

A Novel Solution for Distributed Database Problems

Bishoy Sameeh Matta Sawiris¹

Business Information System Department
Faculty of Commerce and Business Administration
Helwan University
Cairo, Egypt

Manal A. Abdel-Fattah², Assoc.Prof.

Faculty of Computers and Information
Information Systems Department
Helwan University
Cairo, Egypt

Abstract—Distributed Databases Systems (DDBS) are a set of logically networked computer databases, managed by different sites, locations and accessible to the user as a single database. DDBS is an emerging technology that is useful in data storage and retrieval purposes. Still, there are some problems and issues that degrade the performance of distributed databases. The Aim of this paper is to provide a novel solution to distributed database problems that is based on distributed database challenges collected in one diagram and on the relationship among DDB challenges in another diagram. This solution presents two methodologies for Distributed Databases management Systems: deep learning-based fragmentation and allocation, and blockchain technology-based security provisioning. The contribution of this paper is twofold. First, it summarizes major issues and challenges in the distributed database. Additionally, it reviews the research efforts presented to resolve the issues. Secondly, this paper presents a distributed database solution that resolves the major issue of distributed database technology. This paper also highlights the future research directions that are appropriate for distributed database technology after the implementation in a large-scale environment and recommended the technologies that can be used to ensure the best implementation of the proposed solution.

Keywords—Distributed database; database challenges; deep learning; fragmentation; blockchain; security

I. INTRODUCTION

Distributed Database (DDB) is the storage paradigm where the users are allowed to store and preserve their data anywhere and are also allowed to access data from any locations [1]. In other words, DDB is constructed with logically distributed multiple databases that are interrelated with. DDB is illustrated in Fig. 1.

As shown in Fig. 1, DDB has multiple databases that are run over different physical or virtual machines at various locations. For an end user, it looks like a single centralized database (i.e.) the user accesses data or uploads data through a centralized server. The centralized server is responsible for managing multiple databases over the system. The main objective of DDB is to prevent overloading at a single point [2]. In conventional centralized databases, all data is stored and accessed in a single server, which increases load on the central server. To overcome this, DDB distributes the load among multiple servers to prevent single point overloading. In DDB, the distributed servers are connected through wire or wirelessly but do not share the memory or clock. In recent times, several advantages have been addressed in DDB systems and those are provided in Table I [3].

Recently, DDB has been adapted in cloud computing technology [4].

Integration of DDB in cloud computing improves usability, scalability and system monitoring. Although DDB has many advantages, it also has some issues that can be listed as follows:

- High Complexity.
- Lack of Security.
- High Cost.
- Lack of Integrity.
- Lack of Standards.
- Complex Design.
- Difficulty in Operating System.

Current research efforts have aimed at optimizing the above issues in DDB system.

A. PROCESS IN DB

In DDB and distributed database management systems (DBMS), two major processes are included. The main processes are:

1) *Replication* – This process looks for changes in DDB. If any change occurs, then a replication process is initiated to make all servers look exactly the same.

2) *Duplication* – To ensure the data is available in all distributed databases, a duplication process is performed. In duplication, a server is considered as master then the database is duplicated as required. Through this process the user is allowed to change the master database without overwriting other distributed databases [5].

B. DB Architecture Model

These processes are important in both homogeneous and heterogeneous databases. In a homogeneous database, all servers have identical software while a heterogeneous database has different software for different servers [6]. In general, there are three common architecture models for DDB. Those are,

- Client-server Architecture.
- Peer-to-Peer Architecture.
- Multi-DBMS Architecture.

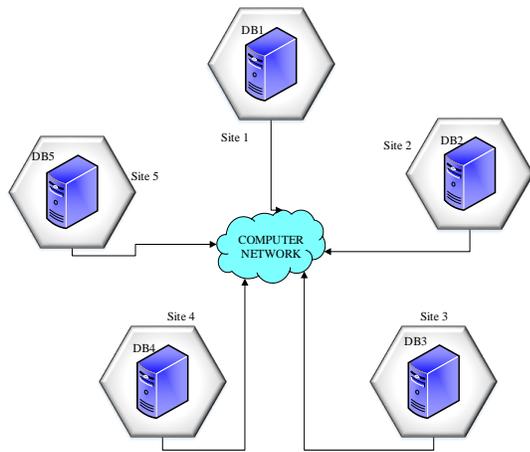


Fig. 1. Distributed Database System.

