. 24 and Fig. 25, show the effectiveness result when increasing the number of slave peers but the data to replicate and reconcile successively from a single table and two table.

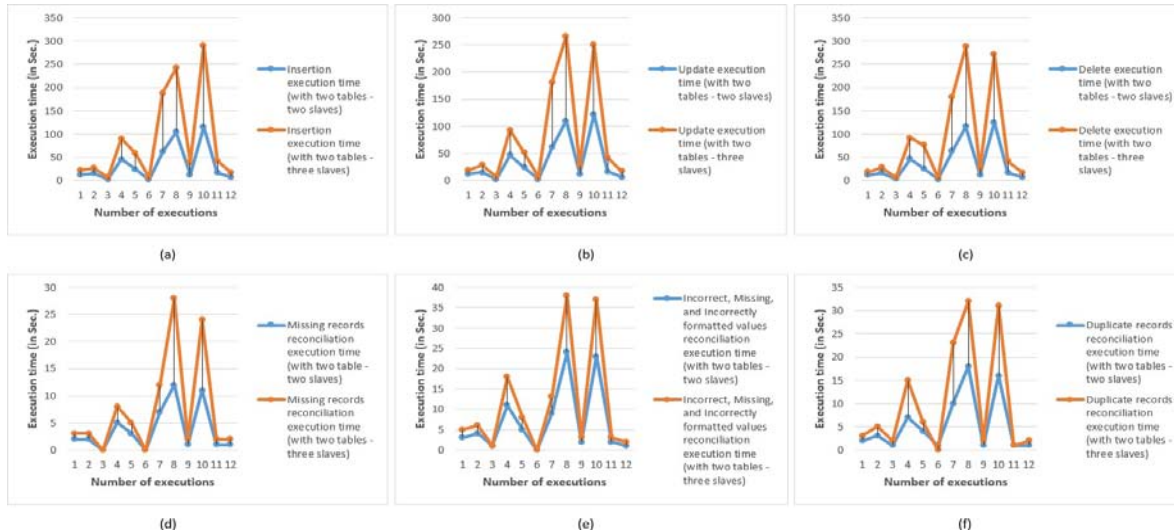


Fig. 25: Effectiveness of replication and reconciliation based two tables stored on a master peer with two slave peers vs two tables stored on a master peer with three slave peers.

After increasing the number of slave peers, the execution time of the replication transaction as well as the reconciliation of the data, successively

from a table, as illustrated in Fig. 24 and two tables, as shown in Fig. 25, knows a significant increase. This increase in execution time affects negatively the

performance of replication and reconciliation transactions. While the synchronization algorithm is only constituted by these two types of transactions, this loss of performance of said transactions involves the loss of performance of the whole synchronization algorithm.

This phenomenon can be explained in two ways:

- Firstly, by comparing the averages of the execution time which is explained by the graphs of Figs. 24 and 25, with illustrative curves of replication execution time (Figs 24 and 25 (a), (b) and (c)) and reconciliation (Figs 24 and 25 (d), (e) and (f)), with three slave peers are high than those with two slave peers;
- Secondly by comparing the predicted values, in this case the prediction of the number of records to replicate and reconcile to 1 second. After the

successive resolution of the prediction models equations for replication and data reconciliation, we found that the number of records to replicate and reconcile are declining after increasing a slave peer.

However, based on these observations from all the cases i.e. with the data to be replicated and reconciled from one or two tables, we can partially conclude that the increase of the number of slave peers on a Replicated Databases over a Decentralized P2P topology is causing the loss of performance of the synchronization algorithm.

b) Result summary

In view of what we have just achieved as a result, it is necessary to summarize and give a general conclusion. Thus, the Table 6 here below will first give a summary of the results.

Table 6: Results summary

Experimental scenarios	T ransaction	Operator	Model	R ²	R	Prediction (to 1 Sec.)
1. Experimentation based one table stored on a master peer with two slave peers	Replication	Insert	$y = 0.0302x - 0.5595 + \epsilon$	98.65%	99.34%	52 records
		Update	$y = 0.0318x - 2.0714 + \epsilon$	97.89%	98.94%	97 records
		Delete	$y = 0.0336x - 2.528 + \epsilon$	96.63%	96.63%	105 records
	Reconciliation	Insert	$y = 0.0093x - 0.0777 + \epsilon$	98.76%	99.38%	116 records
		Update	$y = 0.0208x - 0.4639 + \epsilon$	99.56%	99.78%	70 records
		Delete	$y = 0.0148x - 0.4124 + \epsilon$	99.22%	99.61%	95 records
2. Experimentation based two tables stored on a master peer with two slave peers	Replication	Insert	$y = 0.0210x - 1.3366 + \epsilon$	98.46%	99.23%	111 records
		Update	$y = 0.0230x - 2.0949 + \epsilon$	99.25%	99.63%	135 records
		Delete	$y = 0.0239x - 2.4175 + \epsilon$	98.32%	99.16%	143 records
	Reconciliation	Insert	$y = 0.0093x - 0.0671 + \epsilon$	96.91%	98.44%	115 records
		Update	$y = 0.0184x - 0.4798 + \epsilon$	98.32%	99.16%	80 records
		Delete	$y = 0.0136x - 0.1746 + \epsilon$	98.59%	99.29%	86 records
3. Experimentation based one table stored on a master peer with three slave peers	Replication	Insert	$y = 0.0348x - 0.5762 + \epsilon$	99.14%	99.57%	45 records
		Update	$y = 0.0368x - 2.3047 + \epsilon$	99.05%	99.52%	52 records
		Delete	$y = 0.0387x - 2.8053 + \epsilon$	98.49%	99.24%	98 records
	Reconciliation	Insert	$y = 0.0106x - 0.0883 + \epsilon$	99.64%	99.82%	103 records
		Update	$y = 0.0235x - 0.5576 + \epsilon$	97.35%	98.67%	66 records
		Delete	$y = 0.0176x - 0.4611 + \epsilon$	98.48%	99.24%	83 records
4. Experimentation based two tables stored on a master peer with three slave peers	Replication	Insert	$y = 0.0539x - 2.9424 + \epsilon$	94.95%	97.44%	73 records
		Update	$y = 0.0527x - 4.3298 + \epsilon$	96.22%	98.09%	101 records
		Delete	$y = 0.0566x - 5.5273 + \epsilon$	97.05%	98.51%	115 records
	Reconciliation	Insert	$y = 0.0206x - 1.0387 + \epsilon$	95.93%	97.94%	99 records
		Update	$y = 0.0293x - 0.8713 + \epsilon$	97.09%	98.53%	64 records
		Delete	$y = 0.0266x - 0.7763 + \epsilon$	98.12%	99.05%	67 records

Starting from the results presented above and summarizing in Table 6, our first group of hypotheses of the significance test of each independent variable gives the conclusion that each independent variable is a significant predictor of the dependent variable. In other words, the number of records in each table (x_1), the number of tables whose data has changed (x_2), the number of peers connected during the propagation of updates (x_3) and other factors (ϵ) like number of columns per table, data types columns, etc., each taken separately predict significantly the execution time (y) of the

replication transaction as well as that of reconciliation because almost all coefficient of determination (R^2) are greater than or equal to the confidence level of 95%. In all the cases the execution time depend on other factors beyond 95% and these factors correlate positively and tightly of the totality. This means that the changes made to one of these independent variables affect in 95% or more of the dependant variable and vice versa. Hence, we accept the alternative hypothesis (H_1) and thus reject the null hypothesis (H_0).

As for the second group of hypotheses, since for all experimental scenarios all independent variables (the number of records in each table (x_1), the number of tables whose data has changed (x_2), the number of peers connected during the propagation of updates (x_3) and other factors (ϵ) like number of columns per table, data types columns, etc.) are significant predictors of the dependent variable which is the replication and reconciliation transaction execution time (y), the overall model of the regression is significant, at the same thresholds significance derived from the combination of factors by the experimental scenarios summarized in the Table 6 above.

The experimental results show that our algorithms are performant since when to 1 second, a time elementary unity, it can replicate and reconcile a considerable number of records, like present the last column in the Table 6, for the present experimental environment. However, since the performance of a computer algorithm is due to its execution time, this is how we assert our main hypothesis that P2P replicated databases systems experience the weak performance, especially since the time of transmission of updates from a Master Peer toward Slave Peers dependent in more than 95% of the number of records, the number of tables whose data know changes, the number of peers connected during the propagation of updates and other factors.

Nevertheless, as we have just seen, when we take two by two experimental scenarios those can be noted successively I: 1 and 2, II: 3 and 4, III: 1 and 3 and finally IV: 2 and 4 of Table 6 above, I made a good performance, II also made a performance gain but not far from the average, III made a loss of performance and IV made a loss as well. Taking III and IV it emerges the variation of number of peers connected whereas from I and II emerge the variation of the tables. During the experiment, it was found that the variation of number of the tables did not lose the performance, contrariwise it improved it. Moreover, among the independent variables, the number of records and the number of tables being factors directly related to the database before even hinting at the data replication, it is clear that it is the growth of number of connected peers which is at the base of the considerable loss of the performance i.e. the increase of the execution time of a synchronization algorithm of distributed databases.

Thus, as a future work to be carried out, as part of improving the performance of this proposed algorithm, the thought will revolve around synchronization algorithm for replicated databases over a decentralized P2P architecture with super-nodes or super-peers [31], [32] belonging to peers clusters in order to reduce execution time of

transactions and to reach load balancing during data transmission [35].

VI. CONCLUSION

This article proposes a prototype of a synchronizer-mediator for lazy replicated databases over a decentralized P2P architecture in a Graphical User Interface. The motivation arises from the common problem of databases replication consisting to maintain consistent replicated databases over a decentralized P2P network.

However, two specific problems caught our attention: transactions broadcasting updates from different peers are performed concurrently on a destination peer replica, which always causes transactions conflicts and data conflicts. Moreover, during data migration, connectivity interruptions and network overload corrupt transactions so that destination peer databases can contract duplicated records, unsuitable data or missing records which make replicas inconsistent. Different methodologies have been used to solve these problems: the audit log technique to capture and store data changes in audit tables; the algorithmic method to design and analyse algorithms for transactions serialization, for data replication transactions and the replicas reconciliation transactions end finally the statistical method to analyse the performance of algorithms and to produce prediction models of the execution time.

The C # prototype software has been designed to implement algorithms and permit to execute the test in order to make out the effectiveness of each experimental scenarios. Afterwards it has been shown that the algorithm has a good performance because it can replicate and reconcile a considerable number of records to 1 second. Finally, the assumption according to which "The execution time of replication and reconciliation transactions totally depends on independent factors" has been affirmed.

ACKNOWLEDGEMENT

Firstly, we are grateful to the Grace of Almighty God. We would also like to thank the academic corps of the Butembo (D. R. Congo) Institute of Building and Public Works for their encouragement and follow-up of our investigations. On finish, we thank the Research Technology and Development Centre (RTDC) of Sharda University, for its facilities to realize this work.

REFERENCES RÉFÉRENCES REFERENCIAS

1. Kituta, K., Agarwal, R., Kaushik, B.: Synchronous and Asynchronous Replication. In: International Conference on Machine Learning and Computational Intelligence-2017, International

- Journal of Scientific Research in Computer Science, Engineering and Information Technology, Vol. 2, No. 7, pp. 347-354, (2017).
2. Özsu, M. T., Valduriez, P.: Principles of Distributed Database Systems (3rd ed.). In: Springer Science & Business + Media, New York, USA (2011).
3. Magdalena, N., I.: The Replication Technology in E-learning Systems. In: Procedia - Social and Behavioral Sciences, Publisher: Elsevier, Vol. 28, pp. 231 – 235, (2011).
4. Wiesmann, M., et al.: Understanding Replication in Databases and Distributed Systems. In: IEEE 20th International Conference on Distributed Computing Systems, (2002).
5. Gudakesa, R., Sukarsa, M., Sasmita, G.: Two - ways database synchronization in homogeneous DBMS using audit log approach. In: Journal of Theoretical and Applied Information Technology, Vol. 65, pp. 854-859, (2014).
6. Pandey, S., Shanker, U.: IDRC: A Distributed Real-Time Commit Protocol, In: 6th International Conference on Smart Computing and Communications ICSCC 2017, Procedia Computer Science, Publisher: Elsevier, Vol. 125, pp. 290–296, (2017).
7. Kudo, T., et al.: An implementation of concurrency control between batch update and online entries, In: 18th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems - KES2014, Procedia Computer Science, Publisher: Elsevier, Vol. 35, pp. 1625–1634, (2014).
8. Diallo, O., Rodrigues, Joel, J., Sene, M., Lloret, J.: Distributed Database Management Techniques for Wireless Sensor Networks, In: IEEE Transactions on Parallel and Distributed Systems, Vol. 26, No. 2, 604– 620, (2015).
9. Silberschatz, A., Korth, H F., Sudarshan, S.: Database system concepts. In: McGraw-Hill, New York, (1997).
10. Vu, Q., Lupu, M., Ooi, C.: Peer-to-Peer Computing - Principles and Applications, In: Springer, ISBN: 978-3-642-03513-5, (2010).
11. Filip, I., Vasar, C., Robu, R.: Considerations about an Oracle Database Multi-Master Replication. In: IEEE 5th International Symposium on Applied Computational Intelligence and Informatics, (2009).
12. Mansouri Y., Buyya R.: Dynamic replication and migration of data objects with hot -spot and cold-spot statuses across storage data centers, In: Journal of Parallel and Distributed Computing, Publisher: Elsevier, Vol. 126, pp. 121-133, (2018).
13. Sebastian, M.: Fundamentals of SQL Server 2012 Replication. In: Simple Talk Publishing, New York, United States of America, (2013).
14. Kirtikumar, D.: Oracle Streams 11g Data Replication. In: McGraw-Hill, New York, United States of America, (2011).
15. George, A., Balakrishnan, C.: An optimized strategy for replication in peer-to-peer distributed databases. In: IEEE International Conference on Computational Intelligence and Computing Research, (2012).
16. Zhang, T.: A Novel Replication Model with Enhanced Data Availability in P2P Platforms. In: International Journal of Grid and Distributed Computing, Vol. 9, No. 4, pp.151-160, (2016).
17. Ting, Z., Yu, W.: Database Replication Technology having high Consistency Requirements. In: IEEE Third International Conference on Information Science and Technology, (2013).
18. Cormen, T., H., et al.: Introduction to Algorithms (4th ed.). In: The MIT Press, London, England, (2012).
19. Kothari, C., R., Garg, G.: Research methodology methods and techniques (3rd ed.). In-House, Ed., New-Dheli, India: M.P Printers, 2014.
20. Microsoft Corporation web site (2018). <https://docs.microsoft.com/en-us/sql/relational-databases/replication/transactional/peer-to-peer-conflict-detection-in-peer-to-peer-replication?view=sql-server-2017>
21. Oracle Corporation web site (2018).
22. <https://docs.oracle.com/database/121/REPLN/replconflicts.htm#REPLN005>
23. Experian Ltd web site (2018). <https://www.edq.com/uk/glossary/data-reconciliation/>
24. Shahin, K., Pedram, G., Khuzaima, D.: Dynamic Data Allocation with Replication in Distributed Systems. 30th IEEE International Performance Computing and Communications Conference, (2011).
25. Oracle Corporation web site (2018).
26. https://docs.oracle.com/cd/E80148_01/html/Upgrade_Tool_Kit/UTKRCN07.htm
27. Oracle Corporation web site (2018). <https://dev.mysql.com/doc/mysql-utilities/1.5/en/mysqlldbcompare.html>
28. Jonathan, H., MySQL_Diff: Database Schema Difference Reconciliation (2018). <https://www.perpetual-beta.org/weblog/mysql-diff.html>
29. DB Convert Company web site (2018). <https://dbconvert.com/>
30. Pragmatic Works Inc. web site (2018).
31. http://pragmaticworks.com/Products_Old/LegitEst/Feature/Reconcile-Your-Production-Data
32. ApexSQL LLC web site (2018).

33. <https://solutioncenter.apexsql.com/fr/synchroniser-les-bases-de-donnees-sql-server-dans-differentes-sources-distantes/>
34. Kituta, K., Kant, S., Agarwal, R.: Analysis of database replication protocols. In: Special Issue ICRMR-2018, International Journal of Latest Trends in Engineering and Technology, pp. 075-083, (2018).
35. Spaho, E. et al.: P2P Data Replication: Techniques and Applications. In: Xhafa F., Barolli L., Barolli A., Papajorgji P. (eds) Modeling and Processing for Next -Generation Big-Data Technologies. Modeling and Optimization in Science and Technologies, Publisher: Springer Vol. 4, pp 145-166, (2015).
36. X. Fatos, et al.: Data Replication in P2P Collaborative Systems. In: IEEE Seventh International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, (2012).
37. Kituta, K., Kant, S., Agarwal, R.: A systematic review on distributed databases systems and their techniques. In: Journal of Theoretical and Applied Information Technology, Vol. 96, No. 1, pp. 236-266, (2019).
38. Souri, A., Pashazadeh, S., Navin, A., H.: Consistency of data replication protocols in database systems: A review. International Journal on Information Theory (IJIT), Vol. 3, No. 4, pp. 19-32, (2014).
39. M., Santana, Enrique, J., Francesc, D.: Evaluation of database replication techniques for cloud systems. In: Computing and Informatics, Vol. 34, pp. 973-995, (2015).



This page is intentionally left blank





Study and Performance Analysis of Different Techniques for Computing Data Cubes

By Aiasha Siddika

Stamford University

Abstract- Data is an integrated form of observable and recordable facts in operational or transactional systems in the data warehouse. Usually, data warehouse stores aggregated and historical data in multi-dimensional schemas. Data only have value to end-users when it is formulated and represented as information. And Information is a composed collection of facts for decision making. Cube computation is the most efficient way for answering this decision making queries and retrieve information from data. Online Analytical Process (OLAP) used in this purpose of the cube computation. There are two types of OLAP: Relational Online Analytical Processing (ROLAP) and Multidimensional Online Analytical Processing (MOLAP). This research worked on ROLAP and MOLAP and then compare both methods to find out the computation times by the data volume. Generally, a large data warehouse produces an extensive output, and it takes a larger space with a huge amount of empty data cells. To solve this problem, data compression is inevitable. Therefore, Compressed Row Storage (CRS) is applied to reduce empty cell overhead.

