Review paper for fragmentation and allocation in distributed database systems

Researcher 1 : Ghassan Awad Hamad
ghassanimc@gmail.com
Department of Computer Science, Collage of Computer Science and Mathematics, Tikrit University, Iraq
Researcher 2 : Asst. Prof. Dr. Saadi Hamad Thalij
Sadi_aluhabi@yahoo.com
Department of Computer Science, Collage of Computer Science and Mathematics, Tikrit University, Iraq

ABSTRACT:

The use of fragmentation and allocation in distributed databases for replication technology has seen a rise in interest in recent years. The significant works in this topic that were published after 2020 are reviewed, In numerous research, fragmentation-based replication approaches have been investigated in relation to various application domains, such as cloud computing, big data analytics, and IoT. For instance, fragmentation can be used in cloud computing to disperse data over several nodes, enhancing performance and lowering the likelihood of data loss. Data fragmentation in big data analytics can allow for parallel processing, which can drastically cut down on processing time. Fragmentation in the Internet of Things can make data transfer more effective while using less energy, And other methods used by researchers in this field, which will be reviewed in this paper.

introduction

Modern distributed databases must incorporate replication technology because it increases the database's availability, dependability, and scalability by making numerous copies of it. Fragmentation and allocation, which divides the database into smaller pieces known as fragments and distributes them across other network nodes, is one of the primary replication techniques [1] . Improved performance, greater resource use, and increased failure resilience are just a few advantages of this strategy.

After 2020, fragmentation-based replication strategies have drawn more and more attention in the literature. Data is fragmented when it is broken up into smaller groups or pieces that can be handled and stored separately. This strategy can increase the effectiveness, scalability, and fault tolerance of data processing. [2]

RELATED WORKS

MAIKEL LÁZARO PÉREZ GORT 2020 , offers double watermark fragmentation leveraging the virtual primary key set's current redundancy. By doing so, he ensures that the watermark can be correctly identified even if some relational properties are deleted. [3]

To increase the effectiveness of data transmission in wireless sensor networks, Zhou et al. (2020) suggested a fragmentation-based data replication strategy for the Internet of Things. To save energy and increase data transmission reliability, the plan involves breaking up data into little chunks and replicating them over several nodes. Through simulations and experiments, the authors analyzed their strategy and demonstrated that it outperforms current data replication system. [4]

For cloud storage systems, Chen et al(2020) .'s study suggested a fragmentation-based data replication technique. By replicating data among

several cluster nodes, the plan is intended to lower the chance of data loss. Through simulations and tests on a real-world cloud storage system, the authors proved the viability of their strategy. [6]

Li et al(2021) .'s fragmentation-based data replication approach was suggested in another study for big data analytics. To facilitate parallel processing, the approach entails breaking up the data into little pieces and reproducing them among other nodes. Using a benchmark dataset, the authors assessed the effectiveness of their approach and showed how it may shorten processing times. [7]

A. NORAZIAH 2021 , To manage fragmented database synchronous replication, the Binary Vote Assignment on Grid Quorum with Association Rule (BVAGQ-AR) algorithm is suggested.
The BVAGQ-AR technique can divide the database into separate chunks. [1]

Deep Reinforcement Learning Approach for Dynamic Fragmentation and Allocation in Distributed Databases by X. Wang et al. (2021) - This study uses deep reinforcement learning to introduce a unique method for allocation and fragmentation in distributed databases. The algorithm that the authors suggest is based on reinforcement learning and dynamically fragments and distributes data in response to adjustments in the database workload and network circumstances. The algorithm is proved to perform better in terms of performance and resource consumption than conventional fragmentation and allocation approaches. [8]

By Y. Liu et al. (2021), "A Comparative Study of Fragmentation and Allocation Techniques for Distributed Databases" - This paper offers a thorough analysis of several allocation and fragmentation methods for distributed databases. In terms of query response time, load balancing, and resource usage, the authors assess the performance of horizontal

fragmentation, vertical fragmentation, hybrid fragmentation, and cost-based fragmentation and allocation. According to the study's findings, the best strategies for maximizing resource use and performance are cost-based fragmentation and hybrid fragmentation. [9]