TABLE I. ADVANTAGES OF DDB

Advantage	Description
Availability	As the data is replicated in multiple sites, DDB generally assures availability of data at all times. If any one of the servers failed to provide data, then the other servers can help.
Data Sharing	Users from anywhere are allowed to share and transmit data in any distributed server.
Robustness	Failures in any one of the servers will not affect the entire DDB system since multiple servers are connected and work together.
High Performance	DDB, in general, enables concurrent data transmissions, which assures a high level of performance. It also shows better delay performance as the local queries are at local servers.
Scalability	In DDB, overloading is not an issue such that it is easy to add more users in the system.

In the client-server architecture model, the functionality is divided into servers and clients. Here, a client server function manages user interface, while a server function covers data

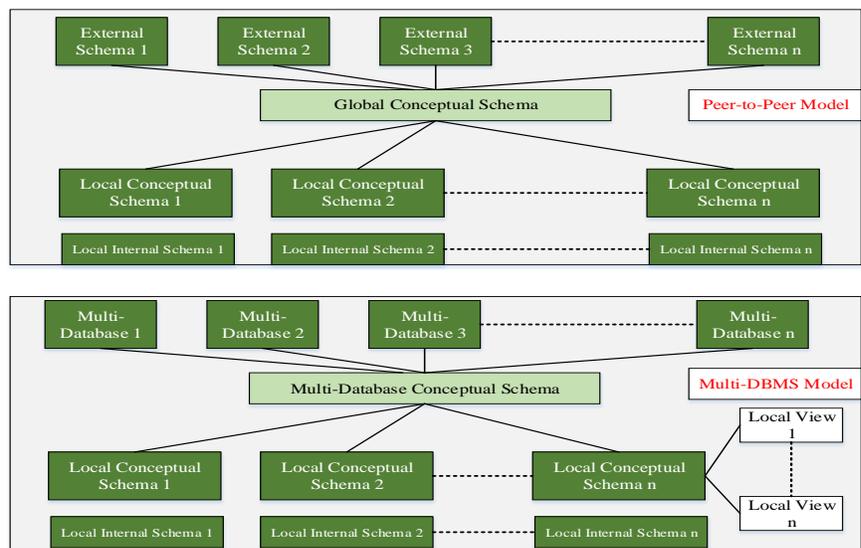
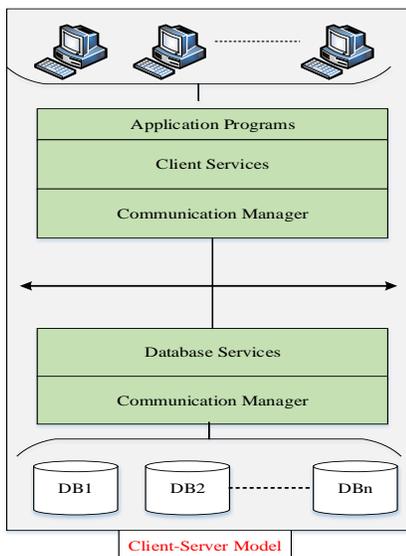


Fig. 2. DDB Architecture Models.

management, query processing, optimization and transaction management. Further, the client-server model involves two different models: single server multiple clients and multiple server multiple clients. In a peer-to-peer model, all servers and clients are connected as peers. For co-ordination, it uses global conceptual schema, local conceptual schema, local interface schema and external schema. The multi-DBMS is formulated by collection of more databases. It uses six level schemas for functioning. Those are multi-database view level, multi-database conceptual level, multi-database internal level, local database view level, local database conceptual level and local database internal level [7]. All three kinds of architectures are illustrated in Fig. 2.

