KeAi
CHINESE ROOTS
GLOBAL IMPACT

International Journal of Innovation Studies

IMPROVEMENT IN SERVER PERFORMANCE AND AVAILABILITY USING MULTI-CLUSTER LDAP SERVERS AND IMPROVED REPLICATION SYSTEM

Dr. Suhaas K P[1]
Assistant Professor
Department of Information Science and Engineering
The National Institute of Engineering, Mysuru, INDIA
suhaaskp@nie.ac.in

Dr. Shashidhara H R[2]
Associate Professor
Department of Electronics and Communication
The National Institute of Engineering, Mysuru, INDIA
shashidharahr@nie.ac.in

Abstract — A big issue in the storage of data and the capability to re-establish them, while maintaining the elements of performance and robustness if a particular event does not favour it. Storing mechanisms prove to be very useful during utilization of the stored data to meet the expected standards. Solution for the fault tolerance can be provided by storing multiple copies or replicas of the data where the lost data can be recovered from another replica. This mechanism can impact on the performance of the system. A good storage mechanism that is flexible to the desired expectation of raising the fault tolerance level without compromising on the performance is imperative. This paper outlines one such system which mainly addresses the above said expectation with the use of certificates for authentications in the architecture.

Keywords — Lightweight Directory Access Protocol server (LDAP), Certificate Authority (CA), RAID Levels, Transport Layer Security (TLS), IPSecurity (IPSec), Secure Socket Layer (SSL), StrongSwan, X.509.

I. Introduction

A storage server plays a fundamental role in any data-based infrastructure, serving as the backbone for storing, managing, and accessing vast amounts of data. These storage servers are essential for ensuring that data is securely housed and readily available to entities that need to access it, whether for routine operations or more critical tasks. The efficiency of a storage system depends not only on its capacity to store data but also on its ability to ensure that the data remains accessible, even during periods of high demand or network congestion [1].

In modern architectures, storage servers are often interconnected, allowing seamless read/write operations across various units. This interconnection is made possible through a Storage Area Network (SAN), a high-speed, dedicated network designed specifically to link storage devices with processors and servers. Unlike traditional methods of connecting storage to servers, SANs introduce a more advanced, efficient approach. The speed and scalability of SANs are comparable to Local Area Networks (LANs), enabling faster data access and better resource allocation [2]. SANs have a wide range of applications that are critical to enterprise-level data management. They are commonly used for disaster recovery, ensuring that data remains protected and can be restored in case of system failure. SANs also facilitate data protection, sharing, and create a centralized storage hub for data vaults and backups, safeguarding against data loss or corruption. Additionally, the architecture of SANs makes it easier to manage large volumes of data across multiple servers, increasing the system's overall efficiency and reliability.

A key feature of storage servers within SANs is their use of Redundant Array of Independent Disks (RAID) technology, which enhances both performance and fault tolerance. RAID works by distributing and duplicating data across multiple disks, allowing the system to continue functioning even if one or more disks fail [3]. Different RAID levels offer varying balances between performance and fault tolerance, enabling organizations to choose a configuration that best suits their needs. RAID ensures that data stored in the disk subsystems of these storage servers can be recalled quickly and efficiently, even under challenging conditions.

Fault tolerance, in this context, refers to the system's ability to recover from hardware or software failures and to reconstruct lost or corrupted data. Higher levels of fault tolerance mean the system can handle more severe failures without compromising data integrity. Performance, on the other hand, measures how effectively the system operates under various conditions, such as high traffic or heavy data loads. A storage system with high fault tolerance and performance ensures that data remains accessible, secure, and recoverable, even in the face of challenges like hardware malfunctions or network interruptions. The greater the fault tolerance and performance of a storage system, the more reliable and acceptable it becomes for critical applications. Systems with high fault tolerance reduce the risk of data loss, which is particularly important for disaster recovery and backup solutions. At the same time, high performance ensures that the system can handle large-scale operations, including real-time data processing and sharing, without experiencing delays or inefficiencies [4]. As a result, storage servers and SANs that optimize both of these metrics become indispensable components in any robust, scalable data infrastructure.

