ACADEMICIA: An International Multidisciplinary Research Journal
ISSN: 2249-7137 Vol. 12, Issue 05, May 2022 SJIF 2022 = 8.252
A peer reviewed journal

PROBLEMS OF DATA REPLICATION IN DISTRIBUTION SYSTEMS

Dadamukhamedov Alimjon*

*Senior Lecturer,
Department of "Modern information and communication technologies",
International Islamic Academy of UZBEKISTAN
Email id: chinororg@mail.ru

DOI: 10.5958/2249-7137.2022.00585.7

ABSTRACT

In this article, we will compare the replication methods available in database systems.These problems are to maintain consistency between the actual state of the real-time object of the external environment and its images reflected in copies distributed across multiple nodes. Nowadays, modern applications of devices connected to the Internet are experiencing rapid growth and variability of transactional workloads. Database replication should increase access to databases to calculate efficiency. The replication algorithm allows high-speed distribution of changes in the database to all replicas, which ensures the robustness of all replications. However, a fragmented routing algorithm is used to consistently balance the load of incoming transactions on existing instances. Shows how it can perform almost linear measurements of workload for databases. To expand the idea of large-scale database modeling, we will consider improving the consistency and scalability of data using an algorithm that is applied and available in the database. Individual levels of iteration to prevent overuse of resources, all of which together help solve the problem of scalability for distributed real-time database systems.

KEYWORDS: *Method, Replicated database, Replicated Database Design, Replicated database protocols, Transactional replication, Data consistency and Scalability, Active and Passive replication, Recognition.*

INTRODUCTION

Data replication is an enticing backup method because of two main reasons: its safety and its quick convenience. The method helps organizations maintain multiple up-to-date copies of their data, distributing it to data centers close to remote offices.

When creating an application for work within information processing distribution systems, programmers often need to develop a data replication mechanism.Replication is a method of data exchange to ensure data compatibility between redundant resources such as software, hardware components to improve reliability, fault tolerance, usability.Replication in distributed computing systems is the process of copying information from one database to another and then combining them.Most modern distribution applications run on multiple databases.The amount of information processed by the software that is part of the distribution system is currently so large that it is technically impossible and inexpedient to store it in a single computing network node.As a result, systems using distributed storage and processing technologies are becoming more common. Within each system of this type, there are several nodes for storing information.A

constant condition for the normal functioning of the system is the logical coordination of data operations performed on different nodes of the system.Fulfillment of this condition ensures the use of the distribution transaction mechanism. At this point, it may be necessary for the distribution system to be compatible with the data at different nodes[2].

As a rule, the need for such a combination of structural data sources arises between independent system nodes, such as a separate database server, between the primary and backup database servers, between the database server and the semi-autonomous workstations.The compatibility of different sources of information is achieved by replicating the data.The following is a detailed look at the different aspects of a distributed computing system that require replication.Many modern information processing distribution systems are based on the operation of one or more database servers. Reliability of information storage and functional stability of the server are the main factors determining the performance of the system.It is known that the most common way to protect against data loss is to create working backups of the database.Having a backup in case of data corruption on the server allows you to quickly restore the database.The advantage of this method of interrupt protection is its versatility and general usability.The process of creating backups does not require the addition of additional expensive hardware to the system, as backups can be created on the server itself or on external media.Recovering data from an enterprise-wide system database backup can take several hours, which is completely unsuitable for production.In addition, backing up a database is the only way to protect against data loss.Functional interruptions to the entire server often require a full reinstallation of the entire system software components to restore its functionality, which takes a long time.

LITERATURE REVIEW

The strategy we used to create the search strings was as follows [2] [19]: • finding papers about distributed information system. • Listing keywords mentioned in primary studies, which we knew about. • Use synonyms word (usage) and sub subjects of network technology in data replications such as (distributed information system, database, replication, query intensity, telecommunication network, datacenter). • Use the Boolean OR to incorporate alternative spellings and synonyms. • Use the Boolean AND to link the major terms from population, intervention, and outcome. The complete search string initially used for the searching of the literature was as follows: network technology AND data replications. It has been highlighted in [4] [16] that there are two main issues on conducting an SLR search which are the sensitivity and specificity of the search. In our preliminary search, when we used the complete search string defined above we retrieved a very high number of articles. For instance, Google scholar, Scopus, ProQuest education, IEEEXplore, Science Direct, Springer Link retrieved more than two hundred results. Therefore, we have deepened our search and used this search string: (Adoption OR Usage)AND (religious database. ORdatabase) AND (query intensity OR telecommunication network). The revised search string has given us a reasonable number of studies and we finally selected relevant empirical studies