C. Contributions

The major contribution of this paper relies on summarization of research issues in DDB and DBMS. Further contributions are provided as follows:

- The research issues of DDB and DBMS are clearly analyzed and summarized. Background of all issues has been analyzed.
- A novel DDB system is designed as the solution to resolve the research issues. The proposed system relies upon optimizing overall DDB and DBMS through an optimal architectural design.
- Future research directions for improving DDB performance is provided in the conclusion of this paper.

D. Paper Organization

The rest of this paper is organized as follows: Section 2 clearly explains the issues of DDB in architectural and performance aspects. Section 3 reviews the existing research works held on DDB and DBMS. In Section 3, the main research gap is identified and summarized. In Section 4, the researcher’s proposed solution for DDB system is presented. Section 5 concludes the researcher’s contributions and highlights the future research directions.

II. OVERVIEW ON DDB PROBLEMS

This section provides a detailed summary of the research problems in DDB systems. The major problems are depicted in Fig. 3.

A. Distributed Query Processing

The first and foremost problem in DDB is the development of distributed design. The problem is related to placing data and applications in the optimal place across the distributed sites. In DDB, non-replicated (portioned) and replicated alternatives are utilized for placing data optimally. Here, the major issue relies on fragmentation which separates the database into partitions. The other issue is the optimum distribution of those fragments [8].

B. Directory Management

In DDB, the data can be stored in any server across the system. Database directory generally contains the information about the all data items in the DDB system. The directory

maintains descriptions and locations of all data items stored across the distributed system. The main issue is related to the management of directory (i.e.) updating and deletion of data item description and location (i.e. in the system) [9].

C. Distributed Query Processing

Query processing involves acquisition of user query and processing it to retrieve the required data. For that designing optimum algorithms is still an issue since the process of analyzing queries and converting queries into data manipulation operations are complex issues. Thus, designing and optimizing algorithms for query processing is still an issue [8].

D. Concurrent Processing Control

The main aim of DDB system is to serve multiple users concurrently by distributing workload to multiple servers. However, synchronization of these processes is the foremost issue in DDB.



Fig. 3. DDB System Problems.

E. Deadlock Management

Deadlock is an important issue in DDB system that occurs due to the competition between users for accessing the same set of resources, for instance, if many users are requesting for the same data at the same time. Here, the synchronization works upon the locking [10].

F. Reliability

Maintaining consistency of the database affects the reliability of the DDB system. In addition, occurrence of failures is also the main issue for reliability degradation. To ensure reliability, prevention, detection and mitigation of faults improve reliability.

G. Replication Problem

Though replication is one of the key processes of DDB, the algorithm design must ensure the consistency of the copies of data items stored in distributed system.

H. Heterogeneity

Middleware design of DDB system for managing a heterogeneous environment is still a major problem. Middleware is the set of services that acts as interface between end users and DDB system. However, designing such middleware to support heterogeneous users is the main issue.

I. Security and Trust

Security and trust management of the stored data remains an unsolved issue since involvement of attackers crack the security of the system. Unauthorized user access and insecure data storage degrades the security level of overall system [11].

J. Transparency

The main goal of DDB system is that it looks like a single system for the end users. The transparency quality also ensures that the DDB is perceived by the users as a single entity rather than a collection of autonomous systems [12].