S. Zhang and colleagues' "Efficient Fragmentation and Allocation in Replicated Databases Using Blockchains" (2022) - This study introduces a novel blockchain-based method for allocation and fragmentation in replicated databases. The authors suggest a blockchain-based system that permits decentralized data allocation and fragmentation, which offers higher security and dependability than conventional approaches. The study's findings demonstrate that the suggested strategy is successful in obtaining high performance and cutting costs. [10]

By Y. Wei et al. (2022), "A Hybrid Approach to Fragmentation and Allocation in Replicated Databases Using Machine Learning" - This study uses machine learning to provide a hybrid solution to allocation and fragmentation in replicated databases. The authors suggest a system that dynamically distributes pieces based on shifting demand and network circumstances by combining conventional fragmentation and allocation approaches with machine learning algorithms. The study's findings demonstrate that the proposed system performs better in terms of performance and resource use than conventional methods. [11]

By Z. Li et al. (2022), "An Energy-Efficient Approach to Fragmentation and Allocation in Replicated Databases" - This research introduces a novel energy-efficient method for fragmentation and allocation in replicated databases. The authors offer a cost-based approach that strikes a compromise between energy use and performance, and they assess how well the suggested method performs in terms of energy efficiency, query response time, and resource use. The study's findings demonstrate that the suggested

approach is successful in lowering energy usage while retaining high performance and resource efficiency. [12]

Distributed Databases

A database that has two or more files spread across various sites, whether they are connected by the same network or distinct networks, is said to be dispersed. The processing is split among numerous database nodes, and portions of the database are physically stored in various places [13]

Replication Database

The technique of maintaining numerous copies of a database on various servers is known as replication in databases. A number of benefits include better performance and scalability, increased availability and fault tolerance, and the flexibility to read and write data to various locations. [14]

Replication Database forms

Replication can take many different forms:

- In a master-slave replication system, one server manages all write activities while the slave servers repeat the data from the master and manage read operations.
- Replication using multiple masters allows for universal acceptance of writes and change propagation between servers.
- Peer-to-peer replication: Each server is capable of receiving writes and disseminating updates to the other servers in a peer-to-peer fashion.
- One or more of the aforementioned replication types combined to form hybrid replication.

- Replication can be helpful for increasing a database's availability, scalability, and performance, but it can also complicate the system and raise the risk of data inconsistencies. [15]

types of replication

two different types of replication schemes: static and dynamic If a distributed database has a set number of replicas, it is said to be **static scheme**. Reconfiguring and restarting the system will change the number of copies in this scheme. The difficulty of building a distributed database to accommodate a company's expanding needs is a drawback of this replication technique. [16]

Depending on the volume of client inquiries, **the dynamic scheme** either creates or removes new replicas on the fly without restarting the system. [17]

You can boost throughput by expanding the number of active servers. As long as synchronization is maintained between a newly added replica and the other replicas, the database administrator can change the number of replicas in a cluster. Geographically distributed systems, where any number of new replicas can be formed in each local zone, benefit the most from the usage of dynamic schemes.

Dynamic methods, however, have the following drawbacks:

finding the physical addresses of additional servers that contain replicas and their importance, This multiplies the messages sent and received by the servers [18]

Fragmentation distributed database

In a distributed database system, "fragmentation" describes how data is physically dispersed among a number of network nodes or storage devices. Performance may suffer as a result of the need to retrieve data from numerous nodes in order to satisfy a query. [10]

- A distributed database system experiences two basic types of fragmentation: Horizontal fragmentation: This sort of fragmentation happens when data is divided over numerous nodes according to a certain attribute or collection of attributes, such as when customer data is kept on different nodes for each region. [19]
- Vertical fragmentation: Based on the burden of the queries and the patterns of access, this sort of fragmentation happens when only a portion of the columns or attributes of a database are kept on a certain node. [19]

Database managers may employ data partitioning techniques, such as hash-based or range-based partitioning, to make sure data is distributed equally across nodes in order to reduce fragmentation in a distributed database system. To maintain high availability and lessen the effect of node failures, they might additionally employ replication or data redundancy strategies. [9]