THE MAIN PART

A faster recovery of the system after a downtime is possible when using a backup server, which is a complete analogue of the database server.During system operation, the information stored on the working server is periodically replicated to the backup server.In case the system's working server fails, it can be reset to work with the backup server as soon as possible.This method of

interrupt protection is more expensive, requiring the allocation of additional hardware nodes under the database backup server, but in the process of continuous production, such costs are fully justified by the possibility of rapid system recovery in the shortest possible time[6].Data replication on the backup server is an important part of the described scheme of system protection against interruptions. The speed and accuracy of the synchronization between the database on the worker and the backup servers depends on the ability to update the system operation of the backup server based on the minimum loss of information and working time (Figure 1).

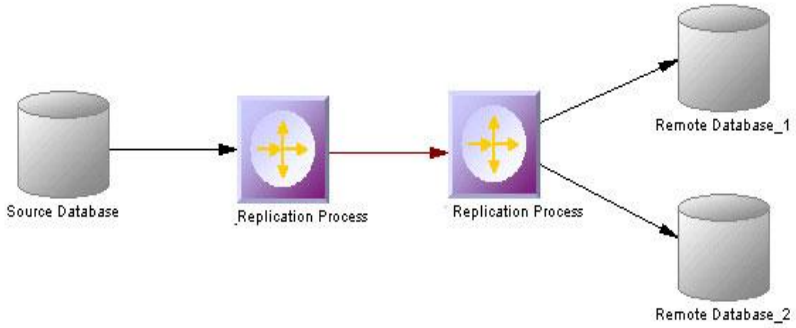

Figure 1. Replication Process

Another example of the use of replication between databases can be found in complex unified systems built by combining several subsystems, each of which works with its own private BD server.Separate segments of such a system can be data servers and connected workstations located at great distances from each other.As a rule, this situation makes it impossible for the system nodes to interact in the form of a direct permanent connection.One solution to this problem is to replicate data between system servers for public use[5].

Data replication should be performed in the background without interrupting the normal operation of the system.The need to provide system users with the most reliable and up-to-date information requires that data be replicated at regular intervals.In addition, the low bandwidth of the communication channels used to interconnect the servers is based on the need to minimize the amount of data transmitted across the network per replication session[9].

Thus, in this case, as the third example of the replication distribution architecture, the distribution computing system plays an important role in ensuring the normative operation.The practice of data replication plays an important role in his work.It is possible to consider the organization of the module of the distribution system in a semi-automatic mode.An example of a successful solution to a number of problems is the semi-automated mode of application operation within the distribution system, when the use of system resources is limited.

First, when using unreliable communication channels, the semi-automatic mode of application operation increases the stability of the application operation. It is known that online applications can work only when there is a connection to the server, and semi-autonomous applications can work when there is no connection to the server and when there is no connection[11].

Second, the use of semi-autonomous applications allows you to avoid increasing network congestion by minimizing network traffic.If there are enough online applications in the system, the development of a semi-autonomous application will be a successful solution, which will help to prevent the increase in traffic to the communication channel by adding new jobs.

Third, the semi-autonomous modules can be configured in such a way that no data exchange takes place during the high load on the semi-autonomous workstations and the central database server.This system allows the central server to distribute the load more evenly and optimally over time[15].

Creating semi-autonomous modules within a distribution system is especially effective when the system is used in practice within its resource capabilities.As the number of online jobs increases, the load on the central server increases, network traffic increases, which leads to a decrease in the efficiency of the entire system.In this case at least the central server hardware needs to be upgraded or more serious architectural changes need to be made in order to restore the system to its proper level.

For the system module to work effectively in a semi-autonomous mode, it is necessary to organize the use of the application to a certain part of the information stored in the central database.There is no connection between the workstation and the BD server. This task can be solved by creating a local database on a workstation designed to store an exact copy of the information required from the central database.The presence of such a database on the workstation ensures that the application has access to data even when it is not always connected to the server[14].

When working with a local database, the chances of getting up-to-date information are reduced compared to an online procedure.The data uploaded to the local database can only guarantee compatibility with the information on the central server at the time of download.Subsequently the probability of a gap between these data warehouses increases at a rate corresponding to the average intensity of changes in the data entered into the central database.The longer the time intervals the greater the differences that can be collected in the local database.In order to ensure the compatibility of the information in the local database, it is necessary to ensure strict periodicity of replication from the central database to the workstation.The practice of replicating data from a central database to a workstation must meet a number of requirements.

First, replication practices must ensure the reliability of the information uploaded to the local database.