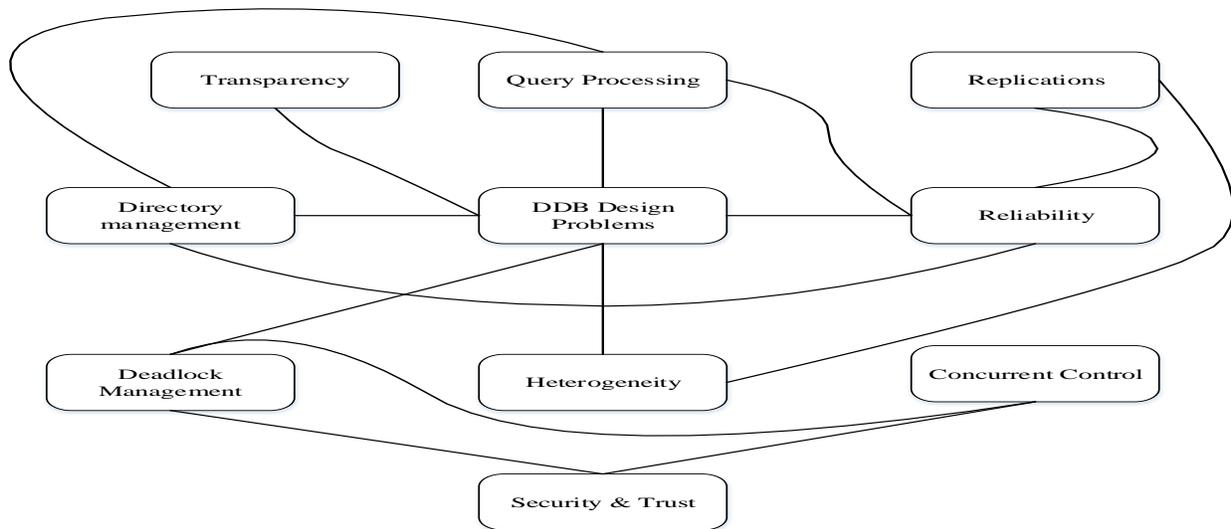


Fig. 4. Relationship among DDB Problems.

All these problems are unsolved in the DDB systems. Although all problems seem to be individual, there is a relationship among them. In Fig. 4, the relationship among DDB problems is summarized.

III. LITERATURE REVIEW

In recent years, many research works have been focused on optimizing the performance of DDB systems. The significant works are analyzed in this section.

Many research works have concentrated on replication and fragmentation management. A new approach was presented in DDB for managing fragmentation and data allocation [13]. This approach splits the data into pair-wise disjoint fragments. Then, it determines whether the fragments were already allocated to any system. Further, high speed clustering technique was presented. First, data was allocated to a single cluster. Then, it was replicated to other clusters. However, this work increases complexity since replication is performed in a non-optimal manner. A cluster-based fragmentation and data replication methodology was presented for flexible query answering in DDB systems [14]. This work uses a standard

clustering algorithm to determine the semantic fragmentation of data in the database. Further, an intelligent query processing methodology was introduced for managing queries in distributed databases. This work also supports load balancing in distributed systems. A combined approach was presented for DDB systems [15]. An optimized heuristic algorithm was designed for horizontal fragmentation and data allocation. The heuristic approach was made up of data fragmentation strategy, data allocation and replication, and clustering strategies. In general, heuristic approaches consume a large amount of time and addition of multiple strategies in the heuristic approach increases time consumption rapidly. A distance-based site clustering approach was presented to support effectual fragmentation [16]. The method of cluster relies on the distance between sites i.e. nearer sites form clusters. For fragmentation, dynamic methods were utilized since static allocation of fragments degrades the response system.

Although this system works better than static methods, this work is unsuitable for large-scale database systems. However, the DDB system is generally large-scale that has to support a

large number of users and data. A query processing methodology was presented to optimize DDB system [17]. Here, the query execution plan was executed then it was used in the search space. The input SQL query was parsed and translated at first. An optimal query search strategy was introduced upon a new tree structure. This work has a higher computational cost in terms of time consumption and computations.

Another main aspect focused upon by many researches works and emerging in DDB is security. A key agreement protocol was used to secure the DDB environment [18]. The proposed scheme was named as key agreement based secure Kerberos authentication protocol (KASKAP). The KASKAP authenticates the distributed servers in order to ensure that only trusted nodes process the user queries.

Further, the trusted nodes were allowed to adjust the process of the DDB system. The conventional Kerberos protocol, which has a lower security level, is used in this work. Thus, this work is not able to achieve the required level of security. In addition to key agreement, cryptography algorithms are also useful in providing security for DDB systems [19]. Symmetrical encryption algorithms were utilized to secure the system. Considered algorithms were: Rijndael algorithm, Reverts Cipher (RC2), advanced

encryption standard (AES), data encryption standard (DES) and triple data encryption standard (TDES). All these conventional algorithms are lower in security level but higher in complexity. However, lightweight and fast algorithms are more suitable for DDB security. An access control mechanism was introduced to ensure security level in DDB systems [20]. Here, the access control policy was applied for the users validating the user’s identity. Security dimension was applied for the users to calculate the permission level. The access control policy is applied based on a single metric which is ineffectual, and the policies are static, which increases malicious user access.

