HNSJ Humanities & Natural Sciences Journal
ISSN: (e) 2709-0833
www.hnjournal.net

Peer-Reviewed Journal
Indexed at: NSP - SJIF
Impact Factor 2022 = 4.91

RESEARCH TITLE

Remote Systems: Building up a Distributed Architecture

Myasar Salim[1] Ali Abdallah[2]

[1] Faculty of Engineering, Computer Communication, Islamic University, Beirut, Lebanon

Email: myasar11sa@gmail.com

[2] Faculty of Engineering, Computer Science Department, American University of Science and Technology (AUST), Beirut, Lebanon.

Email: abadallah@aust.edu.lb

HNSJ, 2023, 4(10); https://doi.org/10.53796/hnsj41023

Published at 01/10/2023 **Accepted at 12/09/2023**

Abstract

Due to rapid technological advancements, organizations and institutions started to face challenges related to efficiency management at their distributed locations while maintaining centralized control. This work explores a comprehensive case study focused on the transition from a centralized operational architecture to a distributed network architecture. It tries to establish a synchronized system across different locations. The central location serves as the focal point, while the other locations are linked and integrated into the newly distributed framework. try to address the intricacies of synchronizing critical components and services including file transfer management, remote database access record management, and active directory services. Leveraging a combination of innovative technologies and well-defined strategies, the work outlines the steps and methodologies employed to ensure seamless integration, data consistency, and uninterrupted services across remote locations. identify the encountered challenges and identify proper solutions to mitigate potential bottlenecks and operational disruptions. through a comparative analysis of the centralized and distributed systems. highlight the valuable insights and practical recommendations for organizations seeking to embark on a similar journey of transforming their centralized systems into a distributed network, fostering a more agile and resilient operational infrastructure capable of meeting the evolving demands of the modern era.

Key Words: Distributed systems, Domain controller, Active directory, Distributed file system (DFS), Database replication.

I. INTRODUCTION

In today's interconnected world, organizations often find themselves grappling with the need to strike a delicate balance between centralized and distributed system topologies. The ever-evolving landscape of technology and business demands a new approach to operational structure, one that can seamlessly link multiple remote locations and branches dispersed across different cities, countries, or regions. This work embarks on a journey of transformation, wherein a centralized system is reconfigured into a distributed network, thereby synchronizing critical components such as files, databases, and active directory services across four remote locations [1].

The central branch, which is considered the hub of operations, becomes the fulcrum upon which this transformation pivots. The remaining three other remote locations or branches, which are geographically dispersed or operate under different administrative jurisdictions, are brought into the fold, creating a networked ecosystem that is capable of adapting to the demands of the modern age.

The transition from a centralized to a distributed system is not a mere technological shift; it is a strategic undertaking with far-reaching implications. It involves a meticulous blend of cutting-edge technologies, streamlined processes, and meticulous planning. We try to set out to explore the intricacies of this transformation, shedding light on the methodologies, challenges, and solutions that come into play [2].

Key among the challenges is the seamless synchronization of vital systems, including files, databases, and active directory services, while ensuring uninterrupted services for end-users. As we delve into the case study presented herein, we aim to unravel the strategies deployed to address these challenges and create a harmonized operational environment. This research endeavor to provide valuable insights and practical recommendations for organizations contemplating a similar transformation journey [3].

A distributed cluster, called distributed computing or a distributed database, is a network of physically independent nodes that are connected other using a distributed middleware. They work together synchronously and appear as a single system to the end user, Figure (1).

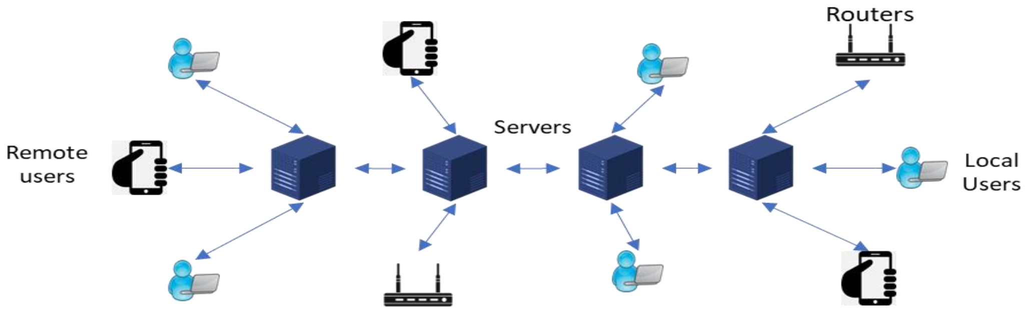


Figure 1 A network of physically independent nodes that are interconnected using distributed middleware.