Keywords: data cube, compressed row storage, MOLAP, ROLAP.

GJCST-C Classification: H.2.7



Strictly as per the compliance and regulations of:



Study and Performance Analysis of Different Techniques for Computing Data Cubes

Aiasha Siddika

Abstract- Data is an integrated form of observable and recordable facts in operational or transactional systems in the data warehouse. Usually, data warehouse stores aggregated and historical data in multi-dimensional schemas. Data only have value to end-users when it is formulated and represented as information. And Information is a composed collection of facts for decision making. Cube computation is the most efficient way for answering this decision making queries and retrieve information from data. Online Analytical Process (OLAP) used in this purpose of the cube computation. There are two types of OLAP: Relational Online Analytical Processing (ROLAP) and Multidimensional Online Analytical Processing (MOLAP). This research worked on ROLAP and MOLAP and then compare both methods to find out the computation times by the data volume. Generally, a large data warehouse produces an extensive output, and it takes a larger space with a huge amount of empty data cells. To solve this problem, data compression is inevitable. Therefore, Compressed Row Storage (CRS) is applied to reduce empty cell overhead.

Keywords: data cube, compressed row storage, MOLAP, ROLAP.

I. INTRODUCTION

Online Analytical Processing (OLAP) is a database acceleration techniques used for deductive analysis. The main objective of OLAP is to have constant-time or near constant time answers for many typical queries. The widespread use of Online Analytical Processing (OLAP) is to resolve multi-dimensional analytical (MDA) queries expeditiously. Business intelligence, report writing, and data mining are also some immense categories of OLAP areas along with some applications like business reporting, marketing analogy, management reporting, business process management, budgeting and forecasting, and financial reporting with other similar areas. OLAP has been created with a slight alteration from the conventional database term Online Transaction Processing (OLTP) [1].

OLAP tools have been adopted extensively by users from various perspectives for the evaluation of multidimensional data. Consolidation (roll-up), drill-down, and slicing-dicing are three basic analytical operations of OLAP. Consolidation associates with data aggregation and stores it in one or more dimensions. In contradiction, the drill-down involves analyzing thorough details of data. Capturing a specific set of data from

OLAP cube called Slicing and create different viewpoints labeled as Dicing. Usually, there are two primary variations of OLAP: Relational Online Analytical Processing (ROLAP) and Multidimensional Online Analytical Processing (MOLAP). ROLAP works straight with relational databases where the dimension tables stored as relational tables, and new tables are created to hold the aggregated information by the tools. Data manipulation on this method provides an aspect of slicing and dicing functionality of traditional OLAP's. ROLAP tools feature the ability to answer all queries because the methodology does not limited to the contents of a cube. It can also drill down to the lowest dimension of the database. Differently, Multidimensional Online Analytical Processing (MOLAP) uses optimized multi-dimensional array storage to store data, in alternate of the relational database. It requires the pre-computation and storage information in the cube (the data cube) - the operation known as processing. And the data cube comprises all the possible answers to a given range of queries. MOLAP provides quick response time and the tools have a very fast capacity to write back data into the data set [2].

While designing an OLAP solution, the type of OLAP storage is one of the crucial decisions. Both ROLAP and MOLAP have their advantages and disadvantages. ROLAP can handle large amounts of data, and it can also leverage functionalities inherent in the relational database, but its performance can be slow or limited by SQL functionalities. On the contrary, in MOLAP, because of all calculations performed at the cube computation, it is not possible to include a large amount of data in the data cube itself, and it requires additional investment. Also, MOLAP cubes are created for fast data retrieval and optimal for slicing and dicing operations. It can perform complex calculations that have been pre-generated when the data cube created. Hence, complex calculations are not only doable, but they return quickly [3]. The implementation of both techniques may give a better competitive result. Data cube computation often produces excessive outputs with empty memory cells thus, make wastage of memory storage. To solve this problem, I will cover an efficient computation method called Compressed Row Storage (CRS).

In this paper, I have implemented ROLAP on manipulating the data stored in the relational database to give the appearance of traditional OLAP's slicing and

Author: Department of Computer Science and Engineering, Stamford University Bangladesh. e-mail: aiasha.arshi@gmail.com

licing functionality, MOLAP on a Multidimensional array and CRS on a multidimensional cube to eliminate unnecessary elements. And finally, Compare these three methods of data cube computation according to their execution time. The next portion of this work is the background study discussion; part 3 explains the methodology and implementation phase; part 4 shows result analysis.

II. LITERATURE REVIEW

As described in [4], Cubes in a data warehouse stored in three different modes. Relational Online Analytical Processing mode or ROLAP is a relational storage model, while a Multidimensional Online Analytical processing mode is called MOLAP. There's another OLAP named Hybrid Online Analytical Processing mode or HOLAP, where dimensions stored in a combination of the two approaches. One advantage of ROLAP over the other styles of OLAP tools is that it is considered more scalable in handling massive amounts

of data. It sits on top of relational databases, therefore, enabling it to leverage several functionalities that a relational database is capable of. Managing both numeric and textual data is another efficiency of it. Bassiouni M. A. [5] states that ROLAP applications display a slower performance as compared to another style of OLAP tools since, often, calculations performed inside the server. Another demerit of a ROLAP tool is that as it is dependent on the use of SQL for data manipulation, it may not be ideal for the performance of some calculations that are not easily translatable into an SQL query. However, ROLAP technology tends to have greater scalability than MOLAP technology. The DSS server of Micro strategy, for example, adopts the ROLAP approaches [6].

The implementation phase of ROLAP uses aggregate functions and GROUP BY operator to return a single value combined with the ROLL UP and get the total value which is similar to the CUBE operator. It is as akin to the following figure 2.1 [7].

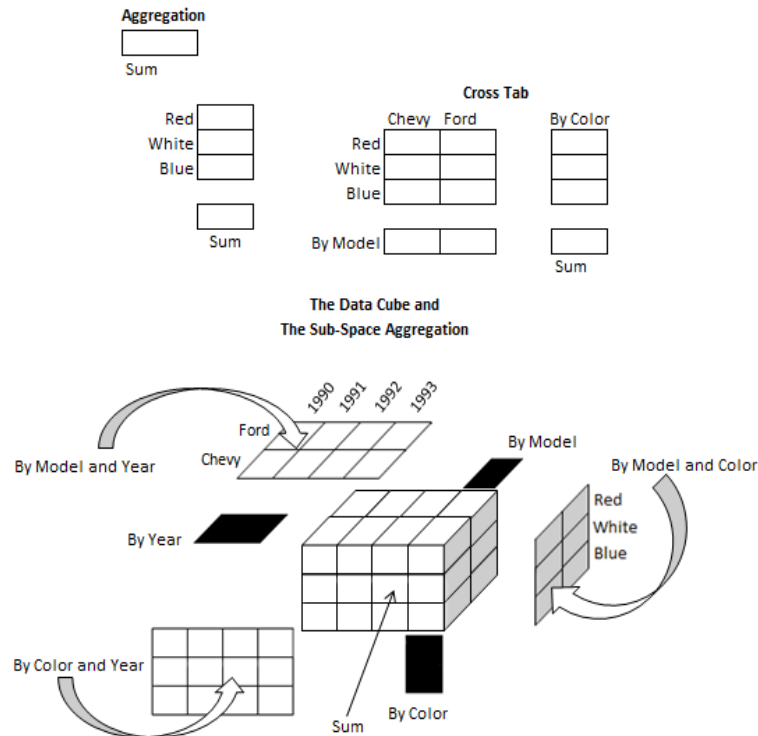


Figure 2.1: Group By relational operator with aggregate function SUM ()

MOLAP is the traditional mode of OLAP analysis that provides excellent query performance, and the cubes built for fast data retrieval. Since all calculations have been pre-built in data cube creation, the cube cannot be derived from a large volume of data, and it also requires excessive additional investment as cube technology is proprietary and the knowledge base may not exist in the organization as described in [8]. It supports the multidimensional views of data through array-based multidimensional storage engines. They map multidimensional views directly to the data cube

array structures. The advantage of using a data cube is that it allows fast indexing to precomputed summarized data. Notice that with multidimensional data stores, the storage utilization may be low if the data set is sparse. In such cases, exploring sparse matrix compression techniques are a must. Many MOLAP servers adopt a two-level storage representation to handle dense and sparse data sets: dense sub-cubes are identified and stored as array structures, whereas sparse sub-cubes employ compression technology for efficient storage utilization [9].

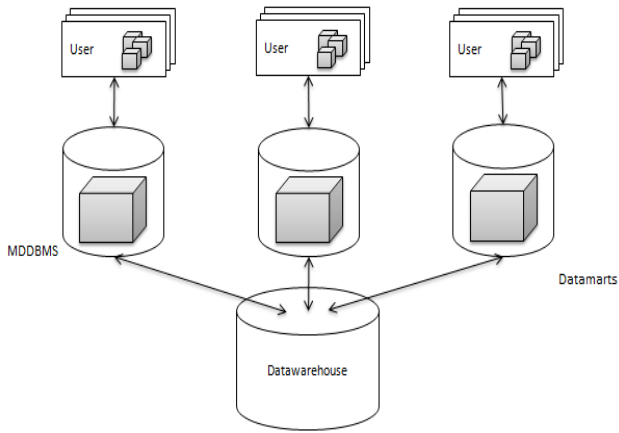


Figure 2.2: MOLAP architecture [10]

Compressed Row Storage (CRS) widely used due to simplicity and purity, with a weak dependency between array elements in a sparse array. In the proposed method of the CRS scheme in [11], it uses one one-dimensional floating-point array VL and two one-dimensional integer arrays RO and CO to compress all the nonzero elements along the rows of the multidimensional sparse array. The base of these arrays is 0. Array VL stores the values of nonzero array elements. Array RO stores information of nonzero array elements of each row. If the number of rows is k for the array, then RO contains the $k+1$ element. RO[0] contains 1; RO[1] holds the summation of the number of nonzero elements in row 0 of the array and R [0]. In general, RO[i] holds the number of nonzero elements in $(i-1)^{th}$ row of the array plus the contents of RO[i-1]. The number of non-zero array elements in the i^{th} row obtained by subtracting the value of RO[i] from RO[i+1]. Array CO stores the column indices of nonzero array elements of each row. Here's an example of the CRS scheme for a two-dimensional array.

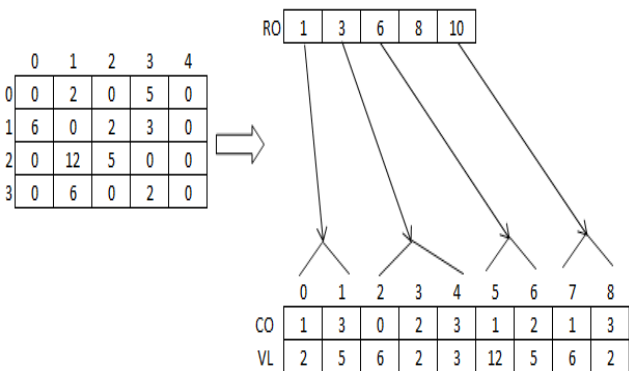


Figure 2.3: The CRS compressing scheme for sparse multidimensional array [11]

The Number of the nonzero elements of row 1 can be found by $RO[2] - RO[1] = 3$. The column indices of the nonzero array elements of row 1 stored in $CO[RO[1]-1]$, $CO[RO[1]]$, and $CO[RO[1]+1]$ i.e. $CO[2]$,

$CO[3]$, and $CO[4]$, since there are 3 nonzero array elements exist in row 1. Finally, the values of the nonzero array elements of row 1 can be found in $VL[2]$, $VL[3]$, and $VL[4]$.

III. METHODOLOGY AND IMPLEMENTATION

Decision support queries answered in the order of seconds on OLAP servers. So, it is pre-eminent to support highly efficient cube computation techniques, access methods, and query processing techniques for data warehouse systems [12]. In this paper, issues relating to the efficient computation of data cubes have explored. As the implemented static data warehouse has three dimensions (Model (), Year (), Color ()), and one fact table, this would like the following figure with their multidimensional views.

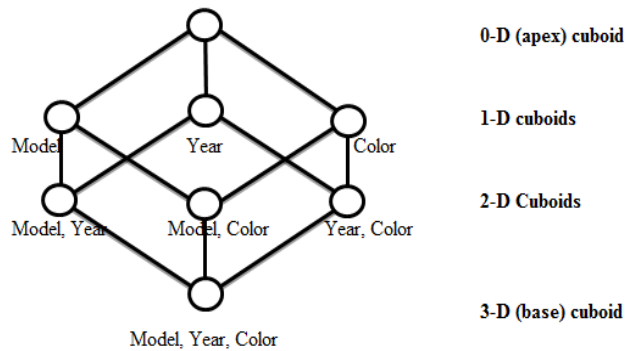


Figure 3.1: Lattice of cuboids, making up a 3-D data cube. Each cuboid represents a different group by; base cuboid contains three dimensions Model, Year, and Color.

a) Computing data cube for ROLAP

ROLAP differs significantly from MOLAP in that it does not require the *pre-computation* and storage of information. Alternatively, ROLAP tools access the data in a *relational database* throughout generating *SQL* queries to calculate information at the appropriate level as an end-user request it. With ROLAP, it is possible to create additional database tables (summary tables or aggregations) that summarize the data at any desired combination of dimensions [13].

For ROLAP, the two sub-problems take on the following specialized forms:

Data cube computation is defined by the scanning of the original data, employing the required aggregate function to all groupings, and generating relational views with the corresponding cube contents.

Data cube selection is the issue of creating the subset of the stored data cube views. Selection approaches avoid storing some parts of data cube items in line with certain criteria to create the balance between query latency and cube resource specifications.

Both of these problems studied in the past only in a fragmented fashion [14]. Some works to fill this gap

and presents the first systematic analysis of all relevant solutions. But that was only analysis base, here's the flowchart of our methodology of implementing ROLAP:

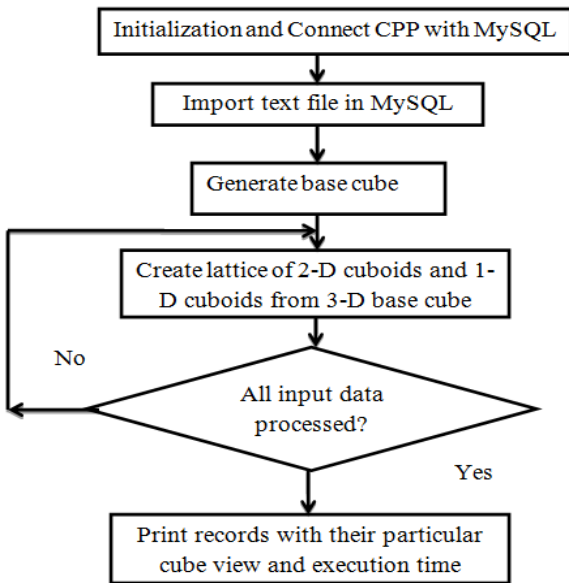


Figure 3.2: Flowchart of ROLAP implementation steps

The input table and generated cuboid are attached to appendix A.

b) Computing data cube for MOLAP

MOLAP supports the multidimensional view of data through array-based multidimensional storage engines. They map multidimensional views directly to the data cube array structures. Flowchart of the implementation phase of MOLAP is in figure 3.3.

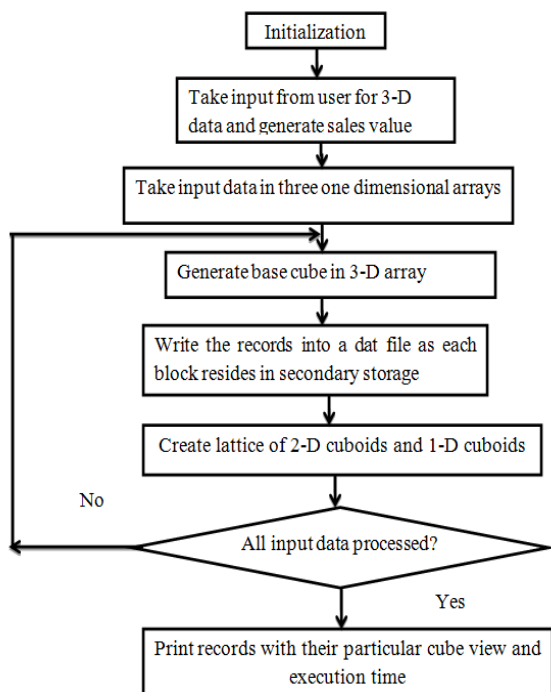


Figure 3.3: Flowchart of MOLAP implementation steps

c) Computing data cube for CRS

The main disadvantage comes from the fact that, in practice, cubes are sparse, with a large number of empty cells, making ROLAP and MOLAP techniques inefficient in storage space. To eliminate those empty cells, CRS is applied here. This row compression changes the physical storage format of the data associated with a data type but not its syntax or semantics. The flowchart of the implementation stages gives the following presentation.

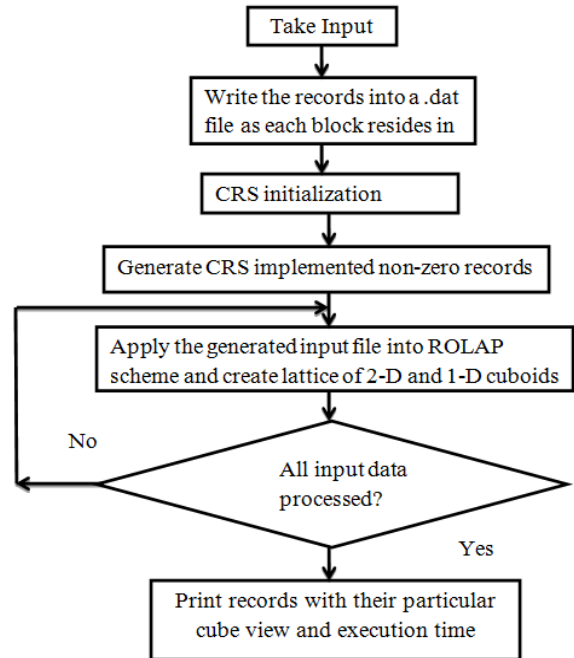


Figure 3.4: Flowchart of CRS implementation steps

The implemented view of ROLAP, MOLAP for a sample data set attached in Appendix B. Also, its slice and dice view is created. A very tiny amount of data manipulation is shown only because of reducing lots of paper work. As it has used a very small size of data, the nature of execution time is a little bit biased. In the next chapter, I have presented the graphical representation of ROLAP, MOLAP, and CRS with huge amount of data.