It is impossible to overemphasize the significance of security services in a network that includes secure data, whether it is a local, regional or global network. In cryptographic security service, the process of vouching the claimed identity of one entity by another entity that is also involved in providing security service is referred as authentication. These authentication mechanisms shall needs to be implemented at the points where it connects or communicates with its peer entity [5]. Certification is employed for verifying a number of network entities. The certificate is an additional file to the electronic message that is used for the security needs only. To send an

encrypted message one needs a digital certificate, which has been provided by the Certificate Authority (CA). The most popular type of standard for the digital certificate is known as the X. 509. The type of certificates that are put in servers is checked each time there is an association with the latter. If certificates are un-trusted then the data is not safe and the connection is described as insecure. There is the need to enhance security authentication in storage area network to improve the overall integrity of the storage area architecture.

II. Existing Methodology

The current architecture of storage area network is based on the storage of the data within the disk subsystem with the objective of providing efficient means of implementing fault tolerance and optimal performance factor [6], [7], [8]. The storage architecture remains linked with the outside IT architecture in a manner that forms a network. At the internal level of the storage system, there are servers in which each server is made up of the disk-subsystem where there are various physical hard disks. The actual processing of storing the data is done using Redundant Array of Independent Disks (RAID) levels. Some of the RAID mechanisms implemented are based on the main design developed with RAID 4 and RAID 5 which employ the method of parity of data blocks placing them on a physical hard disk within the disk subsystem [9]. Primary and general objective to adapt the RAID levels is to enhance I/O operations and ensure greater degree of reliability. It is within the disk subsystem that multiple copies/replicas of data are maintained. Also, there exists a concept of remote mirroring in which is used to create and maintain the same data at different geographical locations in a synchronized manner [10]. Although the synchronisation is used to achieve consensus, it serves as a disadvantage at certain times. Considering the advantage of synchronisation, in the case of a total breakdown of the primary centre, the secondary centre should supply the requests. But, a third, a governing entity is needed for overseeing these two centres both of them. Under normal circumstances, it is even observed that three replicas of the data are created in such a distributed networking system [11]. In such servers, there exists a disk subsystem that is present inside a distributed server, and any loss of the entire server can lead to the complete failure of all the data, regardless the number of replicas inside them. Here, it is mostly used when there is an emphasis on fault tolerance for one specific disk subsystem of a server. Consequently, the failure case leads to the low fault tolerance for the entire server [12], [13].

From the security point of view it is also pertinent to choose right authentication enabled on server to ensure right access to data by the peer server in the mirroring architecture and in client server architecture [14]. Therefore it is necessary to have the data available during failures i.e. having a fault tolerant system with consideration of factor performance and a governing body to meet the targets..

III. Proposed Methodology

In the proposed methodology, we utilize clusters with several LDAP servers, which are configured internally. The first significant justification for the adoption of LDAP server is the unprecedented read/write query speed for massive datasets. It also highlights the "single-logon" technique that enables the client to access the data and do not get a bottleneck each time the client is being authenticated. Moreover, it also facilitates Secure Socket Layer (SSL) and Transport Layer Security (TLS) to ensure only the intended recipient can comprehend it. The Figure 1 illustrates the internal configuration of the two clusters. A client system exists along with the 'Cluster Witness Server' that has a VPN connection to the cluster head of both clusters. The Cluster Witness Server participates in the functionality of the voting process, as to decide which LDAP node is to remain being part of the cluster in order to form the quorum. The cluster head is one among several LDAP servers present inside the clusters. These cluster heads are chosen based on any distributed system algorithm- Leader Election protocol.

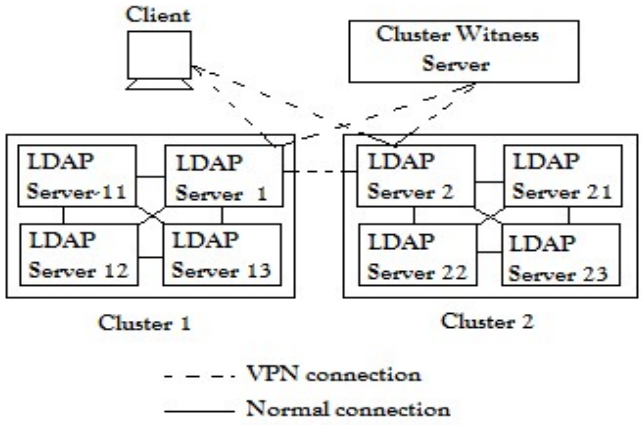

Figure 1: The Proposed Architecture

In other words, the chosen cluster head is also in charge of other data indexing mechanisms and is also a form of buffer when reading or writing data. That is, the data is stored in different LDAP servers that are integrated into the cluster. It is further ensured that there are at least two copies of data available in a cluster. Every other LDAP server in the cluster is connected to the LDAP leader server as is every other cluster. The other cluster also has LDAP servers and a LDAP leader server like in other cluster. This connection is made through the LDAP leader server of the other cluster. Therefore, the other cluster is similar to the first one and is known as the primary cluster and the secondary cluster. The two clusters also operate an updating mechanism so that everyone has the latest view of the data set. The relationship between the two clusters is established by the Virtual Private Network (VPN).