Second, the time spent on replication should be minimized as much as possible.

Third, the way the program accesses the data stored in the central database must meet the security requirements of the system.

The suitability of the data replication mechanism for these conditions depends on the efficiency and reliability of the semi-autonomous module and the efficiency of the automated production process[19].

Data in a distributed system is stored between multiple computers on a network. Some of the reasons for data duplication in distributed systems are:

Error resistant: The system works even if there are network problems. If one replica fails, the service can be provided with another replica.

Reduced latency: By storing data closer to the consumer geographically, Replication helps reduce data request delays.

Reading Scope: Reading requests can be provided from copies of the same data that are repeated. This increases the overall throughput of the queries.

High availability: Replication in distributed systems is the most important aspect of increasing data availability.The data is duplicated in many places, so the user can log in even if some copies are not available due to site glitches.

There are several types of data duplication in a distributed system based on certain types of architecture:

• Asynchronous and synchronous replication

• Active and passive replication

• Based on server model

• Replication schemes

Asynchronous replication: In this replication, the replica is changed after a commitment is made to the database.

Synchronous replication: In this replication, the replica is changed immediately after some changes are made to the relationship table**(Figure 2)**.

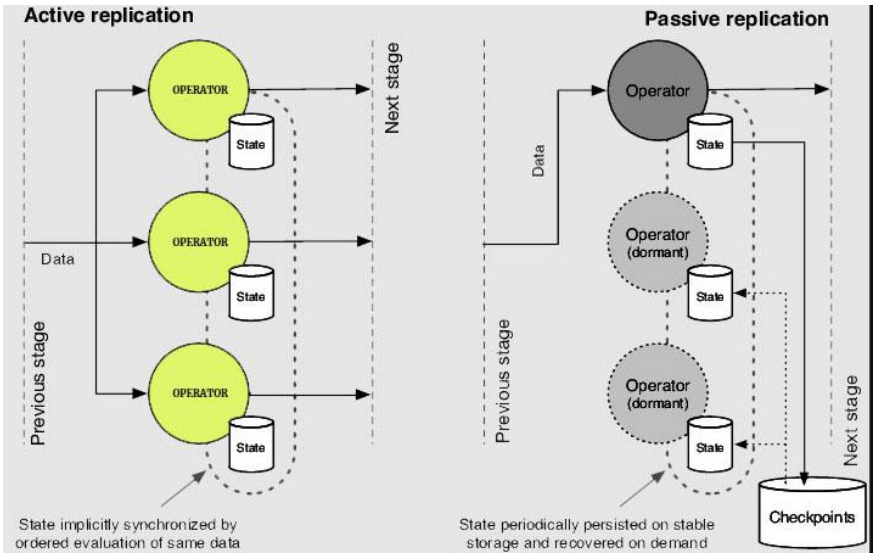

Figure 2.Active vs Passive Replication

Active replication: Active replication is a decentralized replication mechanism. The basic idea is that all copies receive and process the same customer requests.Consistency is ensured by assuming that the replicas produce the same result when the same input is given in the same

sequence. This assumption indicates that the servers respond deterministically to the queries.Clients refer to a group of servers, not a single server.Client requests can be transmitted to servers via Atomic Broadcast so that they receive the same access in the same sequence[18].

Passive Replication: Client requests in Passive Replication are processed by only one server (primary name).After the main server processes the request, it changes the status of the other (backup) servers and responds to the client.

If the primary server fails, one of the backup servers will take over. Even non-deterministic processes can benefit from passive replication.

The disadvantage of passive replication over active replication is that the response is delayed in case of failure.

Based on server model.

Single Leading Architecture: In this architecture, a single server accepts client write and copies data from it. This is the most popular and traditional method. It's a synchronous technique, but it's also very rigid. Multi-Leading Architecture: In this architecture, multiple servers can accept write and serve as templates for copies. Copies should be distributed so as not to be delayed, and managers should be close to all of them[14].

No management architecture: Each server in this architecture can accept write and act as a replication model. While it provides maximum flexibility, it makes synchronization difficult(Figure 3).