IV. RESULT ANALYSIS

In this experiment, I have used visual C++ and MySQL DBMS platform. A sample input table with a limited size of data and its generated output may look like the following:

Model	Year	Color	Sales
Chevy	1990	Red	41
Chevy	1990	Blue	67
Chevy	1991	Red	34
Chevy	1991	Blue	0
Ford	1990	Red	69
Ford	1990	Blue	24
Ford	1991	Red	78
Ford	1991	Blue	58
BMW	1990	Red	62
BMW	1990	Blue	64
BMW	1991	Red	5
BMW	1991	Blue	45
All	1990	Red	172
All	1990	Blue	155
All	1991	Red	117
All	1991	Blue	103
Chevy	All	Red	75
Chevy	All	Blue	67
Ford	All	Red	147
Ford	All	Blue	82
BMW	All	Red	67
BMW	All	Blue	109
Chevy	1990	All	108
Chevy	1991	All	34
Ford	1990	All	93
Ford	1991	All	136
BMW	1990	All	126
BMW	1991	All	50
All	All	Red	289
All	All	Blue	258
All	1990	All	327
All	1991	All	220
Chevy	All	All	142
Ford	All	All	229
BMW	All	All	176
All	All	All	547

Figure 4.1: Input table and generated cube for ROLAP

a) Base cube comparison

With data volume 64000 records (256KB), 125000 records (500KB), 216000 records (864KB), and 343000 records (1.3MB), a 3-D base cube has been created with their respective execution time according to three schemes, ROLAP, MOLAP, and CRS. These operations provide the following graphical views:

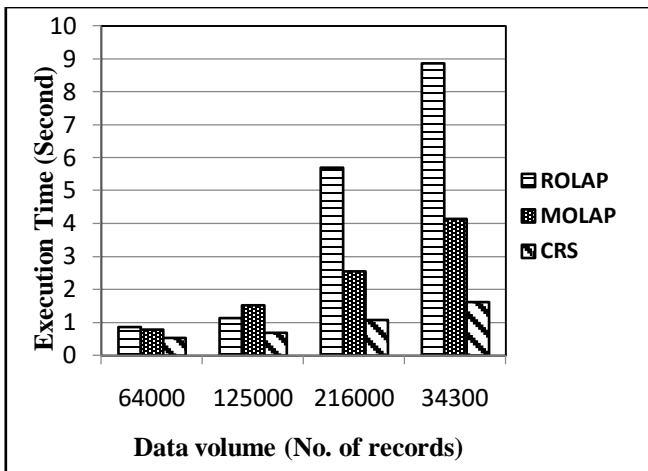


Figure 4.1: Comparison of the 3-D base cube of ROLAP, MOLAP, and CRS

From figure 4.1, the graphical plot of ROLAP gives the highest execution time, MOLAP gives better results compared to ROLAP, but with increasing density ROLAP getting worst, MOLAP takes a longer time where CRS provides a continuous compressed value with a short executing duration. This graphical representation

shows the underlying characteristics of these three methodologies.

b) Dice operation comparison

With the same data volume presented in the previous section, dice operations have been performed to create 2-D cuboids for ROLAP, MOLAP, and CRS. It creates three tables like 'Model-Year', 'Model-Color', and 'Year-Color' (as I use four columns named Model, Year, Color, and Sales showed on figure 4.1). In this section, all the 2-D cuboids of dice operations shown in the separate graphical plot. The following graphical representations give a clear view of the dice operation.

i. Model-Year view

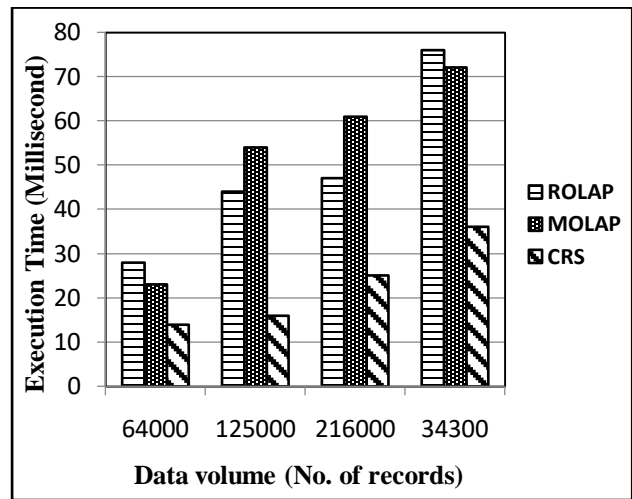


Figure 4.2: Comparison of dice operation (Model-Year) of ROLAP, MOLAP, and CRS

ii. Model-Color view

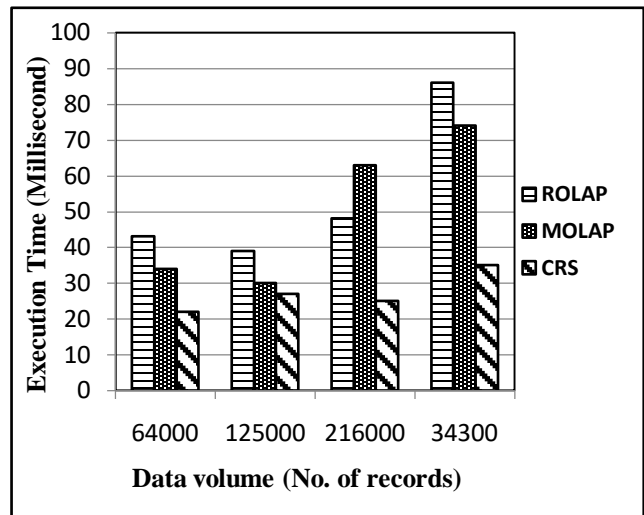


Figure 4.3: Comparison of dice operation (Model-Color) of ROLAP, MOLAP, and CRS

iii. Year-Color view

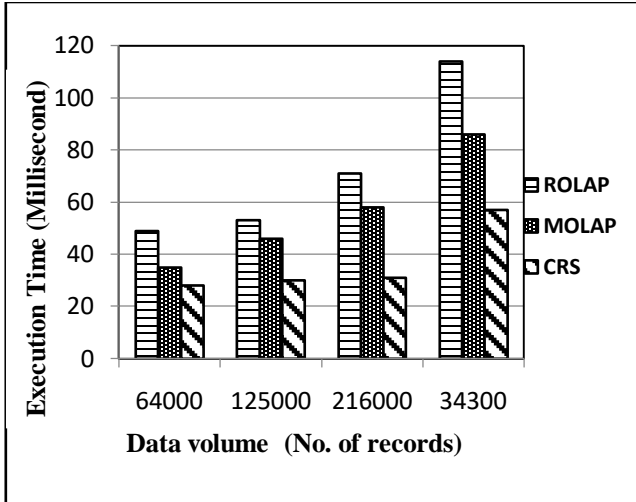


Figure 4.4: Comparison of dice operation (Year-Color) of ROLAP, MOLAP, and CRS

Dice operation gives nearly the same result as the base cube view. For a small amount of data, ROLAP gives roughly good outcomes than MOLAP, but with increasing density, it can cause the worst case. CRS always takes very little execution time in comparison with MOLAP and ROLAP.

c) Slice operation comparison

With the same data volume, slice operation has been performed to create 1-D cuboids and take execution time for both ROLAP and MOLAP. It creates three tables like all combinations of models 'Model', all combinations of years 'Year', and all combinations of colors 'Color'. In this section, all the 1-D cuboids of slice operations are shown in the separate graphical plot. These operations give a graphical chart shown below:

i. Model view

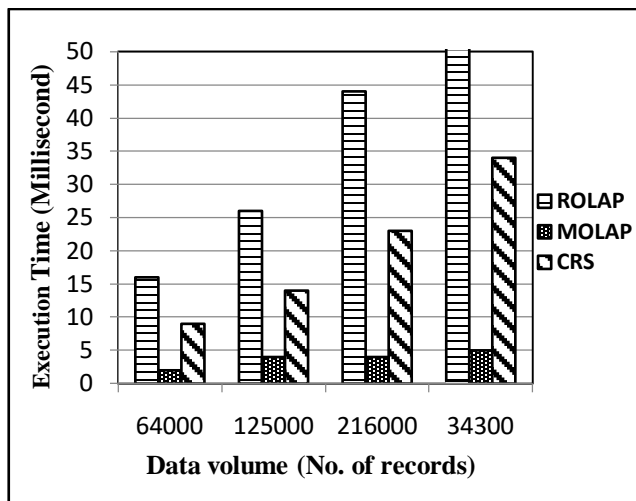


Figure 4.5: Comparison of slice operation (Model) of ROLAP, MOLAP, and CRS

ii. Year view

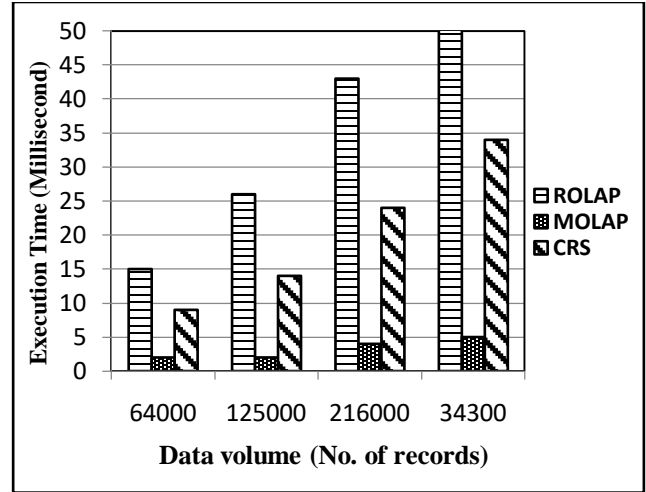


Figure 4.6: Comparison of slice operation (Year) of ROLAP, MOLAP, and CRS

iii. Color view

From the graphical view of slice operation, we found that MOLAP gives better results than ROLAP and CRS. It is because of the characteristics of the data, less dimension and also for the nature of the ROLAP scheme as we have implemented CRS through ROLAP.

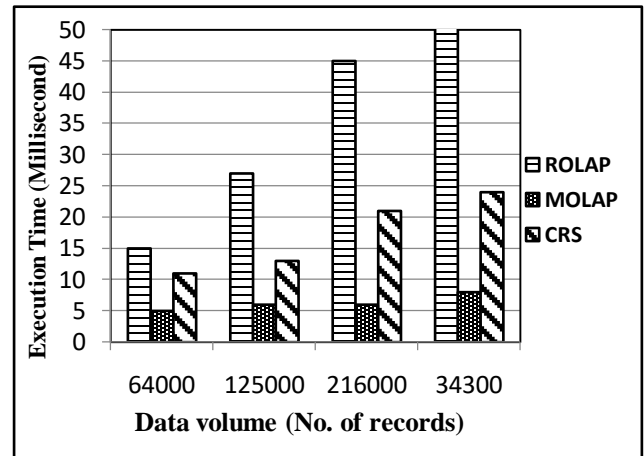


Figure 4.7: Comparison of slice operation (Color) of ROLAP, MOLAP, and CRS

In this chapter, ROLAP, MOLAP, and CRS implementation have been presented elaborately so that one can easily understand. Experimental results also discussed with the graphical figures. The performances of these three schemes have been measured concerning the execution time and data volume.

V. CONCLUSION

The objectives of this work are to implement ROLAP on base data, MOLAP on the multidimensional array, and implement CRS to eliminate empty storage cell. ROLAP has been implemented using a relational database through basic SQL queries; the base data

along with the dimensional table stored in the database and computes different cuboids with different memory allocation. MOLAP does not use the relational database rather than an optimized multidimensional array. CRS is implemented to remove zero values of storage to reduce memory wastage. Then the comparison of these three methods to find out that which gives better performance by the execution time and data density. Generally, MOLAP provides better performance with a small amount of data, if the data volume is high, the cube processing takes a longer time, whereas in ROLAP, data stored in the underlying relational database. ROLAP can handle a huge volume of data. Compressed Row Storage (CRS) on ROLAP to compress the aggregated data then applied. There are some scopes to extend this work in the future. Here, CRS is implemented through ROLAP only. However, in future, CRS can be integrated both with ROLAP and MOLAP, which can provide a more effective analysis of the advantages of applying CRS.

REFERENCES RÉFÉRENCES REFERENCIAS

1. Online analytical processing (last modified on 9 September 2014) http://en.wikipedia.org/wiki/Online_analytical_processing.
2. Antoshenkov G. & Ziauddin M. (1996), Query processing and optimization in oracle Rdb, VLDB Journal, 229–237.
3. Buccafurri F., Furfaro F., Sacca D. & Sirangelo C. (2003), A quad tree based multirole solution approach for compressing data cube, SSDBM 2003, 127-140.
4. Barret R., Berry M., Chan T.F., & Dongara J., Eljkhout V., Pozo R., Romine C. & Van H., (1994). Templates for the solution of linear systems: Building blocks for the iterative methods, second ed. SIAM.
5. Bassiouni M. A., (1985), Data Compression in Scientific and Statistical Databases, IEEE Transaction on Software Engineering, 1047-1057.
6. Acharya S., Gibbons P. B., AND Poosala V. 2000. Congressional samples for approximate answering of group-by queries. In Proceedings of ACM Special Interest Group on Management of Data (SIGMOD). 487–498.
7. Agarwal, S., Agrawal, R., Deshpande, P., Gupta, A., Naughton, J. F., Ramakrishnan, R., And Sarawagi, S. 1996. On the computation of multidimensional aggregates. In Proceedings of Very Large Data Bases (VLDB). 506–521.
8. Acharya S., Gibbons P. B., and Poosala V. 2000. Congressional samples for approximate answering of group-by queries. In Proceedings of ACM Special Interest Group on Management of Data (SIGMOD), 487–498.
9. Relational Online Analytical Processing (ROLAP) (Copyright © 2010 - 2014 Janalta Interactive Inc) <http://www.techopedia.com/definition/14719/relational-online-analytical-processing-rolap>.
10. Gray J., Bosworth A., Layman A., 1996. Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-total. In Proceedings of International Conference on Data Engineering (ICDE), 152–159.
11. K. M. Azharul Hasan, Tatsue Tsuji, Ken Higuchi, 2007. "An efficient implementation for MOLAP basic data structure and its evaluation", DASFAA 2007, LNCS 4443, 288 – 299.
12. Han J., Pei J., 2001. Efficient computation of iceberg cubes with complex measures. In Proceedings of ACM Special Interest Group on Management of Data (SIGMOD), 1–12.
13. Liang W., Wang H., & Orlowaska M. E. 2001. Materialized view selection under the maintenance time constraint, 203–216.
14. Konstantinos Morfonios, University of Athens, ROLAP Implementations of the Data Cube, ACM Computing Surveys, Vol. 39, No. 4, Article 12, October 2007.

APPENDIX

A. Input table and generated ROLAP cube

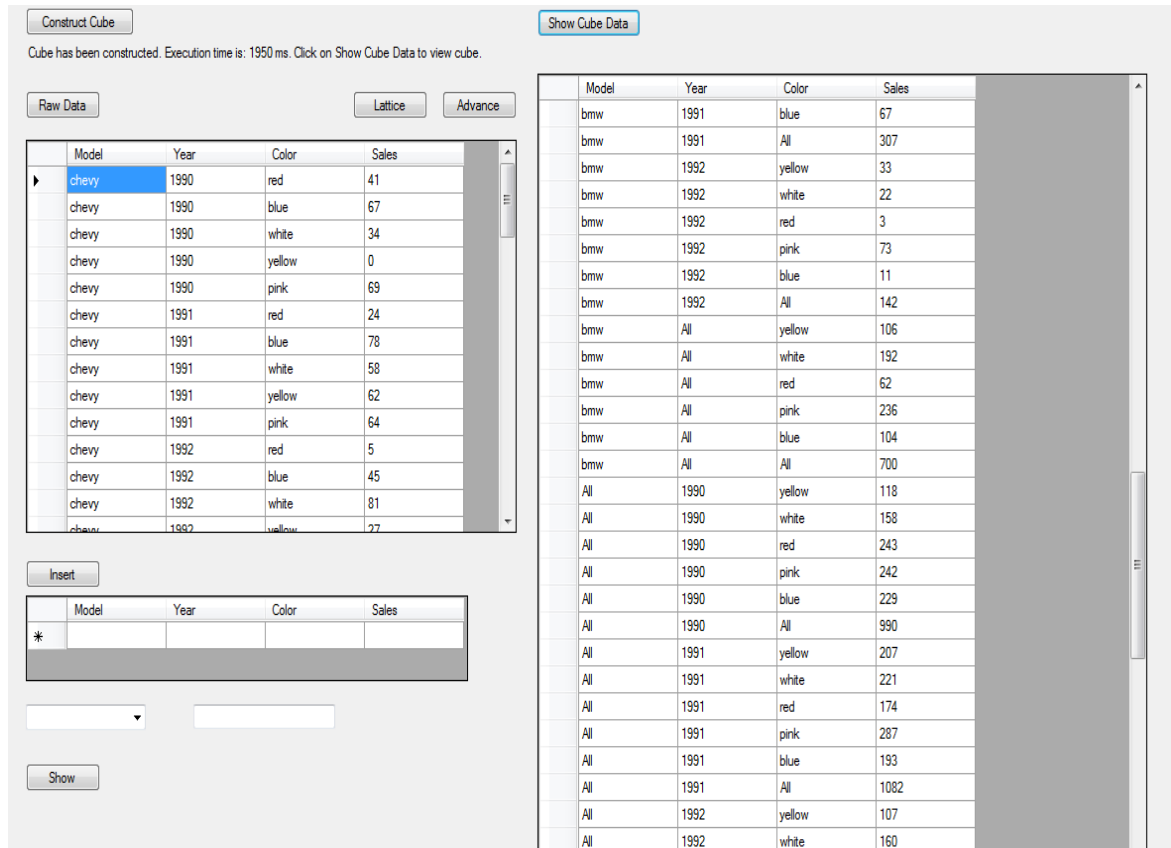


Figure A.1: Input table and computed base 3-D data cuboid for ROLAP

B. A sample input table and comparison of slice-dice execution time

Table B.1: Table for input data

Model	Year	Color	Sales
Chevy	1990	Red	0
Chevy	1990	Blue	57
Chevy	1990	White	0
Chevy	1991	Red	81
Chevy	1991	Blue	59
Chevy	1991	White	48
Ford	1990	Red	0
Ford	1990	Blue	90
Ford	1990	White	83
Ford	1991	Red	75
Ford	1991	Blue	0
Ford	1991	White	86
BMW	1990	Red	72
BMW	1990	Blue	52
BMW	1990	White	0
BMW	1991	Red	0
BMW	1991	Blue	0
BMW	1991	White	0

The dice operation produces a sub cube by allowing the analyst to pick specific values of multiple dimensions.