While a LDAP leader server supervises the operations within a cluster, a cluster witness server supervises the operations of both the clusters. The cluster witness server is accessible on VPN to

the LDAP leader server of both the clusters. A client is the one that requires the data to be collected and processed in order to get the reply to the requests. Whenever the client asks the servers for the data, the request is attended to by any of the two clusters provided that the two are up and fully functioning. The specific procedure of the servicing cluster is to update the status in the client witness server while the synchronisation operation is effected in the other non-servicing cluster. It ensures that the request is serviced by any of the two clusters which the cluster witness server belongs to. Therefore, when the primary cluster is handling the given request, the secondary server shall make use of sync mechanism and when it is the other way around. In case a particular LDAP server in a cluster is inaccessible, data is retrieved from other LDAP servers of the same cluster. If the cluster fails, the data is obtained from the other cluster and the cluster witness server prevents the defective cluster to attend to the request.

Algorithm:

Step I: Through VPN connection, client forwards the incoming service request to cluster 1 and cluster 2.
Step II: If both the cluster C1 and cluster C2 are active and functioning.
Step II.a: The cluster C1 being able to reply to the service request through LDAP leader node.
Step II.b: Modification in cluster witness server done by cluster C1.
Step II.c: Activate data synchronisation in cluster C2.
Else,
Step 2d: Proceed to Step III.
Step III: If cluster C1 is active and cluster C2 is unavailable,
Go to Step II.a and III.b.
Else-if cluster C1 is active and cluster C2 is inactive,
Goto Step IV.
Else,
Both clusters C1 and C2 are non-serviceable. Reply to the service request unserved.
Step IV: Service the request from cluster C2.
Step V: Upgrade in cluster witness server by cluster C2.
Step VI: End.

In the implementation, a single-master and multi-master scenario is considered. Having such a choice of implementation, helps the application to be accessible even in case of prolonged failures depending on the nature of the application that is being hosted in the server. In case of the leader election scenario, the single-master approach is utilized. However, the single-master node is again prone to failures. In such cases, the multi-master selection criteria is considered where another level of RAID implementation is considered for increases fault tolerance.

Authentication in proposed methodology

The purpose of creating an authentication mechanism is to validate an action or endeavour by verification of the real thing. The authentication mechanism is one point that must be included in every entity of the proposed structure. It can be assumed that all the clusters below are with LDAP server and it goes without saying that the LDAP servers support SSL so certificates shall also be used for this purpose. Every LDAP server of a cluster shall have the certificates imported from CA. Because the internal LDAP servers are networked, they create a session key from the public key cryptography. This session key is exclusively used during the communication within and outside the cluster. The internal LDAP servers also holds the certificates imported from the Certificate Authority (CA). Each LDAP server shall have a trust store and a key store. The trust store holds certificates from external bodies requesting to or from the CA to identify other parties who are expected to identify others. Key store holds private keys and the particular certificates bearing the public key.

A Virtual Private Network (VPN) has been used to implement the communication between the two clusters. For this purpose, it uses *StrongSwan*, which is an IPSec for Linux. Almost like IPSec which this is an internet protocol suite that adds security to the IP protocol whereby each IP packet is authorized and encrypted during transmission. Users can use web certificates that are used for secure webserver to webbrowser communication on the client side. Connection to the web server allows the user to get data stored in it using the secure https.

IV. Results and Discussion

As the LDAP servers are already meant for the improvement of fault tolerance, the inclusion of authentication plays a major role in enhancing the security along with the performance. The key metrics for the comparison include the extent of data availability, Recovery Point Objective (RPO), that indicates the time allowed for a node to be in the inoperative state after a failure and Recovery Time Objective (RTO) that states the maximum loss value that a server can withstand before failure.