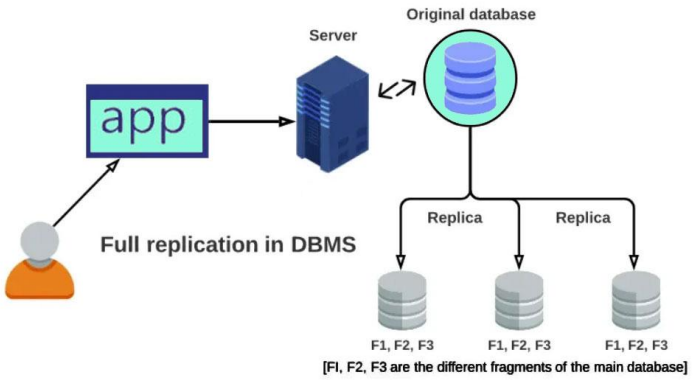

Figure 3.No management architecture

Partial data replication: Here only selected parts of the database are repeated depending on the importance of the data on each site. The number of copies, in this case, can be any from one to the total number of nodes in the distributed system[18].

Partial databases are stored on personal computers and such replication can be effective for members of sales and marketing teams who are regularly synchronized with the main server (Figure 4).

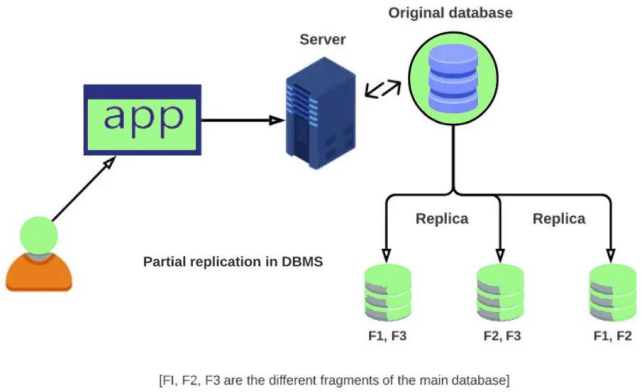

Figure 4.Partial data replication

No Replication: In this Replication scheme, each node in the distributed system receives a copy of only one partition of the database.

While the lack of replication may be related to the simplicity of data recovery, it can slow down the execution of queries as multiple users access the same server.

The absence of data duplication in DBMS ensures that data is available relative to alternative replication methods(Figure 5).

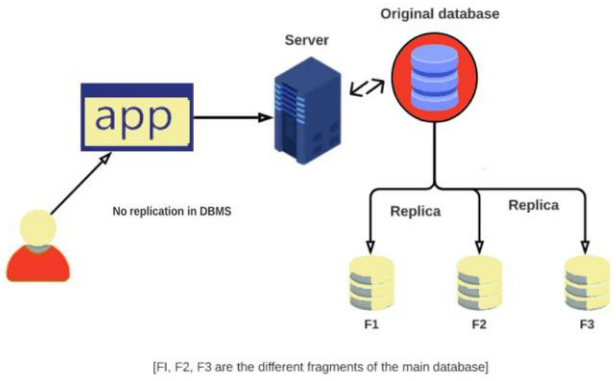

Figure 5.No Replication:

Data Modification: The best class and local help to change complex data is at your fingertips. Code & No-code Fexibilty is for everyone.

Smooth Mapping of Schemes: Manage fully automated circuits for incoming data with the desired location.

Scalable: An excellent horizontal measurement with minimal delay for modern data needs.

Built-in connectors: support more than 100 data sources, including databases, SaaS platforms, files and more. Native Webhooks and the REST API connector are available for custom resources.

Quick setup: Easy interface to work with minimal setup time for new customers.

Exception security: a fault-tolerant architecture that provides zero data loss.

Live support: Available around the clock to extend great support to its customers through chat, email and support calls.

Advantages of data replication in distributed systems

Some of the advantages of replicating data in a distributed database or system are:

Improving Analysis: The team can implement Analytics without compromising productivity by having a separate, complete copy of the database.

Increase availability: A distributed database allows multiple users to view and manage data without interfering with each other.

Ensures business continuity: Increasing data on distributed systems as part of your emergency recovery strategy ensures that an off-site copy of the system is available in the event of a hardware failure or a payment program attack. This allows businesses to recover data while ensuring business continuity[13].

Advanced Performance: Because the same data is stored in different locations, users can access data from a server near them which reduces network latency and speeds up.

Allows multiple users to access: Multiplying data helps to execute queries, especially when multiple users access the database.

Disadvantages of data replication in distributed systems

Duplicating data in distributed systems can cause several problems, as discussed below:

It can take up a lot of storage space, especially when fully replicated. If multiple copies need to be updated at the same time, this can lead to significant financial costs or performance degradation. Maintaining data consistency can be difficult when using merging or peer-to-peer replication.

Different sources may not be synchronized with each other due to incorrect or outdated replication. This can lead to unnecessary data warehouse costs spent on processing and storing unnecessary data.