The implemented Dice view of ROLAP, MOLAP and CRS for three Models, two Years and three Colors gives the representation alike:

Table B.2: Table for Dice operation with Year-Color

Execution Time	Model	Year	Color	Sales
ROLAP: 27 ms MOLAP: 15 ms CRS: 15 ms	All	1990	Red	72
	All	1990	Blue	199
	All	1990	White	83
	All	1991	Red	156
	All	1991	Blue	59
	All	1991	White	134

Table B.3: Table for Dice operation with Model-Color

Execution Time	Model	Year	Color	Sales
ROLAP: 28 ms MOLAP: 16 ms CRS: 17 ms	Chevy	All	Red	81
	Chevy	All	Blue	116
	Chevy	All	White	48
	Ford	All	Red	75
	Ford	All	Blue	90
	Ford	All	White	169
	BMW	All	Red	72
	BMW	All	Blue	52
	BMW	All	White	0

Table B.4: Table for Dice operation with Model-Year

Execution Time	Model	Year	Color	Sales
ROLAP: 31 ms MOLAP: 16 ms CRS: 14 ms	Chevy	1990	All	57
	Chevy	1991	All	188
	Ford	1990	All	173
	Ford	1991	All	161
	BMW	1990	All	124
	BMW	1991	All	0

Slice is the act of picking a rectangular subset of a cube by choosing a single value for one of its dimensions, creating a new cube with one fewer dimension.

The implemented *Slice* view of ROLAP, MOLAP and CRS for three Models, two Years and three Colors gives the representation alike:

Table B.5: Table for Slice operation with Model

Execution Time	Model	Year	Color	Sales
ROLAP: 13 ms	Chevy	All	All	245
MOLAP: 2 ms	Ford	All	All	334
CRS: 5 ms	BMW	All	All	124

Table B.6: Table for Slice operation with Year

Execution Time	Model	Year	Color	Sales
ROLAP: 12 ms	All	1990	All	354
MOLAP: 2 ms	All	1991	All	349
CRS: 4 ms	All	1991	All	349

Table B.7: Table for Slice operation with Color

Execution Time	Model	Year	Color	Sales
ROLAP: 11m ms	All	All	Red	228
MOLAP: 1 ms	All	All	Blue	258
CRS: 6 ms	All	All	White	217

Table B.8: Table for base cube generation time

Method	ROLAP	MOLAP	CRS
Execution Time	234ms	32ms	121ms





Algorithm and Design Techniques - A Survey

By A. Sheik Abdullah, T.G.R Abiramie Shree, P Priyadharshini
& T Saranya

Abstract- Algorithm design plays a significant role in development of any application that is concerned with engineering and technology. Advancement in implementation levels of algorithms made a good impact with the model developed. Meanwhile, the time and space complexity of the execution of the algorithm varies with regard to the input to the algorithm upon fixation of various parametric levels. This paper summarizes a survey on various algorithm design techniques and its applications. The applicability of the algorithms varies with regard to the problem and the nature of computation levels.

Keywords: algorithm design; linearity; dynamic programming; greedy techniques; non-linearity.

GJCST-C Classification: I.1.2



Strictly as per the compliance and regulations of:



Algorithm and Design Techniques – A Survey

A. Sheik Abdullah ^α, T.G.R Abiramie Shree ^σ, P Priyadharshini ^ρ & T Saranya ^ω

Abstract- Algorithm design plays a significant role in development of any application that is concerned with engineering and technology. Advancement in implementation levels of algorithms made a good impact with the model developed. Meanwhile, the time and space complexity of the execution of the algorithm varies with regard to the input to the algorithm upon fixation of various parametric levels. This paper summarizes a survey on various algorithm design techniques and its applications. The applicability of the algorithms varies with regard to the problem and the nature of computation levels.

Keywords: *algorithm design; linearity; dynamic programming; greedy techniques; non-linearity.*

I. INTRODUCTION

An algorithm is a step by step procedure which helps us to solve a problem in a sequential order. The problems are all different from one another. There are different techniques of algorithm which are suitable for each problem. The techniques are Divide and Conquer method, Greedy Method, Dynamic Programming and Backtracking. In divide and conquer, the original problem is divided into sub problems and these sub problems are solved individually then combine the solved problems to get a solution for the original problem. In greedy algorithm, it produces an optimal solution for a problem. Greedy algorithm is generally used to solve a complex problem rather than an easy one. It breaks the complex problem into a small instance and solve them recursively until the best optimal solution is gathered from the small instances of that problem. Dynamic programming technique is used to optimize the result by divides the problem into smaller sub problems and solving the smaller problems to obtain the ultimate problem. This technique recursively solves the program to obtain the optimal solution for the problem. Backtracking algorithm is an optimization technique where the solution for the problem can be backtracked many number of times until the best solution is obtained from the algorithm. The backtracking process is different for the different problem. Branch and bound technique is the another method of solving a problem to get the optimal solution. This algorithm technique is mainly used to produce the lower cost for the problem along with optimal solution. The last technique is the linear programming whereas the solution for the problem includes the maximum profit, shortest path and also in lower cost for that solution. In algorithm, the there are two types of data

structure. They are liner data structure and nonlinear data structure. In linear data structure, the data are arranged in the sequential order but in nonlinear data structures, the data are not arranged in sequential order but in random manner. The linear and nonlinear structures helps to store the data with the algorithm technique this data can be efficiently handled. There are many algorithms which help to manipulate the data in different forms. The linear algorithms are array, linked list and stack and queue. The array helps to store data in the each index present in the array. The linked list is of three types, they are single linked list, double lined list and circular linked list. These linked lists are used to store data in the form of node. Each node consists of both data section and link section, where the data section consists of the data of each node and in the link section it contains the address of the next data to be stored. In stack data structure, the data are stored in FILO fashion. The first entered data are to be deleted at the last time form the stack. In the nonlinear data structures the algorithms presented are trees, graphs, dictionaries, heaps and tries. In trees, the data are stored in tree format where the tree can be in various forms such as binary search tree, AVL tree, B tree and Splay tree. In graph data structure, there are algorithms such as Bellman ford and Floyd warshall. In dictionary, skip list algorithm is used to help the data structure for the efficient storing of data. In network data structure, the Ford Fulkerson and Edmond karps algorithm are used to store the data in handling the data in the network format. In tires, there are three formats. They are standard tries, compressed tries and also suffix tries. In addition to these algorithms there are text matching algorithms such as Brute force algorithm, Boyer Moore algorithm and KMP algorithm. The text compression algorithms are Huffman coding algorithm where the compression of data is used to handle the encoding of data.

II. LITERATURE SURVEY

Maximum Independent Set Approximation Based on Bellman-Ford Algorithm by Mostafa H. Dahshanis proposed by in this algorithm to be found the approximate solution for the maximum independent set problem. It is treated to least cost problem to be taken by in this novel approach. It can be used to bellman ford algorithm adapted version Source is consist of all vertex and vertex should be measured by the number of vertex excluded in this vertex independent set are included. The run time of the these algorithm is approximately

Author ^{α σ ρ ω}: Department of Information Technology, Thiagarajar College of Engineering, Madurai. e-mail: asait@gmail.com

$O(n^2-m)$). The proposed algorithm is significant changed by several bench marks and random generated graph are improved. In this method include one of the best greedy algorithms. Future work is proposed algorithm to be only focused on run time and space requirement and lower bound theoretical independent numbers.

Mathematical formulations and exact algorithm for the multitrip cumulative capacitated single-vehicle routing problem by Juan Carlos Riveraa,b,*, H. Murat Afsara, Christian Prinsa is proposed by In this paper is CCRVP disaster manage the single tripe arrive problem. Single vehicle arrive the short period reach in to destination and to successive tripe to set off affect sites in which that select the minimization of standard. If it is used to two types of programs like flow based model and set partitioning model. Bellman ford algorithm to direct acyclic resource constrains so it's reduce the path of the sites. The future work the version of node routing problems to be detect in selective VRP. Mt-CCRVP includes the column generation scheme. Similar to the sum of the arrival time to be reflected is also interesting the priorities.

An Efficient Implementation of the Bellman-Ford Algorithm for Kepler GPU Architectures by Federico Busato and Nicola Bombieri, Member, IEEE. Is proposed by in this algorithm find the single path problem it is find best solution for single source shortest path. They are applied to many core architecture of parallelized. High degree of parallelism is cannot surely performed at the cost of low work adeptness. If they are compare to the similar algorithm the work consumption is a waste of power. The bellman ford algorithm parallel implement to in this process it can be involved to exploits architecture of recent GPU architecture. To improved GPU performance and work efficiency. In this paper focused to two things one is algorithm and another one is GPU architecture. The experimental result of in this proposed work is more implementation of parallel ssp. it works on graph are inefficient. Future work is done by recently available for open CL and CUDA in this algorithm implement used to CL and CUDA. Lower level instruction based implements the different hardware oriented techniques.

Parallel Implementation of Bellman-Ford Algorithm Using CUDA Architecture by Ganesh G Surve, Medha A shah is proposed by the larger graph involves to multiple of vertex and edges. The millions of vertex are apply many real time common applications and face to many challenges. Now a day multiple of application like travelling problem, shortest path problem, routine network and robotic system in these application are based to data routine techniques in this data represent in a graph. The graph is direct and negative cycle weighted graph. Now a day growing the data application but still situation need for speed and real time response. The serial algorithm takes large amount

of time. An optimization problem in graph theory of bellman ford algorithm easy to solve the single source shortest path problem. In this paper improve the GPU architecture of NVIDIA performance and workload efficiency and it is implement the bellman ford algorithm. In this paper newly introduce the parallelizing Bellman-Ford Algorithm and implement the new version of the algorithm over the CUDA framework. The NVIDIA programming interface name is CUDA. The future work is done by they are modifying the bellman ford algorithm. The divided in to the partitioning and execution of algorithm for both CPU and GPU architecture. They are used to obtaining higher massive parallelism and vectorization.

Efficient Implementation of the Bellman-ford algorithm for GPU by Marjan Nazarifard, Davoud Bahrepour is proposed by Bellman ford algorithm is only focused to solve the shortest path to the source node. It can be enabling to negative cycle weighted graph. It is fine single source path problem and analysed by graph used to common algorithm. Moreover, it is represents a class of parallel algorithms, the memory accesses and work distribution of which are both irregular and data-dependent. Recently, graphics processors have been used for implementing many algorithms, as well as an accelerator in supercomputers. Several SSSP algorithms have been proposed based on graphics processing units (GPUs), each of which could traverse a specific type of graph. In this paper, we accelerated the Bellman-Ford algorithm on GPU using CUDA, so that it could traverse dense and sparse graphs (regular and irregular) within the shortest time compared to the previous algorithms. According to the simulation results, the proposed implementation provided an average speed-up of 1.87x compared to most of the previous parallel implementation algorithms. Future work is done by If it is future work to remove the feck threat in beginning of the process. So reduce the time and memory. The way of feck threaded path to be allotted by another thread.

Demonstration of Single Link Failure Recovery using Bellman Ford and Dijil (stra Algorithm in SDN by Syed Waleed, Muhammad Faizan, MaheenIqbal, Muhammad Irfan Anis is proposed to In this paper to be found the failure recovery mechanism was software defined network it is conducted to development of employee nodes. In this SDN domain is rarely used the algorithm field. It's allows to network inside programmability. In node failure recovered dynamic application building of bellman ford algorithm. The bellman ford algorithm is implemented by loop confrontation and reduction. They are compare to another second algorithm is Dijkstra's algorithm. Dijkstra's algorithm failure recovery mechanism to be compared based on the node bandwidth allocation. The future work is the Dijkstra's algorithm techniques are reducing the processing tine and calculate the any

failure occurrence. These techniques used to SDN architecture multiple fault domain failure recover solution.

A bag of paths framework for network data analysis by Kevin Francoisse, IlkkaKivimaki, Amin Mantrach, Fabrice Rossi, Marco Saerens, is proposed to analysis the network data it is generic framework as (BoP). The network path is assign to the probability distribution format. The two node are connected to the probability capture notation based and It has connect high accessibility preferably low cost paths so it is extended to the bellman ford algorithm. Future work done by to be investigating the graph cut the path instead of link. We also plan to evaluate experimental the potential distance as the distance between sequence of character by adapting it to a directed acyclic graph.

To be investigating the graph cut the path instead of link. we also plan to evaluate experimental the potential distance as the distance between sequence of character by adapting it to a directed acyclic graph by Anggie Nastiti, Andrian Rakhmatsyah, Muhammad Arief Nugroho, is proposed to In Telkom University, the topology used does not have backup link for campus internal network in case of link failure because the topology is still based on inter VLAN where each switch only has one path to switch core. Data packets cannot be delivered from source to destination if there is a link failure on the path. Based on the problem, it is proposed a new architecture which is Software Defined Network (SDN) that can overcome the link failure by configuring the controller in order to move to alternative links that have been provided with Open Flow. This architecture separates the control plane and data plane, so it is centralized and programmable. To look for alternative links when a link failure occurs, the shortest path algorithm is Dijkstra and Bellman-Ford algorithms. The test parameters performed in this research are functionality to determine whether the two algorithms can determine the path or not, and convergence time to find out how long it takes to form the path from source to destination. Scenario of the test is done before and after the link failure occurs by using Ryun as controller and Minuet as emulator. Based on the results of the tests conducted, it was found that Dijkstra and Bellman Ford algorithm can be applied well on link failure emulation in accordance with the scenario and topology used in the test. In addition to convergence time parameters obtained that Dijkstra algorithm is superior compared to Bellman-Ford algorithm. The difference gained in both scenarios has a value that is not so great the difference. The future work done by some links are disconnected, both Dijkstra and Bellman-Ford algorithms find alternative paths to delivery data packets from sources to destination. As for the convergence time test, it was found that Dijkstra algorithm is superior to the Bellman-Ford in scenarios before and after a link

failure occurs. The difference between them is also not big. This is because the Bellman-Ford algorithm to find the shortest path always checks if there is negative weight cycles or not, so as to make search path becomes longer. Differences in the number of links it was decided also to affect the value of convergence time obtained.

On the optimality of bellman ford Moore shortest path algorithm by Stasys Jukna, Georg Schnitger, is proposed to The lower bound of the Switching and rectifier networking size is over any semiring is a zero characters. The minimum semi ring is also zero character. So it is used to bellman ford- Moore dynamic algorithm in which to find the shortest path for S-T. the future work is done by In this paper consist of single variable labelled Switches. If can extended in this model added to the some Switching networks. It is extended to random combination with integral factor. So albeit bellman fords Moore switching network does not add some peripheral structures in which problem to other minimization.

Face Image Abstraction by Ford-Fulkerson Algorithm and Invariant Feature Descriptor for Human Identification by Dakshina Ranjan, Kisku Debanjan Chatterjee, S. Trivedy, Massimo Tistarelli is proposed by In this paper discuss about the face image abstraction let used to SIFT features and ford Fulkerson algorithm. Ford Fulkerson algorithm solves the maximum flow in flow network. It can be drawn on face image extract from SIFT feature. If find the augmenting path is exit then and Augmenting path fine the based on vertex. The flow of source to destination of flow value is augmenting path. All vertexes along with edges calculated to one of the paths. In this process until obtained to multiple of times produce the number of different flow paths. The path to be compute residual capacity of augmenting path. The result of in this paper is capture the face image directed graph contains spare characteristics of objects. Ford Fulkerson algorithms apply to direct graph maintain a capacity. Run time of algorithm is $o(VE)$ v is a vertex and E is a edges. Future work is used to face image meth pairs for calculating matching proximity. They are focused to the maximum flow of key words.

Hydraulically informed graph theoretic measure of link criticality for their salience analysis of water distribution networks by Aly-Joy Ulusoy, Ivan Stoianov and Aurelie Chazerain is proposed by in this paper algorithm is include by water Distribution network. This network to be complex for interconnected. It is built to resilience based on energy redundancy. Failure condition operation is maintained. There are analyse the salience of WDN used various methods it is analysed to base on sorrow gate way network measures. The future work is done by apply the WFEB method. They are arrange to operational network in validate of order to robustness. The also explore the hydraulically in formed graph. The theoretic link measure to be critical approach

of hydraulic models. In this model based salience analysed.

Chance distribution of the maximum flow of uncertain random network by Yuhong Sheng and Jinwu Gao is proposed by the uncertain variables and other random variables are called as uncertain random network. In this paper to be solve the maximum flow of uncertain random network. It might be change the distribution of uncertain random network. In this paper implement the maximum flow of distributed uncertain random network. It is only focused on maximum flow of uncertain network. The future work is data analytics method to be used and improve the sufficient mathematical approach.