To evaluate the scenario, the servers needs to have higher data availability. This is mainly supported due to replication. Further, the servers need to have smaller RTO and RPO. This evaluation metrics are in consideration with the single-master or multi-master replication topology. The resultant graph illustrating the data availability, RTO and RPO is shown below for the percentage values denoted in the y-axis. The comparison has been made with the LDAP server and non-LDAP server.

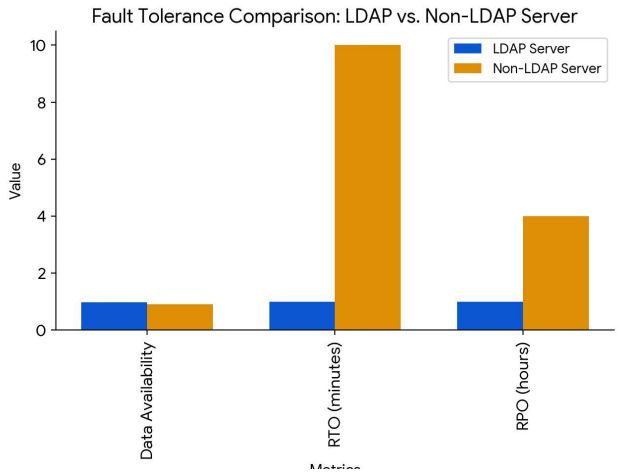

Figure 2: Fault Tolerance Comparison

In the Figure 2, the graph represents the value (represented as %) for data availability, RTO and RPO for both LDAP and non-LDAP servers. It can be observed that the data availability is higher for the LDAP server. This has been attributed to the replication of LDAP servers within the cluster. Further, with the use of multiple clusters and frequent updates in data through the synchronisation between these clusters, the availability of data is higher in the LDAP server. However, a slight increase in the percentage is attributed to the use of cluster-heads that proves an small advantage edge compared to the non-LDAP server. In reference to the RTO, the proposed LDAP server architecture proves lesser time in minutes. Due to the increased availability of data, the time taken to for any application or business process that can stay offline before the LDAP server within a cluster or the complete cluster is restored, is higher. In other words, with the use of the proposed architecture to deploy any business application, the restoration can take time before the complete data unavailability. In reference to the RPO, the proposed architecture displays higher RPO that indicates the extent of data loss that the application or the business can withstand in case of unavailability of LDAP servers/clusters. The reason is contributed to the use of the replication and mirroring model used in the proposed architecture.

Also, it is also utmost necessary to clearly strike the balance between maintaining the availability, RTO and RPO depending on the nature of the application in consideration. These metrics can be suitably optimized to work based on the requirement and nature of application.

As the throughput is concerned with the proposed method, the Figure 3 represents a graph for comparison of throughput and response time for both LDAP and non-LDAP server. The response time indicates the amount of time that the server takes to respond to the incoming request. Although the response time is highly dependent on various factors, the usage of the proposed methodology enhances the response time and better utilisation of the bandwidth. From the Figure 3, the

representation of the throughput and response time for both LDAP and non-LDAP server is shown. It can be clearly realized that both the parameters i.e., response time and throughput is better for the LDAP server. The proposed architecture provides lesser response time and higher throughput while, the non-LDAP server proves itself with higher response time and lower throughput.

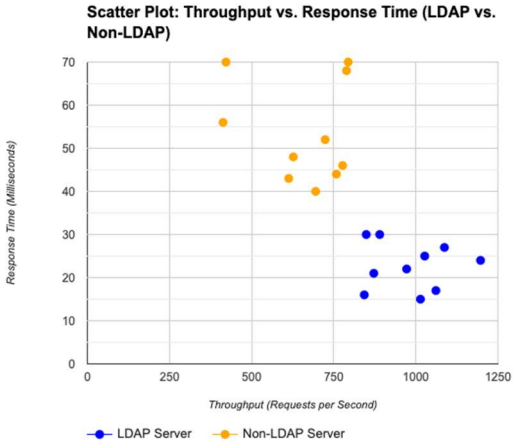

Figure 3: Throughput and Response Time Comparison

The proposed scheme improves the fault tolerance by a factor of four replicas with a two set per cluster strategy. This redundancy extends far from replicating the data in a cluster to replicating it in some other cluster. Therefore, the system can 'platform' a specific LDAP server, all the servers within a cluster, or even one or several clusters, since all the necessary information can be obtained from other replicas. It is noteworthy that this resilience is definitely beyond the fault tolerance when a single data set is in use, and equally or even worse when there is no replication at all.