There are maintenance and other costs associated with using multiple servers. These costs must be borne by the organization or a third party. If they are managed by a third party, the company runs the risk of blocking the vendor or having problems with services that are not under its control.

RESULTS

In order to organize the work of the application in a semi-autonomous manner, it is necessary to consider the mechanism of data replication between the central BD and the local database, as well as to ensure the timely implementation of appropriate data exchange practices.Development of a software architecture model of the replication mechanism:

- Establish basic requirements for the data storage system used to create a local database
- Choice of replication strategy

- Check the ability of the proposed mechanism of replication

There is currently a set of colorful architectural models that allow you to create distribution applications. It is necessary to distinguish the criteria for selecting the application architecture in accordance with the conditions of application of the software product. A number of recommendations have been formulated on the selection of effective software development technology for these applications.It is necessary to determine the approaches to the replication strategy based on the set task conditions.

CONCLUSION

In short, you have the opportunity to multiply data on distributed systems, distributed transactions, and distributed systems. We explored the need for all of these systems and techniques. In addition, you have studied different types of replication in distributed systems.At the end of this article, you explored the various benefits and challenges associated with data reproduction in distributed systems.Today, businesses are facing more diverse and complex data sets than ever before.As a result, organizations can no longer manage their data through simple Replication processes alone. To maintain competitiveness most businesses now use a number of automated processing methods.

REFERENCES

1. E. Tanenbaum, M. Van Steen. distributed systems. Principles and para-radigms. St. Petersburg: Peter, 2003.877p.

2. Georgiou, M., Panayiotou, M., Odysseos, L., Paphitis, A., Sirivianos, M., &Herodotou, H. (2021). Attaining Workload Scalability and Strong Consistency for Replicated Databases with Hihooi. Proceedings of the 2021 International Conference on Management of Data.

3. Belousov, V. E. (2005). Algorithms for data replication in distributed information processing systems (Doctoraldissertation, Penza: PGU, 2005.–184 p.).

4. NishonboevT. Software configured networks. ”Textbook. (Griffith). TUIT printing house named after Muhammad al-Khwarizmi. 2017 (p. 186)

5. Nishanbayev, T.N., Abdullayev, M.M., Maxmudov, S.O.The model of forming the structure of the 'cloud' data center. International Conference on Information Science and Communications Technologies: Applications, Trends and Opportunities, ICISCT 2019

6. V. E. Belousov. Data replication algorithms in distributed information processing systems. Int mat: http://diss.rsl.ru/diss/05/0591/050591031pdf.

7. Belousov V, E. Features of building a queuing system within the framework of the middle level of a distributed information processing system " // System analysis, management and information processing: scientific-tech, collection of articles: 2005, no. No. 1, - Penza, PSU, 2005, - p. 23-31

8. Basharin G.P., et al. Analysis of queues in computer networks: theory and calculation methods / G.P.Basharin, P.P.Bocharov, Ya.N.Kogan. - M.: Nauka, 1989.-336 p.

9. Irgashevich, D. A. (2019). Development of national network and corporate networks (in the case of Tas-IX network). International Journal of Human Computing Studies, 1(1), 1-5.

10. Irgashevich, D.A. (2020). Development of national network (tas-ix). ACADEMICIA: An International Multidisciplinary Research Journal, 10(5), 144-151.

11. Dadamukhamedov, A. I. (2017). Development of a national network and a corporate network (eg Network IX). Current Research in the Modern World, (3-2), 133-137.

12. Dadamuhamedov, I. A. (2020). Cloud technologies in islamic education institutions. TheLightofIslam, 2 (23).

13. V. Feller, Introduction to probability theory and its applications. Volume I. -M.: Mir, 1967.- 498 p.

14. ObergR.J. COM+ technology. Fundamentals and programming. : Per. from English: Uch. settlement - M.: William, 2000, - 480 s: ill.

15. Sunbled C, Sunbled P. Development of scalable applications for Microsoft Windows. Master Class. (Translated from English) - M .: Russian edition, 2002. -416 s: ill.

16. Dadamuxamedov, A., Mavlyuda, X., &Turdali, J. (2020). Cloud technologies in islamic education institutions. ACADEMICIA: An International Multidisciplinary Research Journal, 10(8), 542-557.

17. Dadamuxamedov, A. (2020). The impact of online communication on youth education. International Engineering Journal For Research & Development, 5 (6), 10.

18. Kumar, Sanjay & Sharma, Kunal&Swaroop, Vishnu.Issues in Replicated data for Distributed Real-Time Database Systems.(IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 2 (4) , 2011, 1364-1371

19. https://hevodata.com/learn/data-replication-in-distributed-system/