Minimax properties of some density measures in graphs and digraphs by Janet Anderson, Hong-Jian Lai, Xiaomin Li, Xiaoxia Lin & Murong Xu is proposed. For a graph G , let $f(G)$ denote the connectivity $\kappa(G)$, or the edge connectivity $\kappa(G)$, or the minimum degree $\delta(G)$ of G , and define $f(G) = \max\{f(H) : H \text{ is a sub graph of } G\}$. Mutual in $[K\text{-components, clusters, and slicing in graphs, SIAMJ. } x \text{ theorems related to } \delta(G) \text{ and } \kappa(G)$, and obtained polynomial algorithms to determine $\delta(G)$, $\kappa(G)$ and $\kappa(G)$. The restricted edge-connectivity of G , denoted by $\lambda_2(G)$, is the minimum size of a restricted edge-cut of G . We define $\lambda_2(G) = \max\{\lambda_2(H) : H \subseteq G\}$. For a digraph D , let $\kappa(D)$, $\lambda(D)$, $\delta^-(D)$ and $\delta^+(D)$ denote the strong connectivity, arc-strong connectivity, minimum in-degree and out-degree of D , respectively. The future work is the connectivity of graph is activate to the polynomial algorithm method we are try to new version of the algorithm.

Using basis dependence distance vectors in the modified Floyd–Warshall algorithm by Włodzimierz Bielecki · Krzysztof Kraska · Tomasz Klimek is proposed to In this paper, in this paper to be focused on Floyd–Warshall algorithm. Floyd–Warshall algorithm to be modified for this paper approach. Where the algorithm is most dependent to time consumption it is calculated to self-dependent of loop statement. In this statement is applied to self-dependence. The self-dependence is consisting of distance vector derived from all vectors description. The present approach is reducing the transitive closure calculation. Transitive closure of dependence graph is increase the applicability scope. It is being to build for optimization compiler. The results of experiment for parallel benchmark are discussed in NASA. Future work is application of the presented approach for extracting both coarse- and fine- grained parallelism for different popular benchmarks.

Probabilistic Calculation of Tolerances of the Dimension Chain Based on the Floyd–Warshall Algorithm by A. V. Muholzoeva, V. B. Masyagina, b,* is proposed to In this paper proposed to process to mechanical engineering is a time consuming of tolerance analysed it take requiring automation. In this algorithm developed to individual chain to be identified.

In this task is more difficult. The algorithm is avoiding the difficult logarithmic procedure. They are followed by the Floyd war shall algorithm for probability of closing tolerance units. It calculates the graph length of the path by adding the length of pairs. The algorithm is identifying individual dimension. In this method to be complex for solve the entire structure of the graph. The future work is added to calculate the dimensional circuits and significantly implemented. They are used new version of the graph algorithm.

Shmuel T. Klein, MiriKopel Ben-Nissan (2009) proposed Boyer Moore algorithm can also be used for the binary data where the processing can only be done on the entire blocks so the number of comparisons done can be reduced. This method is applied to the BM algorithm with small change in the delta value. The mismatch occurs only when the length of the suffix of the pattern bytes is not matched with the length of the full text bytes. Here, instead of comparing the one bit with other it can also be done with a four bytes' comparison such as word. It reduces the comparison and also the time complexity of the process. In future, the authors proposed that it can be also used for the Fibonacci codes where the fast search is possible in the binary data.

Frantisek Franek, Christopher G. Jennings, W. F. Smyth (2007) proposed the combination both KMP Knuth Morris Pratt Algorithm along with the Boyer Moore algorithm to form a new algorithm which is a hybrid algorithm. Here, it compares the text with the pattern in the normal form as done on KMP algorithm if the text is not matched with the pattern then the Sunday shift takes places where the method is similar to the BM algorithm's whole shift process where the text is not found anywhere in the pattern. The benefits of both KMP and BM is combined to form this hybrid algorithm. Thus, it producing a time and space complexity of $(m + k)$ times. This proposed method can be applied to the faster algorithm where it avoids the text that are not matching with each other.

Ain Zubaidah Mohd Saleh, Nur Amizah Rozali (2015) proposed the Boyer Moore algorithm which can be used to find the vulnerability in the websites. This method is more useful than that of the other traditional methods in terms of results. Here, the false positive is completely avoided to produce higher results. Uniform Resource Locator (URL) is used in the BM algorithm to detect the vulnerability for finding the two process which are efficiency and accuracy of detection of the vulnerabilities. In future, this proposed method can also have applied with hybrid algorithm to provide more accurate results.

Phyela Mbewe, Sampson D. Asare (2017) proposed that the Huffman coding algorithm which is a text compressing algorithm is used in the images. The image files are compressed with the help this algorithm to reduce the size and time complexity of the process.

Here, the Adaptive Huffman coding results is compared with the Arithmetic Huffman coding results to get the precise results for the image compression. The Arithmetic approach is better the Adaptive in terms of the space where the Adaptive algorithm is better than the Arithmetic algorithm in terms of the time. Here, this method is applicable only to the files of images which are all in larger size. In future, the authors proposed that it can also be applied to the social networking and the big data images also.

Yosang Jeong, Myungho Lee (2015) proposed that the Boyer Moore algorithm calculates the two shift rules of the strings in the preprocessing phase. In the second phase, the matching operations are to be performed against the text of the string. The pattern matching process is a time consuming process and it requires parallelization of the process to reduce the time consuming process. This can be achieved in most of the CPUs and also applied to the many core processors. It can also be applied to the multi core processors. This type of parallelization is helpful for the optimization of the load balancing using threads, and also helps the results generated are less time consuming than the normal process. In future the authors suggested that this can also be applied to the GPUs such as CUDA programming which also helps to retrieve the pattern matching of the strings.

Ahsan Habib, Mohammad Shahidur Rahman (2017) suggested that the compression process of the Huffman coding is same but it has different decoding process which helps in fast production of the results. It can be done with the help of the quaternary tree helps to reduce the time used by the algorithm. It contains both the quaternary Huffman encoding and quaternary Huffman decoding process. It is done by replacing the binary tree with the use of the quaternary tree. This process can be used in the client and server environment which reduces the time delay between the decoding processes. It provides high end security with less amount of time taken for the decoding purpose.

Thierry Lecroq (1994) proposed that the Boyer Moore algorithm where the position of i in the text, which helps to compute the length of the longest suffix which is to be find from the text ending at the position. This technique is used to find the text form the word. This method is better than that of the earlier method of Boyer Moore method. This automaton helps to find the largest number of word from the process. This process is further used to calculate the matching pattern.

Yih-Kai Lin, Shu-Chien Huang, and Cheng-Hsing Yang (2011) proposed that the new Huffman Coding method is better than that of the conventional method of Huffman coding. Here the result produced by the new algorithm ranges from 1.91 to 2.13 where the processing unit is 10. Tree grows and Tree prune is applied to the decoded version the Huffman code tree structure. The running time increases with due to the

increase in the cache misses and also y the reduced time from the average decoding symbol of the table. This process maintains the efficiency of the decoding process of the algorithm.

Bruce W. Watson (2002) proposed that the new algorithm form a precomputed and tabulated function with the help of the shift functions. This shift functions are far better than that of the Knuth Mooris Pratt algorithm where the time complexity is much reduced in this process. Here, the authors used to approach two methods where the first approach helps to use the shift distance which improves the algorithm. In second approach, the shift distance helps to find the matching text from the pattern.

Wei-Wei Lu and M.P. Gough (1998) proposed that Huffman coding algorithm with the use o two trees. They are front tree and back tree. The front tree is an adaptive Huffman code, the symbol is encoded only with the number of occurrence or frequency of the symbol. The symbol with high frequency is placed in the front tree where as the lower frequency is placed in the back tree. The proposed algorithm is 2.5 times faster than that the Huffman coding. The compression efficiency of the new algorithm is affected by the dispersion of the data. Here, the new algorithms have less dispersion of data where the compression of the data is practically higher.

Ghim Hwee Ong (2000) proposed that Heap sorting method is used to sort the data in the Huffman coding algorithm where this method provides efficient result than that of the traditional approach. Here, the Heap sorting method i applied on the binary tree of the algorithm. This algorithm provides less time complexity in worst case also. This process is applied to the reverse process applications which are required in the real time process.

Steven Pigeon, Yoshua Bengio (1993) proposed a new method Huffman Coding method consists three conceptual methods. The methods are set representation, set migration and tree rebalancing. Set migration moves symbols form one set of data to other. This process helps to ease the rebalancing method where the time complexity is at the $O(\log p(s))$ times. This method concludes that the M algorithm is much slower than that of this Huffman coding algorithm. In future, this method can efficiently apply for the memory management method.

Wang Zhe, Chen Jun, Yuan Gang Zhao Zhou Yan (2001) proposed that a new algorithm where the matching of image is done with the help of KMP algorithm. Extraction of the contour, extracting the contour of fragments as done with the help of this algorithm. The whole algorithm depends on the hardware which provides efficiency and accuracy. This can be widely applied to the medical field for the detection of diseased or affected region of the body from normal function.

Qingzhu Meng, Zhenming Lei, Dazhong He (2017) proposed that the KMP algorithm with modification that can be applied to the traffic analysis. Partition algorithm is used here is the k means algorithm produces the actual solution to the problem which improves the efficiency of the KMP. This KMP algorithm is customized with the traffic analysis. This algorithm can be applied to the hardware application is also depends on the efficiency and accuracy of the data. This method is efficiently applied for the secure environment where the traffic signals are to be prompted with this approach.

The improvement of the brute-force searching algorithm is proposed in the name of Star-End-Mid algorithm by Abdeen. This proposed algorithm is not preprocess neither the pattern nor the text to perform searching. The start-to-end algorithm start the searching process by comparing the first character of the pattern and first character of the given text. If the first character is matched then the last character of the pattern and last character of the text from taken sample. If the last character of the sample matched then character by character comparison will take place for the taken sample, for the remaining character there is no need to take comparison. The proposed algorithm start-end-mid avoids the preprocessing for the pattern therefore this improves the time complexity involved in brute-force algorithm. The future work of this algorithm can be implemented for effective string searching on huge volume of data and also well suited for hospital patient management software for simple and fast search of patient details with the given sample of pattern. The algorithm Start-End-Mid works based on the idea of first, last and mid character of the pattern is compared with the first, last, mid character of the text for the given sample. This start by separating the text into two segments from the index 0. In the second step it will compare the first character if the match occur it will move on to the second step else it will move on to next index of character. In the second step the last character of the pattern is compared with last character of the text if the match occurs then it will proceed the next step else it will go to the second step. In the third step it will check the length of the pattern if it has two character then follows the next step. If it has more than two character then compare the floor(length of the pattern/2) of the pattern with the text floor(length of the pattern/2) character, from this the if a match occur, it will take two the next step else it will work ok next segment of the text and follow the step 1. The improved version of brute force algorithm that is Start-End-Mid working process of checking has been improved the time of searching by avoiding the character by character matching comparison. The Start-End-Mid algorithm time complexity is $O((n-m)+1*(m-3))$. LUPIN by using the brute force algorithm, topology of the wireless network is optimized. This optimization will ensure the reduction

involved in computational complexity of the algorithm by implementing multi-thread application for processing optimization. The topology of wireless network connection should contain point-point, point-multipoint, peer-peer. These stages of network topology of spatial distribution of forming communication channel this in turn to create various types of optimization algorithm. The requirements need to be undertaken while optimizing the network topology such that cost, level of security, uniformity. The genetic algorithm is often used to solve problems in optimization, which is based on natural selection. The genetic algorithm can be used where the standard algorithm can't work well to resolve the optimization problem. There is computation complexity and difficulties in genetic algorithm as well as in modified genetic algorithm (Bhondekar et al). the proposed algorithm for optimization is based on search algorithm called Brute Force Algorithm. Due to the improvement in processors and accelerators the implementation of brute force algorithms became more. The finite set of network elements (E_n) occupies the possible position points (P_k). The number of possible position points should be larger than the number of element ($k > n$). The D_n is the N-dimensional vector which is the solution of the problem and it belongs to the position points. F is the function finding distribution of the element ($F(d_n) = \min(\max)$). This problem allows the brute force algorithm to a multi-thread application. The optimization criteria should contradict each other (the accuracy of function should depends on coordination of antennas). This is the problem of multi-criteria optimization. In corresponding to the criteria function, brute force algorithm makes analyses and select the option for that and providing invariance of algorithm. The computation complexity is defined as $O(K^n)$, k^n is calculated for each points. The decision will be obtain as a set of local optimum variants where, by the fragmentation of the topology, the task of network topology design to a simplified to reduce the computation. The parallel platform algorithm is taken because the design of network topology is much complicated. To determine the location of element in wireless network is confirmed by using the brute force algorithm. This brute force algorithm is very effective and there is no need for any reorganization of application due to invariant in relating to criteria function.

The implementation through FFT and IFFT (inverse fast Fourier transform) to examine fast discrete convolution algorithm by HAYANAL. In some case, the combination of FFT and IFFT in fast convolution allows better freedom by selecting valid twiddle factor. By exploiting the freedom and use SAT solvers in order to find new fast convolution algorithm with very minimum count of operation. To find FFT algorithm within larger solution space the working of brute force search algorithm state of the art Boolean satisfiability (SAT) solvers. The future work employed on this algorithm SAT

based brute force search will explore enlarging the solution space, and also formulating search objectives other than reducing the operation count. The proposed SAT based brute force search is used to find FFT algorithm with lower flop count than the split-radix and higher flop count than tangent FFT, due to the constraint that the twiddle factor must be n th root of unity. In order to search selected instance than that will require fewer flops, in large FFT so we can cast the search as based on Boolean satisfiability (SAT) problem. By using satisfiability modulo theory (SMT) the integer arithmetic (mod n) accommodation will done. The search will take place by two modification to FFT search to be made first modification done in the bottom row node of the FFT (no cost).final multiplication undo any of the residual weight on base, these multiplication will not needed in the final fast convolution will equal, therefore FFT flow graphs are feasible to search. Finding fast convolution algorithms with the lowest known operation count is done by the brute force search techniques based on SAT. This ensures the constraints of the formula by the established bounds for the lowest possible FLOP count in the fast convolution.

The parameter search in a four dimensional space using an epipolar parametrization. With very simple and easily parallelizable computation we can do exhaustive search of parameter space by ENGVIST. A simple brute force algorithm which possess Robustness to outliers, No algorithmic degeneracies, cost function based on reprojection errors, not dependent on a good initialization, this ensures that the algorithm resulting in an effective method. The brute force search proposed method is used to estimate the relative orientation is to search rotation matrices R and R' for as many points correspondences as possible. The two rotation matrices is used to represent a relative orientation is an over parametrisation. The rotation s about the z axis is then R , R' and SR , SR' should describe the same relative orientation. For the given level of discretization and error threshold the maximum number of inliers is computed which has relative orientation. The other cost functions: inliers are correspondences with reprojection errors less than some prescribed threshold this will yield good result. In the motion restriction (planar motion, small motion) a few standards are restricted and one advantage of estimating relative orientation is that restricted motion and this can be handled very easily. The main role of motion segmentation is to estimate all these motions as well as the motion of the camera for the given sequence of moving object by the brute force algorithm. Further improve the motion segmentation by spatial prior assuming that close points probably belong to the same motion. Using Map Reduce model we can achieve parallelize the algorithm. CUDA is one of the best example for parallel implementation. Therefore the brute force algorithm approach becomes a viable and robust alternative for real-time visual odometry.

The brute force search algorithm is very essential optimal for the local string search problems and can be substantially be improved when it applied to classical NP hard string by GUO. In this we addressed two types of problem such that Closest string, Longest common subsequence string Closest string: in local search variant of the closest string, let A be the some arbitrary alphabet, and n be a positive integer. The input is a set $T \subseteq A^n$ of string d be the integer. The main aim is to find whether there is a string. The closest string can be solved by brute force algorithm in $O(n^{k+1}.m)$ time. Longest common subsequence problem need to find an input set T of a string S with some specified length L such that S is a subsequence of each string $t \in T$. The brute force algorithm solve this longest subsequence problem in $n^{O(k)}$ time. By considering the Hamming distance as a metric for defining the local neighbourhood where the search is performed, under this metric the brute force algorithm cannot be improved but other metrics are interestingly deserve important consideration. The local search algorithm can be substantially improved by applying brute force algorithm to the NP hard classical string problem. This local search can be applied by implementing brute force algorithm for many NP hard string problems. The optimal solution for local search is brute force algorithm.

The information security can be enhance by implementing hash algorithm for verification of digital signatures, key derivation, and random bit generation by RAVILLA, PUTTA. In this first the Zone routing protocol and hybrid MANET protocol is being implemented in NS2 and hashing algorithm and keyed hash message authentication code - secure hashing algorithm 512 is implemented for the authentication and data integrity of the information. Trust-based system is also used to prevent Denial-Of-Service attacks. The feature of this work will expand ZRP to SZRP so, this ensure the data confidentiality between source and destination. This method of work will lead the military operation into a very secure way by preventing it from eavesdropping. Digital signature is one of the authentication mechanism. It is created by taking the hash of the message and with the help of private key, the message is encrypted. The length of the hash code is very important against cryptanalysis such as brute force attack and also it should be a one way property (irreversible). where $H(x) = h$. SHA-512 is used to hash a message (M), with length(L), 0 less than or equal to L less 2128, each block has 1024 bits which denote the 64 bit words, the output of SHA512 is a 512 bit message digest. In SHA-512 the preprocessing will take care of padding the message, parsing message into message blocks, and setting initial hash value. Trusted based system is the cryptographic based system which ensures the additional security to the network by identifying the malicious code in the network and differentiate it from trusted nodes this done by setting timer while sending

data. The trust value of the node increase for all transmission and reduce for the nodes who not send data, their trust value is reduces for certain threshold, they are deemed malicious. The malicious node is broadcasted and kept isolated from the network (no service is offered). Trust Value = (Sum of '1' or '0')/Total Sent Packets By implementing two techniques namely HMAC-SHA512 for providing data integrity along with authentication and trusted based system to make secure network. The proposed protocol is used give to better result which is based on cryptography based algorithm.

The hash function is designed with new technology based on chaotic neural networks due to the properties of chaos and neural network, such as non-linearity, compression, confusion, and diffusion by ABDOUN, SAFWAN. The proposed system is not using simple chaotic maps, because the simple chaotic map is not robust. It integrates a strong chaotic generator into neurons. This proposed algorithm is very efficiency against strong collision resistant and high message sensitive when compared to SHA-2. It consist of 2 layers an input layer and output layer. Where k is the secret key, M is the input message and H is the hash value. The chaotic generator will supplies the parameter and condition. Thus the parameter and condition constitute the two layers by sub keys. The transfer function will have two chaotic maps such as Discrete Skew Tent Map and Discrete piecewise linear chaotic map which was connected in parallel. The future work will extend the algorithm to be against of cryptographic attacks such as Birthday attack, Collision attack, joux multi-collision, Long message, Second pre image, Herding, and Meet-in-the-middle attack. Thus we obtained the uniformity of hash values and the message sensitive, by a strong hash function therefore the proposed hash function is better then standard SHA-2. This hash function ensures the data integrity, digital signature, and authentication.