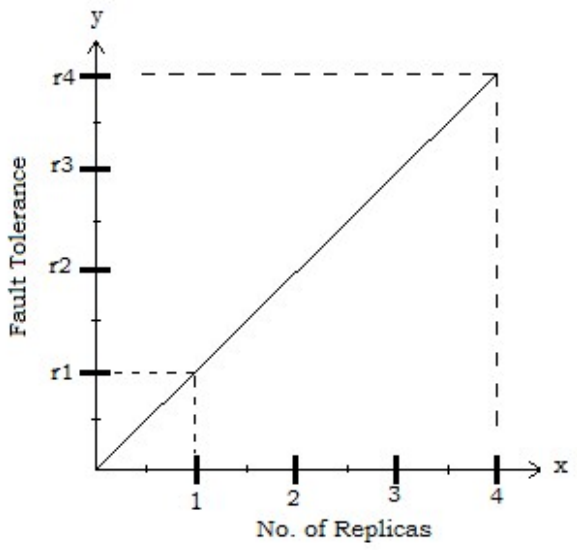

Figure 4: The Expected Graph

In the Figure 4, the graph shows the increase in the level of fault tolerance in association with the number of replicas. The number of replicas are directly dependent on the number of LDAP servers where each LDAP server can maintain more than one replica based on the RAID levels within them. In the figure, the y-axis represents the fault tolerance and number of replicas in the x-axis. The fault tolerance is represented as r_1, r_2, r_3,..r_n, indicating the level of increase in such fault tolerance levels corresponding to the number of increase in replicas. It can be observed that there is a linear increase in fault tolerance with increase in replica count.

As for the conventional disk-subsystem-based methods, data is relegated to a single server disk which results to bottleneck and single point of failure. However, in this system, replicas of data are propagated on various LDAP servers of the cluster as opposed to the disk of one server. This architectural change helps to avoid the shortcomings in using a single server for implementation, while making the system more robust. It is an excellent feature when data is split between numerous LDAP servers; the system remains operational even if one server is unavailable. The other peer LDAP servers can also independently continue with the data search and provides faster access to the data without a huge decline in the rate of access.

Furthermore, in the event of a failure, a server can trigger the recovery process on its own. This recovery process is separate from the read/write operations, which are performed on other operational servers within the cluster. In typical configurations, the recovery processes participate in the competition for system resources with concurrent read and write activities, which impairs system efficiency. It also eliminates resources contention since the recovery process is separated from these operations, thus enabling the failed server to recover at a faster rate. Therefore, it can be stated that the overall system is able to sustain a performance supremacy, in failure scenarios that may even be over concurrently, non-isolated conventional architectures.

In a system where database replication is implemented, data is duplicated and distributed across multiple servers, eliminating the reliance on a single server for data access. This redundancy offers a crucial layer of fault tolerance, ensuring that the system remains operational even in the event of individual server failures. With multiple servers holding the same data, there is no need to route all read and write operations to a single point, which drastically reduces the risk of downtime or performance bottlenecks. Instead, these operations can be dynamically dispatched across numerous LDAP servers, ensuring that the workload is balanced efficiently. This distributed approach not only enhances fault tolerance but also significantly improves the system's ability to handle large volumes of traffic. In high-demand environments, where the number of read and write requests can scale rapidly, having the ability to distribute these operations across multiple servers ensures that the system remains responsive and efficient. The load is shared among several LDAP servers, allowing for parallel processing, which in turn enhances the overall throughput. Each LDAP server

within the cluster can independently process a portion of the workload, reducing latency and improving the user experience, even during peak traffic times.

Moreover, the cluster-based architecture offers substantial benefits in terms of scalability. As data requirements grow or as more clients access the system, it becomes possible to seamlessly integrate additional LDAP servers into the cluster without needing to overhaul the underlying infrastructure. This scalability is key to the long-term adaptability of the system. The ability to incorporate new servers with minimal disruption allows for continuous expansion, ensuring that the system can keep pace with growing demands. The fundamental layout remains unchanged, so the architecture remains stable and secure even as new LDAP servers are added to meet the increased traffic and storage needs.