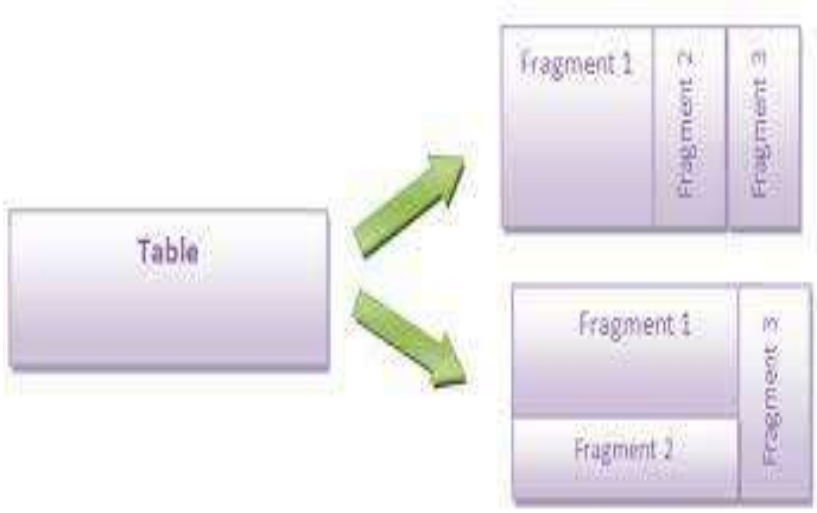


Figure of fragmentation (1)

When physical data is saved in a database, it may be fragmented, which means that it is not stored in a contiguous fashion over one or more disk pages or disk blocks. This can result in decreased speed and lengthy access times, particularly when a database has grown over time and there is not enough contiguous disk space to store newly added data. [20]

Allocation Distributed Database

Allocation in a distributed database refers to the distribution of data among a number of network nodes. Depending on different factors, such as data size, access habits, or data relevancy, this may be done. Allocating resources aims to balance the demand on each node, boost the database's performance, and increase availability and fault tolerance. [21]

When a database is distributed among numerous nodes in a network, allocation is the process of breaking the database up into smaller, more manageable chunks. This enables greater performance, increased availability, and fault tolerance in addition to better resource use. Sharding,

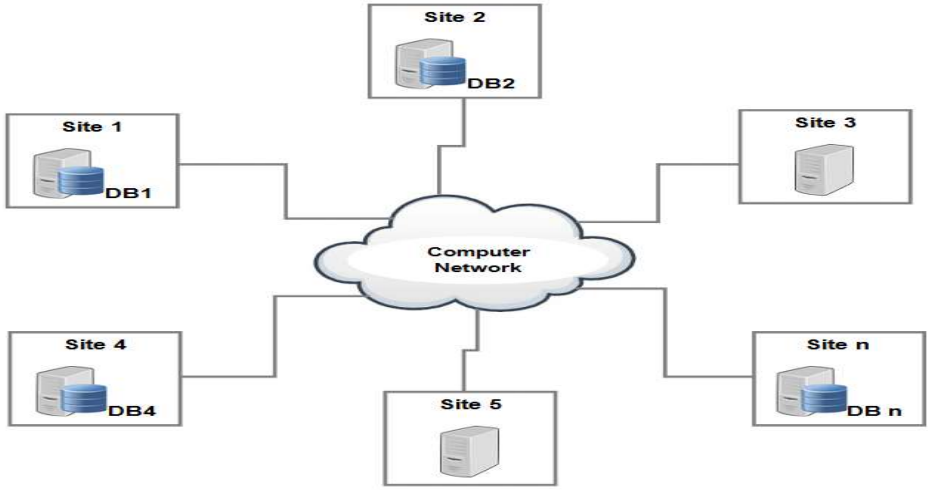
segmentation, and replication are only a few of the allocation techniques available in a distributed database [21]

According to a particular criterion, such as data quantity or access patterns, the database is partitioned into smaller portions using a sort of allocation called sharding. When a user searches the database, the query is routed to the proper node that houses the data because each shard is stored on a different node. Sharding can increase the database's overall performance by lowering the burden on any one node. [22]