The proposed method is based on optimization of classifiers using quadratic probing by KUMAR. The time complexity for sequential search is $O(N)$, where N is the number of rule in a rule set. The search time can be reduced by reducing the quantity N . We found that same address and protocol fields are shared by different header fields, this succeeded by decompose the rule set and map the hash table. The numbers of entries in the classifiers are reduced on the basis of hash value generated by hash function. In the first step an appropriate hash function is selected to generate efficient hash values, this function is applied for each field value of rule-set in classifier. In case collision occurs, the counter is increased by one for the value and a new hash value is generated. This step is continued until collision is resolved or hash table is full. The procedure is susceptible to false negative value if hash table is small. But this situation is avoidable using appropriate size of hash table. With the advancement in

communication technology, gigabit networks are becoming more. The traditional processors do not have sufficient processing capabilities to handle and process packets arriving at such high speed. So in order to fulfil the need of high processing. This techniques has been used in future work by merging this with the tree based classifier and future the processing time is reduced .This method will considers the header fields repeating in the classifier only once so, that the size of the classifier table is not supported for incremental update and the size of the table is reduce significantly. This is tested under three classifier such that ACL rule-set with 916 rules, firewall rule-set with 791 rules, IPC rule set with 1500 rules. Therefore the three values 50,100,150 are used for the modulus operation performing on the header value of classifier. And the values of header and classifier is interchangeably used. Thus the complete packet preprocessing structure for effective scalable packet classification in ip forwarding is done by packet preprocessing scheme. This can also be used with high speed network.

The encrypted message is also not secure over the transmission so the security improved by verifying with digital signature, where the hashing algorithm is used to design the improvement by SHARMA, KOPPAD. SHA-3 algorithm is designed using verilog HDL and simulated in Xilinx ISE v14.2. SHA-3 is designed which has fixed output 512-bit, the improvement is done to increase the performance of frequency pipelining, which also includes Clock gated pipelined SHA-3. the combinational SHA-3 algorithm is designed with fixed output length 512 bits. Input at any size. Then input padding module will pad required numbers of zeros when sel signal for MUX is 0 then input bit connect to theta via MUX, this followed by xoring the input with round constant. The input of MUX and DMUX is controlled by FSM(finite state machine). This has three output mux_sel, dmux_sel, and counter_en and connected to selected line of MUX,DMUX. MUX, DMUX will be one then the output iota stage will fed back to theta stage for next round. This continues for 24 round. The pipeline is done in order to increase the performance of SHA-3.between the step mapping the register is inserted. 1st register is placed between theta and module, second in between rho and pi module, last is placed between chi and iota module. These registers are controlled by the input clock signal and clock enable signal. Clock gating is implemented in pipelined registers. The last based clock gating is applied to the pipelined SHA_3 the clock is controlled by by a clock signal. 0 when clock is not required and 1 when clock is required. This in turns prevents the glitches from propagation to clock.

The complexity is focused to show that the shortest path problem is N-P hard in either additive networks and directed cyclic networks where both models coincide by SINGH, KHOLI. This proposed work

provide a pseudo-polynomial time solution with nonnegative costs and gains. In networks with losses and gains there are two versions: a flow model and a path model. The shortest path problem is solvable in pseudo-polynomial time for nonnegative costs and gains by a dynamic programming approach. These NP-hardness results hold for the out-flow from the source, even for networks with integral capacities and with unit gain or with loss two for each arc, and for the in-flow into the sink, even for networks with unit loss or with unit gain for each arc. Moreover, the maximum flow problem is MAX-SNP-hard, and is hard to approximate. On the contrary, in unit-loss networks the maximum flow problem from the source can be solved efficiently by the Edmonds–Karp algorithm in $O(nm^2)$. Results reveal an essential difference between networks with additive and with (or without) multiplicative losses and gains. From the algorithmic point additive networks are much harder. Here the common flow problems are intractable whereas they are tractable in generalized networks with multiplicative losses and gains and in standard networks. The computation of the maximal flow in N from s to t uses the Edmonds-karp algorithm. This computes the flow augmenting paths according to their lengths, the length in sense number of arcs. Edmonds–Karp algorithm computes a maximal flow, and the flow is integral at every arc.

REFERENCES RÉFÉRENCES REFERENCIAS

1. “Maximum Independent Set Approximation Based on Bellman-Ford Algorithm” Mostafa H. Dahshan (2014), Springer, vol. no: 39, page no: 7003–7011.
2. “Mathematical formulations and exact algorithm for the multi trip cumulative capacitated single-vehicle routing problem” Juan Carlos Riveraa, b,*, H. Murat Afsara, Christian Prinsa (2015), Elsevier, vol. no: 249, page no 93-104.
3. “An Efficient Implementation of the Bellman -Ford Algorithm for Keller GPU Architectures” Federico Busato and Nicola Bombieri, Member, IEEE (2016), IEEE, VOL. 27, NO. 8, s.
4. “Parallel Implementation of Bellman-Ford Algorithm Using CUDA Architecture” Ganesh G Surve, Medha A shah (2017), IEEE, vol .no: 432, page no: 45-66.
5. “Efficient Implementation of the Bellman-ford algorithm for GPU” Marjan Nazarifard, Davoud Bahrepour (2017), IEEE, vol. no: 78, page no: 135-157.
6. “Demonstration of Single Link Failure Recovery using Bellman Ford and Dijil (stra Algorithm in SDN” Syed Waleed, Muhammad Faizan, MaheenIqbal, Muhammad Irfan Anis (2017), IEEE, vol. no: 97, page no: 765-789.
7. “A bag of paths framework for network data analysis” Kevin Francoisse, Ilkka Kivimaki, Amin Mantrach, Fabrice Rossi, Marco Saerens (2017), Elsevier, vol. no 90, pp. 90-111.
8. “To be investigating the graph cut the path instead of link. We also plan to evaluate experimental the potential distance as the distance between sequence of character by adapting it to a directed acyclic graph” Anggie Nastiti, Andrian Rakhmatsyah, Muhammad Arief Nugrohov (2018), IEEE, vol. no: 345, page no: 456-489.
9. “On the optimality of bellman ford Moore shortest path algorithm” Stasys Jukna, Georg Schnitger (2015), Elsevier, vol. no: 628, page no: 101-109.
10. “Face Image Abstraction by Ford-Fulkerson Algorithm and Invariant Feature Descriptor for Human Identification” Dakshina Ranjan, Kisku Debanjan Chatterjee, S. Trivedy, Massimo Tistarelli (2014), Springer, vol. no: 87, page no: 876-895.
11. “Hydraulically informed graph theoretic measure of link criticality for there sillience analysis of water distribution networks” Aly-Joy Ulusoy, Ivan Stoianov and Aurelie Chazerain (2018), Springer open, vol. no: 765, page no: 1457-1873.
12. “Chance distribution of the maximum flow of uncertain random network” Yuhong Sheng and Jinwu Gao (2014), Springer, vol.no:8, page no: 13-23.
13. “Minimax properties of some density measures in graphs and digraphs” Janet Anderson, Hong-Jian Lai, Xiaomin Li, Xiaoxia Lin & Murong Xu (2018), Taylor & Francis, vol. no; 3, page no: 1-12.
14. “Using basis dependence distance vectors in the modified Floyd–Warshall algorithm” Włodzimierz Bielecki · Krzysztof Kraska · Tomasz Klimek (2014), Springer, vol. no: 30, page no: 253–275.
15. “Probabilistic Calculation of Tolerances of the Dimension Chain Based on the Floyd-Warshall Algorithm” A. V. Muholzoeva, V. B. Masyagina, b (2016), Science direct, vol.no: 150, page no: 959-962.
16. “Accelerating Boyer Moore searches on binary texts”, Shmuel T. Klein, MiriKopel Ben-Nissan (2009), Springer, Vol No 410, Pg No. 3563-3571.
17. “A simple fast hybrid pattern-matching algorithm”, Frantisek Franek, Christopher G. Jennings, W. F. Smyth (2007), Science Direct, Vol No 5, Pg No 682–695.
18. “A Method for Web Application Vulnerabilities Detection by Using Boyer-Moore String Matching Algorithm”, Ain Zubaidah Mohd Saleh, Nur Amizah Rozali (2015), Science Direct, Vol No 72, Pg No 112-121.
19. “Analysis and Comparison of Adaptive Huffman Coding and Arithmetic Coding algorithms”, Phyla Mbewe, Sampson D. Asare (2017), IEEE, Vol No 978, Pg No 178-185.
20. “High performance parallelization of Boyer–Moore algorithm on many-core accelerators”, Yosang

- Jeong, Myungho Lee (2015), Springer, Vol No 18, Pg No 1087-1098.
21. "Balancing decoding speed and memory usage for Huffman codes using quaternary tree", Ahsan Habib, Mohammad Shahidur Rahman (2017), Springer, Vol No 4:5, Pg No 1-15.
 22. "A variation on the Boyer-Moore algorithm", Thierry Lecroq (1994), Science Direct, Vol No 92, Pg No 119-144.
 23. "A fast algorithm for Huffman decoding based on a recursion Huffman tree", Yih-Kai Lin, Shu-Chien Huang, Cheng-Hsing Yang (2011), Science Direct, Vol No 85, Pg No 974-980.
 24. "A Boyer-Moore-style algorithm for regular expression pattern matching", Bruce W. Watson (2002), Science Direct, Vol No 48, Pg No 99-117.
 25. "A Fast-Adaptive Huffman Coding Algorithm", Wei-Wei Lu and M. P. Gough (1998), IEEE, Vol No 41, Pg No 535-538.
 26. "An algorithm For Generating n Huffman Codes with a Worst Case Of $(n \log, n + n \log, \log, n)$ Comparisons", Ghim Hwee Ong (2000), IEEE, Vol No 32, Pg No 1-5.
 27. "A memory Effective Adaptive Huffman Coding Algorithm", Steven Pigeon, Yoshua Bengio (1993), Springer, Vol No 21, Pg No 234-256.
 28. "A Quick algorithm for planar fragmented objects stitching Base on KMP algorithm", Wang Zhe, Chen Jun, Yuan Gang Zhao Zhou Yan (2001), Science Direct, Vol No 467, Pg No 23- 29.
 29. "Application of KMP Algorithm in Customized Flow Analysis", Qingzhu Meng, Zhenming Lei, Dazhong He (2017), IEEE, Vol No 45, Pg No 2338- 2342.
 30. "R. Yavne "An economical method of calculating the discrete fourier transform," in proc. AFIPS fall joint computer conf. ACM, 1968, vol. 33, pp.115-125.
 31. D. Bernstein, "The tangent FFT," Applied Algebra, Algebraic Algorithm and Error-Correction Codes, pp-291-300, 2007.
 32. Dilla Ravilla, and Chandra sekara reddy putta "Enhancing the security of mannets using hash algorithm", Computer Science 54 (2015) 196 – 206.
 33. Nabil Abdoun, Safwan El Assad, "Secure hash algorithm based on efficient chaotic neural network", 978-1-4673-8197-0/16/2016 IEEE.
 34. R. Jain, "I want my IPTV," IEEE Multimedia, vol. 12, no. 3, Jul. 2005.
 35. H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable videocoding extension of the H.264/AVC standard," IEEE Transactions on Circuits and Systems for Video Technology, Special Issue on Scalable Video Coding, vol. 17, no. 9, Sep. 2007.
 36. R. Ahlswede, N. Cai, S.-Y. Li, and R. Yeung, "Network information flow, "IEEE Transactions on Information Theory, vol. 46, no. 4, pp. 1204–1216, July 2000.
 37. Alia Arshad, Dur-e-Shahwarkundi, Arshad Aziz, "Compact Implementation of SHA3-512 on FPGA", IEEE Conf. on Information Assurance and Cyber Security (CIACS), Rawalpindi, Pakistan, pp. 29-33, 12-13 June, 2014.
 38. George S. Athanasiou, George-Paris Makkas, GeorgiosTheodoridis,"High throughput pipelined FPGA implementation of the new SHA-3cryptographic hash algorithm", 6th IEEE Int. Symp. On Communications, Control and Signal Processing (ISCCSP), Athens, Greece, pp. 538-541, 21-23 May 2014.
 39. Lenosloannou, Harris E. Michail, Artemios G. Voyiatzis, "High Performance Pipelined FPGA Implementation of the SHA-3 Hash Algorithm", 4th IEEE Mediterranean Conf. on Embedded Computing (MECO), Budva, Montenegro, pp.68-71, 14-18 June 2015.
 40. MuzaffarRao, Thomas Newe and Ian Grout, "Secure Hash Algorithm-3(SHA-3) implementation on Xilinx FPGAs, Suitable for IOT Applications", IEEE Int. Conf. on Computer and Information Technology, Ubiquitous Computing and Communications, Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), Liverpool, England, pp. 2212-2216, 26-28 Oct. 2015.

GLOBAL JOURNALS GUIDELINES HANDBOOK 2019

WWW.GLOBALJOURNALS.ORG

FELLOWS

FELLOW OF ASSOCIATION OF RESEARCH SOCIETY IN COMPUTING (FARSC)

Global Journals Incorporate (USA) is accredited by Open Association of Research Society (OARS), U.S.A and in turn, awards “FARSC” title to individuals. The 'FARSC' title is accorded to a selected professional after the approval of the Editor-in-Chief/Editorial Board Members/Dean.



- The “FARSC” is a dignified title which is accorded to a person’s name viz. Dr. John E. Hall, Ph.D., FARSC or William Walldroff, M.S., FARSC.

FARSC accrediting is an honor. It authenticates your research activities. After recognition as FARSC, you can add 'FARSC' title with your name as you use this recognition as additional suffix to your status. This will definitely enhance and add more value and repute to your name. You may use it on your professional Counseling Materials such as CV, Resume, and Visiting Card etc.

The following benefits can be availed by you only for next three years from the date of certification:



FARSC designated members are entitled to avail a 40% discount while publishing their research papers (of a single author) with Global Journals Incorporation (USA), if the same is accepted by Editorial Board/Peer Reviewers. If you are a main author or co-author in case of multiple authors, you will be entitled to avail discount of 10%.

Once FARSC title is accorded, the Fellow is authorized to organize a symposium/seminar/conference on behalf of Global Journal Incorporation (USA). The Fellow can also participate in conference/seminar/symposium organized by another institution as representative of Global Journal. In both the cases, it is mandatory for him to discuss with us and obtain our consent.



You may join as member of the Editorial Board of Global Journals Incorporation (USA) after successful completion of three years as Fellow and as Peer Reviewer. In addition, it is also desirable that you should organize seminar/symposium/conference at least once.

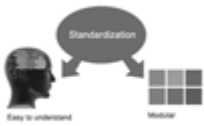
We shall provide you intimation regarding launching of e-version of journal of your stream time to time. This may be utilized in your library for the enrichment of knowledge of your students as well as it can also be helpful for the concerned faculty members.





The FARSS can go through standards of OARS. You can also play vital role if you have any suggestions so that proper amendment can take place to improve the same for the benefit of entire research community.

As FARSS, you will be given a renowned, secure and free professional email address with 100 GB of space e.g. johnhall@globaljournals.org. This will include Webmail, Spam Assassin, Email Forwarders, Auto-Responders, Email Delivery Route tracing, etc.



The FARSS will be eligible for a free application of standardization of their researches. Standardization of research will be subject to acceptability within stipulated norms as the next step after publishing in a journal. We shall depute a team of specialized research professionals who will render their services for elevating your researches to next higher level, which is worldwide open standardization.

The FARSS member can apply for grading and certification of standards of their educational and Institutional Degrees to Open Association of Research, Society U.S.A. Once you are designated as FARSS, you may send us a scanned copy of all of your credentials. OARS will verify, grade and certify them. This will be based on your academic records, quality of research papers published by you, and some more criteria. After certification of all your credentials by OARS, they will be published on your Fellow Profile link on website <https://associationofresearch.org> which will be helpful to upgrade the dignity.



The FARSS members can avail the benefits of free research podcasting in Global Research Radio with their research documents. After publishing the work, (including published elsewhere worldwide with proper authorization) you can upload your research paper with your recorded voice or you can utilize

chargeable services of our professional RJs to record your paper in their voice on request.



The FARSS member also entitled to get the benefits of free research podcasting of their research documents through video clips. We can also streamline your conference videos and display your slides/ online slides and online research video clips at reasonable charges, on request.





The FARSS is eligible to earn from sales proceeds of his/her researches/reference/review Books or literature, while publishing with Global Journals. The FARSS can decide whether he/she would like to publish his/her research in a closed manner. In this case, whenever readers purchase that individual research paper for reading, maximum 60% of its profit earned as royalty by Global Journals, will be credited to his/her bank account. The entire entitled amount will be credited to his/her bank account exceeding limit of minimum fixed balance. There is no minimum time limit for collection. The FARSS member can decide its price and we can help in making the right decision.

The FARSS member is eligible to join as a paid peer reviewer at Global Journals Incorporation (USA) and can get remuneration of 15% of author fees, taken from the author of a respective paper. After reviewing 5 or more papers you can request to transfer the amount to your bank account.



MEMBER OF ASSOCIATION OF RESEARCH SOCIETY IN SCIENCE (MARSS)

The ' MARSS ' title is accorded to a selected professional after the approval of the Editor-in-Chief / Editorial Board Members/Dean.

The “MARSS” is a dignified ornament which is accorded to a person’s name viz. Dr. John E. Hall, Ph.D., MARSS or William Walldroff, M.S., MARSS.



MARSS accrediting is an honor. It authenticates your research activities. After becoming MARSS, you can add 'MARSS' title with your name as you use this recognition as additional suffix to your status. This will definitely enhance and add more value and repute to your name. You may use it on your professional Counseling Materials such as CV, Resume, Visiting Card and Name Plate etc.

The following benefits can be availed by you only for next three years from the date of certification.