The ease of scaling in such a system also contributes to its resilience. By distributing the data and operations across a wide network of servers, the system becomes more robust against potential disruptions. For example, if one or several servers experience failure, the remaining servers can continue to function, ensuring uninterrupted service. This redundancy acts as a safeguard against data loss or downtime, enhancing the reliability of the system. In addition, it allows for maintenance or upgrades to be performed on individual servers without taking the entire system offline, further ensuring continuous operation.
Furthermore, this replication-based, cluster-driven architecture introduces a level of flexibility that traditional systems often lack. It allows administrators to allocate resources dynamically, scaling up or down based on real-time demand. For instance, in periods of lower demand, certain LDAP servers can be temporarily decommissioned to conserve resources, while in periods of high demand, additional servers can be spun up to meet the load. This dynamic adaptability not only optimizes resource usage but also ensures that the system can provide consistent performance under varying conditions.

V. Conclusion

The fault tolerance is an important evaluation parameter in assessing the reliability of the datacentre. The fault tolerance level indicates the availability of data in case of unavailability of service by the servers. In this proposed method, the LDAP servers are used to provide the replication and fault tolerance. Along with the replication, the major use of LDAP server is to build a centralised authentication server. This is used for authentication of users as well.

In the proposed system, the architecture involves the use of clusters, named as C1 and C2, that contains LDAP servers inside them. These internal LDAP servers are connected to each other through the network in both the clusters. Also, with the leader election protocol within the clusters, an cluster-head is chosen to maintain the synchronisation within the cluster. Further, the inter-

cluster communication is through a VPN. Monitoring these clusters, are the cluster witness server than aids in synchronisation of data from one cluster to another cluster. Whenever a client initiates the request, the request is sent to the cluster-head LDAP server. It shall be the responsibility of the cluster-head to forward the request to the internal LDAP servers. In a cluster, the data is replicated inside multiple servers (preferably the replication number is equal to the number of LDAP servers inside a cluster). Therefore, any data or application that exists in the server is replicated multiple times. In case of unavailability of any internal server inside the cluster, the cluster head immediately identifies the same and redirects the incoming traffic to other internal LDAP servers. In this proposed system, the number of LDAP servers are chosen to be 4 to be optimal in various other resource allocations. The same technique is followed by all the clusters present in the datacentre. Supervising the clusters are the Cluster Witness Server that maintains the cluster and keep them up and running. It is the responsibility of the cluster head to initiate synchronisation within the cluster and the responsibility of the cluster witness server to main synchronisation between the clusters.

The results of the proposed system clearly demonstrate a significant improvement in data availability, as well as in Recovery Point Objective (RPO) and Recovery Time Objective (RTO) metrics. Data availability is notably enhanced due to the distributed nature of the architecture, which reduces dependency on a single point of failure and allows continuous data access even during server downtimes. This is crucial for applications that require high uptime and need to minimize disruptions in service.

In terms of performance assessment, particularly with respect to RTO, the results show that the proposed architecture achieves a lower RTO when deploying applications. This lower RTO indicates that the time required to recover from a system failure is significantly reduced, allowing restoration processes to begin promptly, minimizing the duration of complete data unavailability. By shortening the recovery time, the system ensures that critical operations can be resumed faster, which is a key benefit in environments where downtime directly impacts business operations or user experience.

With regard to the RPO, the proposed architecture demonstrates a much higher tolerance for data loss before recovery is required. A higher RPO means that in the event of system failure, the architecture can afford a longer window of data loss that the business or application can withstand without serious consequences. This feature is particularly important for applications handling sensitive or critical data, as it ensures that the system can recover from disruptions with minimal loss of important information, maintaining operational continuity. By optimizing both RPO and RTO, the architecture strikes a balance between minimizing data loss and reducing recovery time, which makes it highly effective in disaster recovery scenarios.

Beyond RPO and RTO improvements, the proposed model also exhibits superior performance in terms of throughput and response time when compared to non-LDAP server architectures. Throughput is improved due to the distributed nature of data storage and replication, allowing multiple LDAP servers to handle read and write requests concurrently. This parallelism results in increased data processing rates, enhancing the system's ability to manage larger workloads more efficiently. Likewise, the reduction in response time is a direct outcome of this distributed workload, as the load is shared across multiple servers, decreasing the latency experienced by users. In high-traffic environments, this reduced response time translates into a smoother, faster user experience and ensures the system can handle spikes in demand without degradation in performance.
In relation to fault tolerance and the number of replications, the results of the proposed system reveal a linear increase in fault tolerance as the number of data replications increases. This linear relationship suggests that with each additional replication of data across LDAP servers, the system becomes more robust and resilient to server failures or data corruption. The higher the number of replications, the greater the system's ability to recover from disruptions, providing an additional safeguard against data loss. This increase in fault tolerance validates the strength of the proposed methodology, confirming its robustness in maintaining system reliability even in the face of potential issues like server crashes or network failures.