Table II summarizes the research works held in DDB systems and their contributions towards solving the DDB problems. It can be seen that still there are major gaps to be addressed in DDB systems. It can be formulated as follows:

- The fragmentation and query processing must have lower delay and complexity while providing better performance.
- Security mechanisms must ensure high level security without increasing time complexity. The solution must be lightweight and capable of handling large numbers of users.

TABLE II. SUMMARIZATION OF EXISTING WORKS

Work	Problem Focused									Research Issue
	Design Issue	Directory Management	Query Processing	Concurrent Control	Deadlock Management	Reliability	Replication	Heterogeneity	Security	
[8]	✓	✗	✓	✓	✗	✓	✓	✗	✗	High Complexity Non-optimal Replication
[9]	✓	✗	✓	✓	✗	✓	✗	✗	✗	Ineffective concurrent processing
[10]	✓	✗	✗	✗	✗	✗	✗	✗	✗	Higher time consumption
[11]	✓	✓	✗	✗	✗	✓	✓	✗	✗	Not suitable for large- scale system
[12]	✓	✗	✗	✓	✗	✓	✗	✗	✗	Higher computational cost
[13]	✗	✗	✗	✗	✗	✗	✗	✗	✓	Security level is marginal
[14]	✗	✗	✗	✗	✗	✗	✗	✗	✓	Slow processing and security level
[15]	✗	✗	✗	✗	✗	✗	✗	✗	✓	Unable to prevent malicious user access

✓-Focused; ✗-Not Focused;

IV. PROPOSED SOLUTIONS

In this section, the researcher introduces an overview of the proposed solution for DDB problems. Two methodologies for DDB management are presented. The first one is deep learning-based fragmentation and allocation. The second one is blockchain technology-based security provisioning. Deep learning is the evolving Artificial Intelligence (AI) technique that resembles the artificial neurons in a deep manner. General Deep Neural network (DNN) involves three kinds of layers. Those are the input layer, the hidden layer and the output layer. Input layer is responsible to get input from the users (in this work, the input is the user query or data). In the hidden layer, the weight value is tuned for each input upon a set of criteria. In the output layer, the output of the given input is obtained. The logical structure of DNN is illustrated in Fig. 5.

Deep learning has many advantages over DDB systems. The important benefit is that it assures a high-level performance by providing an optimal solution in fragmentation and query optimization. In the following pseudocode, the process of DNN is explained in detail. This procedure is adapted in processing of data in the DDB system. Here, the hidden layer is responsible to compute weight value upon objective function. For example, if DNN is sued for clustering sites then the weight value is computed in terms of distance between the sites.

Pseudocode for DNN

Input: Data Points $\{ X = X_1, X_2, X_3, \dots, X_n \}$

Output: Processing outcome as $\{ Y = Y_1, Y_2, Y_3, \dots, Y_n \}$

Begin

 Initialize {bias}

 Assign weight $w = 0$

 While {Iteration < Maximum Iteration} do

 Update weight as w_i //Hidden Layer

 Obtain average weight //Output Layer

 Update Learning rates

 Return (Y)

 End While

End

When it comes to blockchain technology, it is a distributed ledger technology that supports effectual security management over the DDB systems. In blockchain, all transactions (like user credentials, queries, retrievals, server credentials, transaction logs) are stored in the distributed blocks. For each transaction, corresponding hash value is generated and stored in the blockchain structure. For hashing, secure hashing algorithm-256 is applied twice as $(\text{SHA-256})^2$ and signature is verified based on elliptic curve signature generation. The model of blockchain is illustrated in Fig. 6.