MARSS designated members are entitled to avail a 25% discount while publishing their research papers (of a single author) in Global Journals Inc., if the same is accepted by our Editorial Board and Peer Reviewers. If you are a main author or co-author of a group of authors, you will get discount of 10%.

As MARSS, you will be given a renowned, secure and free professional email address with 30 GB of space e.g. johnhall@globaljournals.org. This will include Webmail, Spam Assassin, Email Forwarders, Auto-Responders, Email Delivery Route tracing, etc.





We shall provide you intimation regarding launching of e-version of journal of your stream time to time. This may be utilized in your library for the enrichment of knowledge of your students as well as it can also be helpful for the concerned faculty members.

The MARSC member can apply for approval, grading and certification of standards of their educational and Institutional Degrees to Open Association of Research, Society U.S.A.



Once you are designated as MARSC, you may send us a scanned copy of all of your credentials. OARS will verify, grade and certify them. This will be based on your academic records, quality of research papers published by you, and some more criteria.

It is mandatory to read all terms and conditions carefully.



AUXILIARY MEMBERSHIPS

Institutional Fellow of Open Association of Research Society (USA)-OARS (USA)

Global Journals Incorporation (USA) is accredited by Open Association of Research Society, U.S.A (OARS) and in turn, affiliates research institutions as “Institutional Fellow of Open Association of Research Society” (IFOARS).



The “FARSC” is a dignified title which is accorded to a person’s name viz. Dr. John E. Hall, Ph.D., FARSC or William Walldroff, M.S., FARSC.

The IFOARS institution is entitled to form a Board comprised of one Chairperson and three to five board members preferably from different streams. The Board will be recognized as “Institutional Board of Open Association of Research Society”-(IBOARS).

The Institute will be entitled to following benefits:



The IBOARS can initially review research papers of their institute and recommend them to publish with respective journal of Global Journals. It can also review the papers of other institutions after obtaining our consent. The second review will be done by peer reviewer of Global Journals Incorporation (USA) The Board is at liberty to appoint a peer reviewer with the approval of chairperson after consulting us.

The author fees of such paper may be waived off up to 40%.

The Global Journals Incorporation (USA) at its discretion can also refer double blind peer reviewed paper at their end to the board for the verification and to get recommendation for final stage of acceptance of publication.



The IBOARS can organize symposium/seminar/conference in their country on behalf of Global Journals Incorporation (USA)-OARS (USA). The terms and conditions can be discussed separately.

The Board can also play vital role by exploring and giving valuable suggestions regarding the Standards of “Open Association of Research Society, U.S.A (OARS)” so that proper amendment can take place for the benefit of entire research community. We shall provide details of particular standard only on receipt of request from the Board.



Journals Research
inducing researches

The board members can also join us as Individual Fellow with 40% discount on total fees applicable to Individual Fellow. They will be entitled to avail all the benefits as declared. Please visit Individual Fellow-sub menu of GlobalJournals.org to have more relevant details.



We shall provide you intimation regarding launching of e-version of journal of your stream time to time. This may be utilized in your library for the enrichment of knowledge of your students as well as it can also be helpful for the concerned faculty members.



After nomination of your institution as “Institutional Fellow” and constantly functioning successfully for one year, we can consider giving recognition to your institute to function as Regional/Zonal office on our behalf.

The board can also take up the additional allied activities for betterment after our consultation.

The following entitlements are applicable to individual Fellows:

Open Association of Research Society, U.S.A (OARS) By-laws states that an individual Fellow may use the designations as applicable, or the corresponding initials. The Credentials of individual Fellow and Associate designations signify that the individual has gained knowledge of the fundamental concepts. One is magnanimous and proficient in an expertise course covering the professional code of conduct, and follows recognized standards of practice.



Open Association of Research Society (US)/ Global Journals Incorporation (USA), as described in Corporate Statements, are educational, research publishing and professional membership organizations. Achieving our individual Fellow or Associate status is based mainly on meeting stated educational research requirements.

Disbursement of 40% Royalty earned through Global Journals : Researcher = 50%, Peer Reviewer = 37.50%, Institution = 12.50% E.g. Out of 40%, the 20% benefit should be passed on to researcher, 15 % benefit towards remuneration should be given to a reviewer and remaining 5% is to be retained by the institution.



We shall provide print version of 12 issues of any three journals [as per your requirement] out of our 38 journals worth \$ 2376 USD.

Other:

The individual Fellow and Associate designations accredited by Open Association of Research Society (US) credentials signify guarantees following achievements:

- The professional accredited with Fellow honor, is entitled to various benefits viz. name, fame, honor, regular flow of income, secured bright future, social status etc.



- In addition to above, if one is single author, then entitled to 40% discount on publishing research paper and can get 10% discount if one is co-author or main author among group of authors.
- The Fellow can organize symposium/seminar/conference on behalf of Global Journals Incorporation (USA) and he/she can also attend the same organized by other institutes on behalf of Global Journals.
- The Fellow can become member of Editorial Board Member after completing 3yrs.
- The Fellow can earn 60% of sales proceeds from the sale of reference/review books/literature/publishing of research paper.
- Fellow can also join as paid peer reviewer and earn 15% remuneration of author charges and can also get an opportunity to join as member of the Editorial Board of Global Journals Incorporation (USA)
- • This individual has learned the basic methods of applying those concepts and techniques to common challenging situations. This individual has further demonstrated an in-depth understanding of the application of suitable techniques to a particular area of research practice.

Note :

“

- In future, if the board feels the necessity to change any board member, the same can be done with the consent of the chairperson along with anyone board member without our approval.
- In case, the chairperson needs to be replaced then consent of 2/3rd board members are required and they are also required to jointly pass the resolution copy of which should be sent to us. In such case, it will be compulsory to obtain our approval before replacement.
- In case of “Difference of Opinion [if any]” among the Board members, our decision will be final and binding to everyone.

”



PREFERRED AUTHOR GUIDELINES

We accept the manuscript submissions in any standard (generic) format.

We typeset manuscripts using advanced typesetting tools like Adobe In Design, CorelDraw, TeXnicCenter, and TeXStudio. We usually recommend authors submit their research using any standard format they are comfortable with, and let Global Journals do the rest.

Alternatively, you can download our basic template from <https://globaljournals.org/Template.zip>

Authors should submit their complete paper/article, including text illustrations, graphics, conclusions, artwork, and tables. Authors who are not able to submit manuscript using the form above can email the manuscript department at submit@globaljournals.org or get in touch with chiefeditor@globaljournals.org if they wish to send the abstract before submission.

BEFORE AND DURING SUBMISSION

Authors must ensure the information provided during the submission of a paper is authentic. Please go through the following checklist before submitting:

1. Authors must go through the complete author guideline and understand and *agree to Global Journals' ethics and code of conduct*, along with author responsibilities.
2. Authors must accept the privacy policy, terms, and conditions of Global Journals.
3. Ensure corresponding author's email address and postal address are accurate and reachable.
4. Manuscript to be submitted must include keywords, an abstract, a paper title, co-author(s) names and details (email address, name, phone number, and institution), figures and illustrations in vector format including appropriate captions, tables, including titles and footnotes, a conclusion, results, acknowledgments and references.
5. Authors should submit paper in a ZIP archive if any supplementary files are required along with the paper.
6. Proper permissions must be acquired for the use of any copyrighted material.
7. Manuscript submitted *must not have been submitted or published elsewhere* and all authors must be aware of the submission.

Declaration of Conflicts of Interest

It is required for authors to declare all financial, institutional, and personal relationships with other individuals and organizations that could influence (bias) their research.

POLICY ON PLAGIARISM

Plagiarism is not acceptable in Global Journals submissions at all.

Plagiarized content will not be considered for publication. We reserve the right to inform authors' institutions about plagiarism detected either before or after publication. If plagiarism is identified, we will follow COPE guidelines:

Authors are solely responsible for all the plagiarism that is found. The author must not fabricate, falsify or plagiarize existing research data. The following, if copied, will be considered plagiarism:

- Words (language)
- Ideas
- Findings
- Writings
- Diagrams
- Graphs
- Illustrations
- Lectures



- Printed material
- Graphic representations
- Computer programs
- Electronic material
- Any other original work

AUTHORSHIP POLICIES

Global Journals follows the definition of authorship set up by the Open Association of Research Society, USA. According to its guidelines, authorship criteria must be based on:

1. Substantial contributions to the conception and acquisition of data, analysis, and interpretation of findings.
2. Drafting the paper and revising it critically regarding important academic content.
3. Final approval of the version of the paper to be published.

Changes in Authorship

The corresponding author should mention the name and complete details of all co-authors during submission and in manuscript. We support addition, rearrangement, manipulation, and deletions in authors list till the early view publication of the journal. We expect that corresponding author will notify all co-authors of submission. We follow COPE guidelines for changes in authorship.

Copyright

During submission of the manuscript, the author is confirming an exclusive license agreement with Global Journals which gives Global Journals the authority to reproduce, reuse, and republish authors' research. We also believe in flexible copyright terms where copyright may remain with authors/employers/institutions as well. Contact your editor after acceptance to choose your copyright policy. You may follow this form for copyright transfers.

Appealing Decisions

Unless specified in the notification, the Editorial Board's decision on publication of the paper is final and cannot be appealed before making the major change in the manuscript.

Acknowledgments

Contributors to the research other than authors credited should be mentioned in Acknowledgments. The source of funding for the research can be included. Suppliers of resources may be mentioned along with their addresses.

Declaration of funding sources

Global Journals is in partnership with various universities, laboratories, and other institutions worldwide in the research domain. Authors are requested to disclose their source of funding during every stage of their research, such as making analysis, performing laboratory operations, computing data, and using institutional resources, from writing an article to its submission. This will also help authors to get reimbursements by requesting an open access publication letter from Global Journals and submitting to the respective funding source.

PREPARING YOUR MANUSCRIPT

Authors can submit papers and articles in an acceptable file format: MS Word (doc, docx), LaTeX (.tex, .zip or .rar including all of your files), Adobe PDF (.pdf), rich text format (.rtf), simple text document (.txt), Open Document Text (.odt), and Apple Pages (.pages). Our professional layout editors will format the entire paper according to our official guidelines. This is one of the highlights of publishing with Global Journals—authors should not be concerned about the formatting of their paper. Global Journals accepts articles and manuscripts in every major language, be it Spanish, Chinese, Japanese, Portuguese, Russian, French, German, Dutch, Italian, Greek, or any other national language, but the title, subtitle, and abstract should be in English. This will facilitate indexing and the pre-peer review process.

The following is the official style and template developed for publication of a research paper. Authors are not required to follow this style during the submission of the paper. It is just for reference purposes.



Manuscript Style Instruction (Optional)

- Microsoft Word Document Setting Instructions.
- Font type of all text should be Swis721 Lt BT.
- Page size: 8.27" x 11", left margin: 0.65, right margin: 0.65, bottom margin: 0.75.
- Paper title should be in one column of font size 24.
- Author name in font size of 11 in one column.
- Abstract: font size 9 with the word "Abstract" in bold italics.
- Main text: font size 10 with two justified columns.
- Two columns with equal column width of 3.38 and spacing of 0.2.
- First character must be three lines drop-capped.
- The paragraph before spacing of 1 pt and after of 0 pt.
- Line spacing of 1 pt.
- Large images must be in one column.
- The names of first main headings (Heading 1) must be in Roman font, capital letters, and font size of 10.
- The names of second main headings (Heading 2) must not include numbers and must be in italics with a font size of 10.

Structure and Format of Manuscript

The recommended size of an original research paper is under 15,000 words and review papers under 7,000 words. Research articles should be less than 10,000 words. Research papers are usually longer than review papers. Review papers are reports of significant research (typically less than 7,000 words, including tables, figures, and references)

A research paper must include:

- a) A title which should be relevant to the theme of the paper.
- b) A summary, known as an abstract (less than 150 words), containing the major results and conclusions.
- c) Up to 10 keywords that precisely identify the paper's subject, purpose, and focus.
- d) An introduction, giving fundamental background objectives.
- e) Resources and techniques with sufficient complete experimental details (wherever possible by reference) to permit repetition, sources of information must be given, and numerical methods must be specified by reference.
- f) Results which should be presented concisely by well-designed tables and figures.
- g) Suitable statistical data should also be given.
- h) All data must have been gathered with attention to numerical detail in the planning stage.

Design has been recognized to be essential to experiments for a considerable time, and the editor has decided that any paper that appears not to have adequate numerical treatments of the data will be returned unrefereed.

- i) Discussion should cover implications and consequences and not just recapitulate the results; conclusions should also be summarized.
- j) There should be brief acknowledgments.
- k) There ought to be references in the conventional format. Global Journals recommends APA format.

Authors should carefully consider the preparation of papers to ensure that they communicate effectively. Papers are much more likely to be accepted if they are carefully designed and laid out, contain few or no errors, are summarizing, and follow instructions. They will also be published with much fewer delays than those that require much technical and editorial correction.

The Editorial Board reserves the right to make literary corrections and suggestions to improve brevity.



FORMAT STRUCTURE

It is necessary that authors take care in submitting a manuscript that is written in simple language and adheres to published guidelines.

All manuscripts submitted to Global Journals should include:

Title

The title page must carry an informative title that reflects the content, a running title (less than 45 characters together with spaces), names of the authors and co-authors, and the place(s) where the work was carried out.

Author details

The full postal address of any related author(s) must be specified.

Abstract

The abstract is the foundation of the research paper. It should be clear and concise and must contain the objective of the paper and inferences drawn. It is advised to not include big mathematical equations or complicated jargon.

Many researchers searching for information online will use search engines such as Google, Yahoo or others. By optimizing your paper for search engines, you will amplify the chance of someone finding it. In turn, this will make it more likely to be viewed and cited in further works. Global Journals has compiled these guidelines to facilitate you to maximize the web-friendliness of the most public part of your paper.

Keywords

A major lynchpin of research work for the writing of research papers is the keyword search, which one will employ to find both library and internet resources. Up to eleven keywords or very brief phrases have to be given to help data retrieval, mining, and indexing.

One must be persistent and creative in using keywords. An effective keyword search requires a strategy: planning of a list of possible keywords and phrases to try.

Choice of the main keywords is the first tool of writing a research paper. Research paper writing is an art. Keyword search should be as strategic as possible.

One should start brainstorming lists of potential keywords before even beginning searching. Think about the most important concepts related to research work. Ask, "What words would a source have to include to be truly valuable in a research paper?" Then consider synonyms for the important words.

It may take the discovery of only one important paper to steer in the right keyword direction because, in most databases, the keywords under which a research paper is abstracted are listed with the paper.

Numerical Methods

Numerical methods used should be transparent and, where appropriate, supported by references.

Abbreviations

Authors must list all the abbreviations used in the paper at the end of the paper or in a separate table before using them.

Formulas and equations

Authors are advised to submit any mathematical equation using either MathJax, KaTeX, or LaTeX, or in a very high-quality image.

Tables, Figures, and Figure Legends

Tables: Tables should be cautiously designed, uncrowned, and include only essential data. Each must have an Arabic number, e.g., Table 4, a self-explanatory caption, and be on a separate sheet. Authors must submit tables in an editable format and not as images. References to these tables (if any) must be mentioned accurately.



Figures

Figures are supposed to be submitted as separate files. Always include a citation in the text for each figure using Arabic numbers, e.g., Fig. 4. Artwork must be submitted online in vector electronic form or by emailing it.

PREPARATION OF ELETRONIC FIGURES FOR PUBLICATION

Although low-quality images are sufficient for review purposes, print publication requires high-quality images to prevent the final product being blurred or fuzzy. Submit (possibly by e-mail) EPS (line art) or TIFF (halftone/ photographs) files only. MS PowerPoint and Word Graphics are unsuitable for printed pictures. Avoid using pixel-oriented software. Scans (TIFF only) should have a resolution of at least 350 dpi (halftone) or 700 to 1100 dpi (line drawings). Please give the data for figures in black and white or submit a Color Work Agreement form. EPS files must be saved with fonts embedded (and with a TIFF preview, if possible).

For scanned images, the scanning resolution at final image size ought to be as follows to ensure good reproduction: line art: >650 dpi; halftones (including gel photographs): >350 dpi; figures containing both halftone and line images: >650 dpi.

Color charges: Authors are advised to pay the full cost for the reproduction of their color artwork. Hence, please note that if there is color artwork in your manuscript when it is accepted for publication, we would require you to complete and return a Color Work Agreement form before your paper can be published. Also, you can email your editor to remove the color fee after acceptance of the paper.

TIPS FOR WRITING A GOOD QUALITY COMPUTER SCIENCE RESEARCH PAPER

Techniques for writing a good quality computer science research paper:

1. Choosing the topic: In most cases, the topic is selected by the interests of the author, but it can also be suggested by the guides. You can have several topics, and then judge which you are most comfortable with. This may be done by asking several questions of yourself, like "Will I be able to carry out a search in this area? Will I find all necessary resources to accomplish the search? Will I be able to find all information in this field area?" If the answer to this type of question is "yes," then you ought to choose that topic. In most cases, you may have to conduct surveys and visit several places. Also, you might have to do a lot of work to find all the rises and falls of the various data on that subject. Sometimes, detailed information plays a vital role, instead of short information. Evaluators are human: The first thing to remember is that evaluators are also human beings. They are not only meant for rejecting a paper. They are here to evaluate your paper. So present your best aspect.

2. Think like evaluators: If you are in confusion or getting demotivated because your paper may not be accepted by the evaluators, then think, and try to evaluate your paper like an evaluator. Try to understand what an evaluator wants in your research paper, and you will automatically have your answer. Make blueprints of paper: The outline is the plan or framework that will help you to arrange your thoughts. It will make your paper logical. But remember that all points of your outline must be related to the topic you have chosen.

3. Ask your guides: If you are having any difficulty with your research, then do not hesitate to share your difficulty with your guide (if you have one). They will surely help you out and resolve your doubts. If you can't clarify what exactly you require for your work, then ask your supervisor to help you with an alternative. He or she might also provide you with a list of essential readings.

4. Use of computer is recommended: As you are doing research in the field of computer science then this point is quite obvious. Use right software: Always use good quality software packages. If you are not capable of judging good software, then you can lose the quality of your paper unknowingly. There are various programs available to help you which you can get through the internet.