Another allocation method is partitioning, which divides the database into smaller chunks based on the data. Partitions are created from the data, and each partition is kept on a different node. By enabling numerous nodes to handle data simultaneously, partitioning can enhance database performance. When the amount of data is too great to fit on a single node, this type of allocation is frequently utilized [22]

The third method of allocation is replication, in which different copies of the database are kept on different nodes. This kind of allocation is utilized to increase the database's availability. The data can be recovered from another node if one node fails. By enabling many nodes to handle data concurrently, replication also enhances database performance. [22]

A distributed database's allocation process is an essential one that balances the load on each node, enhances performance, and increases availability and fault tolerance. A hybrid solution that satisfies the particular needs of the database can be produced by combining several allocation techniques, like as sharding, partitioning, or replication. [20]



Figure(2) allocation node in DDBS

conclusion

recent research has produced substantial advancements in the field of fragmentation and allocation, which continues to be a crucial replication approach for distributed databases. The possibilities of fragmentation and allocation techniques have been enhanced by developments in IOT, cloud computing, big data analytics, deep reinforcement learning, blockchains, machine learning, and energy efficiency, opening up new directions for further study. More effective and economical fragmentation and allocation techniques are required as the size and complexity of distributed databases increase in order to maintain high levels of performance, increase the speed, dependability, and scalability, to determine their efficacy in other fields and to create more successful fragmentation-based replication techniques, more research is required.

References:

[1] X. X. Y. &. W. Y. Li, "Fragmentation-based data replication strategy for big data analytics. Information Processing & Management," p. 58, 2021.

[2] H. Z. Y. C. W. &. C. J. Chen, " A fragmentation-based data replication scheme for cloud storage systems. Future Generation Computer Systems," pp. 500-508, 2021.

[3] M. L. P. GORT, "Relational data watermarking resilience to brute force attacks in untrusted environments," 2020.

[4] Q. L. J. &. W. X. Zhou, "fragmentation-based data replication scheme for wireless sensor networks in IoT," 2020.

[5] H. Z. Y. C. W. &. C. J. Chen, "A fragmentation-based data replication scheme for cloud storage systems," 2020.

[6] L. e. al, "fragmentation-based data replication approach," 2020.

[7] X. W. e. al., "Deep Reinforcement Learning Approach for Dynamic Fragmentation and Allocation in Distributed Databases," 2021.

[8] X. L. Y. L. a. Y. W. Y. Liu, "Comparative Study of Fragmentation and Allocation Techniques for Distributed Databases," 2021.

[9] Y. L. a. X. L. S. Zhang, "Efficient Fragmentation and Allocation in Replicated Databases using Blockchains," 2022.

[10] S. Z. a. X. L. Y. Wei, "A Hybrid Approach to Fragmentation and

Allocation in Replicated Databases using Machine Learning," 2022.

[11] J. L. X. L. a. Y. W. Z. Li, "An Energy-Efficient Approach to Fragmentation and Allocation in Replicated Databases".

[12] L. Moore, "distributed database," *ORACLE*, 2020.

[13] M. A. Aslam, "Replication in Cloud Databases: State of the Art and Research Challenges," 2021.

[14] B. Pocze, "MySQL Replication: Advanced Features and Best Practices," 2021.

[15] L. C. S. S. M. &. F. R., "A survey on database replication techniques," 2020.

[16] M. A. &. M. M. El-Nasr, "Multi-Source Data Replication Techniques in Distributed Database Systems," 2021.

[17] Y. K. K. H. J. &. L. D. H. Kim, "Design of Database Replication System Using Cascading Replication," 2021.

[18] T. Ö. a. P. Valduriez, "Distributed Database Systems," 2021.

[19] Z. e. al, "fragmentation-based data replication strategy for the Internet of Things".

[20] S. B. S. S. a. S. L. S. Garg, "A survey of data allocation strategies in distributed database systems," 2021.

[21] L. L. X. &. C. L. Zhang, " Research on Data Sharding and Load Balancing Algorithm in Distributed Database System. Wireless Personal Communications," 2021.

[22] VNIT, N. (2018). Distributed database management system (DDMS). International Journal of Computer Applications, 173(8).

[23] H. W. e. al.