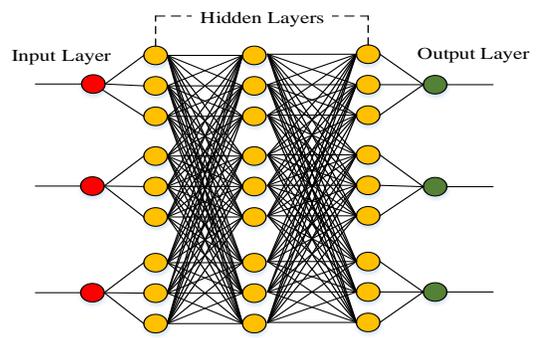


Fig. 5. DNN Architecture.

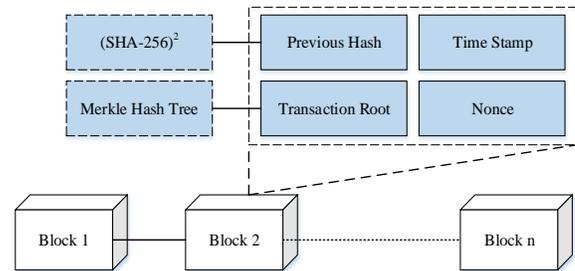


Fig. 6. Blockchain Model for DDB.

With the aid of blockchain ledger, it is possible to perform secure operations in the system. It assures security, integrity, authenticity, access control and non-repudiation, which are important security requirements. The following pseudocode explains the procedure of secure hashing performed in the blockchain. Here, each data accessed or stored in the DDB system is provided with the secure hash function.

Pseudocode for Hash Generation

Input: Data $\{ X = X_1, X_2, X_3, \dots, X_n \}$

Output: Hash of Data $(H\{XX = X_1, X_2, X_3, \dots, X_n \})$

Begin

 Initialize hash values

 From $\{h_0$ to $h_7\}$ as primes

 Initialize Round Constants

 Perform padding on X

 Make X into 512 bits Chunks

 For each chunk Do

 Create Schedule Array

 Extend Schedule Array

 Initialize working variables

 Compress schedule array

 Add compressed Chunk to $H\{Y\}$

 End for

End

In Fig. 7, overall proposed architecture is illustrated. In the proposed solution, all users and DDB system are connected through distributed blockchain to ensure high-level security. In general, blockchain stores the current hash value and previous block hash value which ensures non-repudiation and integrity for the data stored in DDB.

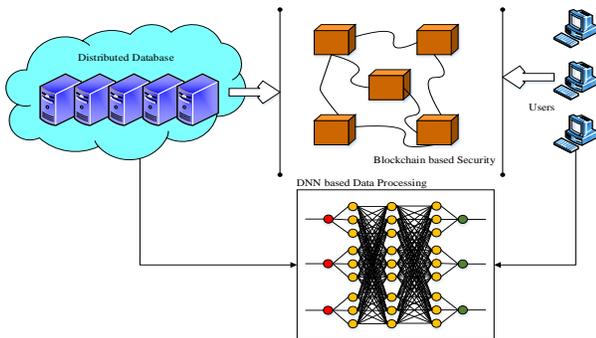


Fig. 7. Proposed DDB Architecture.

Further, user authentication credentials can be stored in the blockchain as an alternate for traditional centralized server. As a result, the single point failure issue of centralized server is overwhelmed. For fragmentation and data allocation problem, DNN based data processing procedure is proposed. Thus, solutions are presented for major problems of DDB systems. Both solutions can work together to ensure efficiency and security.

V. CONCLUSION AND FUTURE DIRECTION

This paper presents an overview on distributed database (DDB) system. First, it summarizes major research problems of DDB and the relationship among the problems. Then, it reviews significant works focus on resolving the problems of DDB. The analysis shows that still there are several issues that are unsolved in DDB systems. After critical analysis, the researcher proposes a novel solution that combines Blockchain and Deep Learning for DDB security and data processing respectively. Overview on the proposed solution is given in detail. Further research can be carried over the following directions:

- The system needs to be tested and evaluated in a large-scale environment.
- Proposed DDB system can be combined with Hadoop file systems to make the processes faster and more efficient.
- Lightweight and scalable blockchain technology will be useful to cope with the DDB environment.
- Lightweight cryptography techniques can be employed to improve security level.
- Unsupervised clustering methods can be tested for fragmentation process.