5. Use the internet for help: An excellent start for your paper is using Google. It is a wondrous search engine, where you can have your doubts resolved. You may also read some answers for the frequent question of how to write your research paper or find a model research paper. You can download books from the internet. If you have all the required books, place importance on reading, selecting, and analyzing the specified information. Then sketch out your research paper. Use big pictures: You may use encyclopedias like Wikipedia to get pictures with the best resolution. At Global Journals, you should strictly follow here.



6. Bookmarks are useful: When you read any book or magazine, you generally use bookmarks, right? It is a good habit which helps to not lose your continuity. You should always use bookmarks while searching on the internet also, which will make your search easier.

7. Revise what you wrote: When you write anything, always read it, summarize it, and then finalize it.

8. Make every effort: Make every effort to mention what you are going to write in your paper. That means always have a good start. Try to mention everything in the introduction—what is the need for a particular research paper. Polish your work with good writing skills and always give an evaluator what he wants. Make backups: When you are going to do any important thing like making a research paper, you should always have backup copies of it either on your computer or on paper. This protects you from losing any portion of your important data.

9. Produce good diagrams of your own: Always try to include good charts or diagrams in your paper to improve quality. Using several unnecessary diagrams will degrade the quality of your paper by creating a hodgepodge. So always try to include diagrams which were made by you to improve the readability of your paper. Use of direct quotes: When you do research relevant to literature, history, or current affairs, then use of quotes becomes essential, but if the study is relevant to science, use of quotes is not preferable.

10. Use proper verb tense: Use proper verb tenses in your paper. Use past tense to present those events that have happened. Use present tense to indicate events that are going on. Use future tense to indicate events that will happen in the future. Use of wrong tenses will confuse the evaluator. Avoid sentences that are incomplete.

11. Pick a good study spot: Always try to pick a spot for your research which is quiet. Not every spot is good for studying.

12. Know what you know: Always try to know what you know by making objectives, otherwise you will be confused and unable to achieve your target.

13. Use good grammar: Always use good grammar and words that will have a positive impact on the evaluator; use of good vocabulary does not mean using tough words which the evaluator has to find in a dictionary. Do not fragment sentences. Eliminate one-word sentences. Do not ever use a big word when a smaller one would suffice.

Verbs have to be in agreement with their subjects. In a research paper, do not start sentences with conjunctions or finish them with prepositions. When writing formally, it is advisable to never split an infinitive because someone will (wrongly) complain. Avoid clichés like a disease. Always shun irritating alliteration. Use language which is simple and straightforward. Put together a neat summary.

14. Arrangement of information: Each section of the main body should start with an opening sentence, and there should be a changeover at the end of the section. Give only valid and powerful arguments for your topic. You may also maintain your arguments with records.

15. Never start at the last minute: Always allow enough time for research work. Leaving everything to the last minute will degrade your paper and spoil your work.

16. Multitasking in research is not good: Doing several things at the same time is a bad habit in the case of research activity. Research is an area where everything has a particular time slot. Divide your research work into parts, and do a particular part in a particular time slot.

17. Never copy others' work: Never copy others' work and give it your name because if the evaluator has seen it anywhere, you will be in trouble. Take proper rest and food: No matter how many hours you spend on your research activity, if you are not taking care of your health, then all your efforts will have been in vain. For quality research, take proper rest and food.

18. Go to seminars: Attend seminars if the topic is relevant to your research area. Utilize all your resources.

19. Refresh your mind after intervals: Try to give your mind a rest by listening to soft music or sleeping in intervals. This will also improve your memory. Acquire colleagues: Always try to acquire colleagues. No matter how sharp you are, if you acquire colleagues, they can give you ideas which will be helpful to your research.



20. Think technically: Always think technically. If anything happens, search for its reasons, benefits, and demerits. Think and then print: When you go to print your paper, check that tables are not split, headings are not detached from their descriptions, and page sequence is maintained.

21. Adding unnecessary information: Do not add unnecessary information like "I have used MS Excel to draw graphs." Irrelevant and inappropriate material is superfluous. Foreign terminology and phrases are not apropos. One should never take a broad view. Analogy is like feathers on a snake. Use words properly, regardless of how others use them. Remove quotations. Puns are for kids, not grunt readers. Never oversimplify: When adding material to your research paper, never go for oversimplification; this will definitely irritate the evaluator. Be specific. Never use rhythmic redundancies. Contractions shouldn't be used in a research paper. Comparisons are as terrible as clichés. Give up ampersands, abbreviations, and so on. Remove commas that are not necessary. Parenthetical words should be between brackets or commas. Understatement is always the best way to put forward earth-shaking thoughts. Give a detailed literary review.

22. Report concluded results: Use concluded results. From raw data, filter the results, and then conclude your studies based on measurements and observations taken. An appropriate number of decimal places should be used. Parenthetical remarks are prohibited here. Proofread carefully at the final stage. At the end, give an outline to your arguments. Spot perspectives of further study of the subject. Justify your conclusion at the bottom sufficiently, which will probably include examples.

23. Upon conclusion: Once you have concluded your research, the next most important step is to present your findings. Presentation is extremely important as it is the definite medium through which your research is going to be in print for the rest of the crowd. Care should be taken to categorize your thoughts well and present them in a logical and neat manner. A good quality research paper format is essential because it serves to highlight your research paper and bring to light all necessary aspects of your research.

INFORMAL GUIDELINES OF RESEARCH PAPER WRITING

Key points to remember:

- Submit all work in its final form.
- Write your paper in the form which is presented in the guidelines using the template.
- Please note the criteria peer reviewers will use for grading the final paper.

Final points:

One purpose of organizing a research paper is to let people interpret your efforts selectively. The journal requires the following sections, submitted in the order listed, with each section starting on a new page:

The introduction: This will be compiled from reference matter and reflect the design processes or outline of basis that directed you to make a study. As you carry out the process of study, the method and process section will be constructed like that. The results segment will show related statistics in nearly sequential order and direct reviewers to similar intellectual paths throughout the data that you gathered to carry out your study.

The discussion section:

This will provide understanding of the data and projections as to the implications of the results. The use of good quality references throughout the paper will give the effort trustworthiness by representing an alertness to prior workings.

Writing a research paper is not an easy job, no matter how trouble-free the actual research or concept. Practice, excellent preparation, and controlled record-keeping are the only means to make straightforward progression.

General style:

Specific editorial column necessities for compliance of a manuscript will always take over from directions in these general guidelines.

To make a paper clear: Adhere to recommended page limits.



Mistakes to avoid:

- Insertion of a title at the foot of a page with subsequent text on the next page.
- Separating a table, chart, or figure—confine each to a single page.
- Submitting a manuscript with pages out of sequence.
- In every section of your document, use standard writing style, including articles ("a" and "the").
- Keep paying attention to the topic of the paper.
- Use paragraphs to split each significant point (excluding the abstract).
- Align the primary line of each section.
- Present your points in sound order.
- Use present tense to report well-accepted matters.
- Use past tense to describe specific results.
- Do not use familiar wording; don't address the reviewer directly. Don't use slang or superlatives.
- Avoid use of extra pictures—include only those figures essential to presenting results.

Title page:

Choose a revealing title. It should be short and include the name(s) and address(es) of all authors. It should not have acronyms or abbreviations or exceed two printed lines.

Abstract: This summary should be two hundred words or less. It should clearly and briefly explain the key findings reported in the manuscript and must have precise statistics. It should not have acronyms or abbreviations. It should be logical in itself. Do not cite references at this point.

An abstract is a brief, distinct paragraph summary of finished work or work in development. In a minute or less, a reviewer can be taught the foundation behind the study, common approaches to the problem, relevant results, and significant conclusions or new questions.

Write your summary when your paper is completed because how can you write the summary of anything which is not yet written? Wealth of terminology is very essential in abstract. Use comprehensive sentences, and do not sacrifice readability for brevity; you can maintain it succinctly by phrasing sentences so that they provide more than a lone rationale. The author can at this moment go straight to shortening the outcome. Sum up the study with the subsequent elements in any summary. Try to limit the initial two items to no more than one line each.

Reason for writing the article—theory, overall issue, purpose.

- Fundamental goal.
- To-the-point depiction of the research.
- Consequences, including definite statistics—if the consequences are quantitative in nature, account for this; results of any numerical analysis should be reported. Significant conclusions or questions that emerge from the research.

Approach:

- Single section and succinct.
- An outline of the job done is always written in past tense.
- Concentrate on shortening results—limit background information to a verdict or two.
- Exact spelling, clarity of sentences and phrases, and appropriate reporting of quantities (proper units, important statistics) are just as significant in an abstract as they are anywhere else.

Introduction:

The introduction should "introduce" the manuscript. The reviewer should be presented with sufficient background information to be capable of comprehending and calculating the purpose of your study without having to refer to other works. The basis for the study should be offered. Give the most important references, but avoid making a comprehensive appraisal of the topic. Describe the problem visibly. If the problem is not acknowledged in a logical, reasonable way, the reviewer will give no attention to your results. Speak in common terms about techniques used to explain the problem, if needed, but do not present any particulars about the protocols here.



The following approach can create a valuable beginning:

- Explain the value (significance) of the study.
- Defend the model—why did you employ this particular system or method? What is its compensation? Remark upon its appropriateness from an abstract point of view as well as pointing out sensible reasons for using it.
- Present a justification. State your particular theory(-ies) or aim(s), and describe the logic that led you to choose them.
- Briefly explain the study's tentative purpose and how it meets the declared objectives.

Approach:

Use past tense except for when referring to recognized facts. After all, the manuscript will be submitted after the entire job is done. Sort out your thoughts; manufacture one key point for every section. If you make the four points listed above, you will need at least four paragraphs. Present surrounding information only when it is necessary to support a situation. The reviewer does not desire to read everything you know about a topic. Shape the theory specifically—do not take a broad view.

As always, give awareness to spelling, simplicity, and correctness of sentences and phrases.

Procedures (methods and materials):

This part is supposed to be the easiest to carve if you have good skills. A soundly written procedures segment allows a capable scientist to replicate your results. Present precise information about your supplies. The suppliers and clarity of reagents can be helpful bits of information. Present methods in sequential order, but linked methodologies can be grouped as a segment. Be concise when relating the protocols. Attempt to give the least amount of information that would permit another capable scientist to replicate your outcome, but be cautious that vital information is integrated. The use of subheadings is suggested and ought to be synchronized with the results section.

When a technique is used that has been well-described in another section, mention the specific item describing the way, but draw the basic principle while stating the situation. The purpose is to show all particular resources and broad procedures so that another person may use some or all of the methods in one more study or referee the scientific value of your work. It is not to be a step-by-step report of the whole thing you did, nor is a methods section a set of orders.

Materials:

Materials may be reported in part of a section or else they may be recognized along with your measures.

Methods:

- Report the method and not the particulars of each process that engaged the same methodology.
- Describe the method entirely.
- To be succinct, present methods under headings dedicated to specific dealings or groups of measures.
- Simplify—detail how procedures were completed, not how they were performed on a particular day.
- If well-known procedures were used, account for the procedure by name, possibly with a reference, and that's all.

Approach:

It is embarrassing to use vigorous voice when documenting methods without using first person, which would focus the reviewer's interest on the researcher rather than the job. As a result, when writing up the methods, most authors use third person passive voice.

Use standard style in this and every other part of the paper—avoid familiar lists, and use full sentences.

What to keep away from:

- Resources and methods are not a set of information.
- Skip all descriptive information and surroundings—save it for the argument.
- Leave out information that is immaterial to a third party.



Results:

The principle of a results segment is to present and demonstrate your conclusion. Create this part as entirely objective details of the outcome, and save all understanding for the discussion.

The page length of this segment is set by the sum and types of data to be reported. Use statistics and tables, if suitable, to present consequences most efficiently.

You must clearly differentiate material which would usually be incorporated in a study editorial from any unprocessed data or additional appendix matter that would not be available. In fact, such matters should not be submitted at all except if requested by the instructor.

Content:

- Sum up your conclusions in text and demonstrate them, if suitable, with figures and tables.
- In the manuscript, explain each of your consequences, and point the reader to remarks that are most appropriate.
- Present a background, such as by describing the question that was addressed by creation of an exacting study.
- Explain results of control experiments and give remarks that are not accessible in a prescribed figure or table, if appropriate.
- Examine your data, then prepare the analyzed (transformed) data in the form of a figure (graph), table, or manuscript.

What to stay away from:

- Do not discuss or infer your outcome, report surrounding information, or try to explain anything.
- Do not include raw data or intermediate calculations in a research manuscript.
- Do not present similar data more than once.
- A manuscript should complement any figures or tables, not duplicate information.
- Never confuse figures with tables—there is a difference.

Approach:

As always, use past tense when you submit your results, and put the whole thing in a reasonable order.

Put figures and tables, appropriately numbered, in order at the end of the report.

If you desire, you may place your figures and tables properly within the text of your results section.

Figures and tables:

If you put figures and tables at the end of some details, make certain that they are visibly distinguished from any attached appendix materials, such as raw facts. Whatever the position, each table must be titled, numbered one after the other, and include a heading. All figures and tables must be divided from the text.

Discussion:

The discussion is expected to be the trickiest segment to write. A lot of papers submitted to the journal are discarded based on problems with the discussion. There is no rule for how long an argument should be.

Position your understanding of the outcome visibly to lead the reviewer through your conclusions, and then finish the paper with a summing up of the implications of the study. The purpose here is to offer an understanding of your results and support all of your conclusions, using facts from your research and generally accepted information, if suitable. The implication of results should be fully described.

Infer your data in the conversation in suitable depth. This means that when you clarify an observable fact, you must explain mechanisms that may account for the observation. If your results vary from your prospect, make clear why that may have happened. If your results agree, then explain the theory that the proof supported. It is never suitable to just state that the data approved the prospect, and let it drop at that. Make a decision as to whether each premise is supported or discarded or if you cannot make a conclusion with assurance. Do not just dismiss a study or part of a study as "uncertain."



Research papers are not acknowledged if the work is imperfect. Draw what conclusions you can based upon the results that you have, and take care of the study as a finished work.

- You may propose future guidelines, such as how an experiment might be personalized to accomplish a new idea.
- Give details of all of your remarks as much as possible, focusing on mechanisms.
- Make a decision as to whether the tentative design sufficiently addressed the theory and whether or not it was correctly restricted. Try to present substitute explanations if they are sensible alternatives.
- One piece of research will not counter an overall question, so maintain the large picture in mind. Where do you go next? The best studies unlock new avenues of study. What questions remain?
- Recommendations for detailed papers will offer supplementary suggestions.

Approach:

When you refer to information, differentiate data generated by your own studies from other available information. Present work done by specific persons (including you) in past tense.

Describe generally acknowledged facts and main beliefs in present tense.

THE ADMINISTRATION RULES

Administration Rules to Be Strictly Followed before Submitting Your Research Paper to Global Journals Inc.

Please read the following rules and regulations carefully before submitting your research paper to Global Journals Inc. to avoid rejection.

Segment draft and final research paper: You have to strictly follow the template of a research paper, failing which your paper may get rejected. You are expected to write each part of the paper wholly on your own. The peer reviewers need to identify your own perspective of the concepts in your own terms. Please do not extract straight from any other source, and do not rephrase someone else's analysis. Do not allow anyone else to proofread your manuscript.

Written material: You may discuss this with your guides and key sources. Do not copy anyone else's paper, even if this is only imitation, otherwise it will be rejected on the grounds of plagiarism, which is illegal. Various methods to avoid plagiarism are strictly applied by us to every paper, and, if found guilty, you may be blacklisted, which could affect your career adversely. To guard yourself and others from possible illegal use, please do not permit anyone to use or even read your paper and file.



CRITERION FOR GRADING A RESEARCH PAPER (COMPILATION)
BY GLOBAL JOURNALS INC. (US)

Please note that following table is only a Grading of "Paper Compilation" and not on "Performed/Stated Research" whose grading solely depends on Individual Assigned Peer Reviewer and Editorial Board Member. These can be available only on request and after decision of Paper. This report will be the property of Global Journals Inc. (US).

Topics	Grades		
	A-B	C-D	E-F
<i>Abstract</i>	Clear and concise with appropriate content, Correct format. 200 words or below	Unclear summary and no specific data, Incorrect form Above 200 words	No specific data with ambiguous information Above 250 words
<i>Introduction</i>	Containing all background details with clear goal and appropriate details, flow specification, no grammar and spelling mistake, well organized sentence and paragraph, reference cited	Unclear and confusing data, appropriate format, grammar and spelling errors with unorganized matter	Out of place depth and content, hazy format
<i>Methods and Procedures</i>	Clear and to the point with well arranged paragraph, precision and accuracy of facts and figures, well organized subheads	Difficult to comprehend with embarrassed text, too much explanation but completed	Incorrect and unorganized structure with hazy meaning
<i>Result</i>	Well organized, Clear and specific, Correct units with precision, correct data, well structuring of paragraph, no grammar and spelling mistake	Complete and embarrassed text, difficult to comprehend	Irregular format with wrong facts and figures
<i>Discussion</i>	Well organized, meaningful specification, sound conclusion, logical and concise explanation, highly structured paragraph reference cited	Wordy, unclear conclusion, spurious	Conclusion is not cited, unorganized, difficult to comprehend
<i>References</i>	Complete and correct format, well organized	Beside the point, Incomplete	Wrong format and structuring



INDEX

A

Albeitbellman · 50
Aperture · 5
Appearance · 38

C

Contradict · 53
Cuboids · 40, 42, 43, 44

E

Exogenous · 15

F

Fragmentation · 1, 53

H

Homogeneity · 2
Hypotheses · 15

I

Interruption · 1, 18

O

Optimizing · 53

P

Parallelism · 49, 51
Parametrization · 54
Persists · 17
Polynomial · 51, 56

R

Redundancy · 4, 16, 50

S

Scenario · 50
Synchronisation · 7
Synchronizer · 1, 2, 5, 15, 18, 34

V

Vector · 51, 53
Vectorization. · 49



save our planet



Global Journal of Computer Science and Technology

Visit us on the Web at www.GlobalJournals.org | www.ComputerResearch.org
or email us at helpdesk@globaljournals.org



ISSN 9754350