Overall, the proposed system offers considerable advantages in terms of data availability, performance, fault tolerance, and disaster recovery capabilities. By optimizing key metrics like RPO and RTO, and ensuring better throughput and response times, the architecture provides a comprehensive solution that enhances system reliability and robustness. The ability to increase fault tolerance through replication further strengthens the system's resilience, making it a viable and superior alternative to non-LDAP server architectures in mission-critical environments.

References

[1] G. A. Gibson *et al.*, "A cost-effective, high-bandwidth storage architecture," in *Proceedings of the Eighth International Conference on Architectural Support for Programming Languages and Operating Systems*, in ASPLOS VIII. New York, NY, USA: Association for Computing Machinery, 1998, pp. 92–103. doi: 10.1145/291069.291029.

[2] Riabov and Vladimir V, *Handbook of Computer Networks: Distributed Networks, Network Planning, Control, Management, and New Trends and Applications*, vol. 3. John Wiley, 2007.

[3] A. Ma *et al.*, "RAIDShield," *ACM Transactions on Storage*, vol. 11, no. 4, pp. 1–28, Nov. 2015, doi: 10.1145/2820615.

[4] D. D. Chambliss *et al.*, "Performance virtualization for large-scale storage systems," in *22nd International Symposium on Reliable Distributed Systems, 2003. Proceedings.*, IEEE Comput. Soc, pp. 109–118. doi: 10.1109/RELDIS.2003.1238060.

[5] Hahnsang Kim, K. G. Shin, and W. Dabbous, "Improving Cross-domain Authentication overWireless Local Area Networks," in *First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM'05)*, IEEE, pp. 127–138. doi: 10.1109/SECURECOMM.2005.24.

[6] Thomas M. Ruwart, "Disk Subsystem Performance Evaluation: From Disk Drives to Storage Area Networks," in *17th IEEE Symposium on Mass Storage Systems / 8th NASA Goddard Conference on Mass Storage Systems and Technologies*, Minnesota: University of Minnesota, 2000.

[7] Nicos A. Vekiarides, "Implementation of a Fault-Tolerant Disk Storage System Using Reflective Memory," M.Sc Project, Carnegie Mellon University, 1995.

[8] P. Bijaoui and J. Hasslauer, "Storage Technologies," in *Designing Storage for Exchange 2007 SP1*, Elsevier, 2008, pp. 75–116. doi: 10.1016/B978-1-55558-308-8.00003-X.

[9] Wiebalck Arne, "ClusterRAID: Architecture and Prototype of a Distributed Fault-Tolerant Mass Storage System for Clusters," Kirchhoff Institute for Physics, Germany, 2005.

[10] R. Yan, J. Shu, and D. Wen, "An Implementation of Semi-synchronous Remote Mirroring System for SANs," 2004, pp. 229–237. doi: 10.1007/978-3-540-30207-0_29.

[11] Lili Qiu, V. N. Padmanabhan, and G. M. Voelker, "On the placement of Web server replicas," in *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No.01CH37213)*, IEEE, pp. 1587–1596. doi: 10.1109/INFCOM.2001.916655.

[12] W. Bartlett and L. Spainhower, "Commercial fault tolerance: a tale of two systems," *IEEE Trans Dependable Secure Comput*, vol. 1, no. 1, pp. 87–96, Jan. 2004, doi: 10.1109/TDSC.2004.4.

[13] M. Marwah, S. Mishra, and C. Fetzer, "Enhanced server fault-tolerance for improved user experience," in *2008 IEEE International Conference on Dependable Systems and Networks With FTCS and DCC (DSN)*, IEEE, 2008, pp. 167–176. doi: 10.1109/DSN.2008.4630085.

[14] B. C. Popescu, M. van Steen, and A. S. Tanenbaum, "A security architecture for object-based distributed systems," in *18th Annual Computer Security Applications Conference, 2002. Proceedings.*, IEEE Comput. Soc, pp. 161–171. doi: 10.1109/CSAC.2002.1176288.