REFERENCES

- [1] D.S.Hiremath, Dr.S.B.Kishor, (2016). "Distributed Database Problem areas and Approaches", IOSR Journal of Computer Engineering, 15-18.
- [2] Raouf, A.A., Badr, N., & Tolba, M. (2017). Distributed Database System (DSS) Design Over a Cloud Environment.
- [3] Sharma, M., Singh, G., & Singh, R. (2016). Design and analysis of stochastic DSS query optimizers in a distributed database system. Egyptian Informatics Journal, 17, 161-173.
- [4] Seybold, D., & Domaschka, J. (2017). Is Distributed Database Evaluation Cloud-Ready? ADBIS.
- [5] Xiong, X., Shen, Y., Feng, G., Guo, M., & Wang, B. (2019). Research on Multi-level Distributed Storage Method of Economic and Financial Management Database. 2019 International Conference on Virtual Reality and Intelligent Systems (ICVRIS), 411-414.
- [6] Zhou, Y., Chen, Q., Shan, B., Jiang, F., & Pang, Y. (2019). A Distributed Storage Strategy For Trajectory Data Based On Nosql Database. IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium, 3487-3490.
- [7] Gupta, Swati & Saroha, Kuntal & Bhawna, & Tech, M & Scholar, Pdmce & Bahadurgarh., (2011). Fundamental Research of Distributed Database. International Journal of Computer Science and Management Studies. 11.
- [8] Katembo Kituta Ezéchiél, Shri Kant, Ruchi Agarwal, (2019). A Systematic Review On Distributed Databases Systems And Their Techniques. 15th January 2019. Vol.96. No 1, 1992-8645.
- [9] M.T. O' zsu, U. Dayal and P. Valduriez (eds.), Morgan-Kaufmann An Introduction to Distributed Object Management, 1994, pages 1-24.
- [10] Swati Gupta, 21, July 2013, Deadlock Detection Techniques in Distributed Database System, International Journal of Computer Applications (0975 - 8887).
- [11] Abubakar Mohammed, Bashir Maina Saleh, January 2017, Centralized Database: A Prerequisite for Security and Sustainable Development in Nigeria, International Journal of Innovative Research in Computer Science & Technology (IJIRCST).
- [12] Shital Jadhav, Ashwini Gawande, February 2017, An Overview of Distributed Database System's Advantages and its Problem Areas, International Journal of Innovative Research in Computer and Communication Engineering, (3297: 2007).
- [13] Al-Sayyed, R.M., Zaghoul, F.A., Suleiman, D., Itriq, M., & Hababeh, I. (2014). A New Approach for Database Fragmentation and Allocation to Improve the Distributed Database Management System Performance. Journal of Software Engineering and Applications, 07, 891-905.
- [14] Wiese, L. (2014). Clustering-based fragmentation and data replication for flexible query answering in distributed databases. Journal of Cloud Computing, 3, 1-15.
- [15] AliA.Amer, Adela.Sewisy, TahaM.A.Elghendy. (2017). An optimized approach for simultaneous horizontal data fragmentation and allocation in Distributed Database Systems (DDBSs). Heliyon3.
- [16] Suganya, A., & Kalaiselvi, R. (2013). Efficient Fragmentation and Allocation in Distributed Databases. International journal of engineering research and technology, 2.
- [17] Alom, B., Henskens, F., & Hannaford, M. (2009). Query processing and optimization in distributed database systems.
- [18] Natarajan, M., & Manimegalai, R. (2018). Key agreement based secure Kerberos authentication protocol (KASKAP) for distributed database access in secured manner. International journal of engineering and technology, 7, 24.
- [19] Khamis, A.D., & Subair, S.O. (2019). Security Framework for Distributed Database System.
- [20] Guclu, A., Bakir, C., & Hakkoymaz, V. (2020). A New Scalable and Expandable Access Control Model for Distributed Database Systems in Data Security. Hindawi.