Users all over the globe are using the internet to accomplish their jobs through web browsers. Online servers are the procedures that exchange documents with the global community and therefore numerous clients or users. Facebook and Google are good examples of online services that receives millions of daily requests. Serving millions of remote people requires a solid infrastructure [4]. Distributed systems usually distribute the load or clients' requests and manage the traffic to keep the system efficiently functioning. One of the great advantages of such an approach is that the system handles the workload and distributes it among active devices or servers. The system keeps running when one or more of the servers go down, they simply redistribute the load among active servers or devices.

This work will investigate, propose, and implement a distributed system based on four different remote locations. The proposed system will analyze, design, and implement different remote services including domain controller, active directory, database, and online application. High traffic analysis will be monitored and reduced, redundant links will be considered to improve performance and maintain availability and scalability.

II. RELATED WORK

Thousands of distributed websites provide information, using a distributed database system, which is a collection of multiple interconnected databases distributed across a network. Using a distributed data environment and client/server applications allows reliability, scalability, and consistent access to information, and improves connectivity in data processing across different network locations. Figure (2) shows a group of remote locations participating in the database to provide information [5].

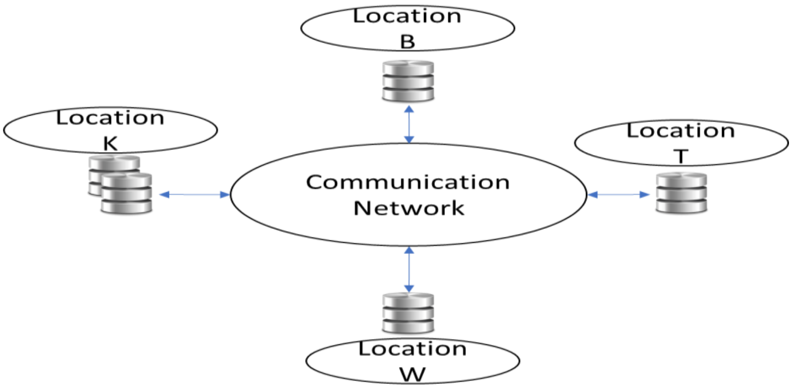


Figure 2 Distributed DBMS Environment

Active Directory (AD) is used as an authentication and authorization service in organizations to control user access and network resources. AD is a Microsoft technology that provides a centralized database of user accounts and security information, allowing administrators to manage access to resources throughout the network [6].

Data replication is the process of creating and maintaining multiple copies of the same data in different locations or databases. In the context of a single database server used for data entry and reporting, data replication can bring several advantages including performance improvement, data availability, disaster recovery, scalability, and localized access [7]. Three different types of database replication techniques are available including snapshot replication, transactional replication, and merge replication. Choosing the right replication method depends on the frequency of data updates, network bandwidth, latency requirements, and data consistency. SQL Server provides several features and technologies that can be used to achieve mid-level database caching in a multi-tiered environment. One of the main approaches to achieving transparency in caching is through the use of rendered views and distributed queries. SQL Server supports the execution of distributed queries, which allows to access data from multiple servers or databases as if they were part of a single database. In the context of mid-level database caching, we can configure an intermediary server to partially copy data from the back-end database server using replication or other mechanisms [8]. The three types of replications commonly used in relational databases are Snapshot Replication, Merge Replication, and Transactional Replication.

The results obtained from analyzing the execution performances of these replication types can be valuable in various research fields. Adaptive e-learning applications can benefit from efficient and scalable data replication to ensure consistent access to educational resources.

Medical diagnosis systems can leverage real-time updates and data consistency provided by transactional replication for accurate and up-to-date patient information. Artificial intelligence applications often rely on large datasets, and replication performance analysis helps optimize data availability and reduce access latency. In business management, replication performance insights can assist in ensuring timely and consistent access to critical business data across multiple locations, enabling efficient decision-making. Figure (3) shows Snapshot replication.

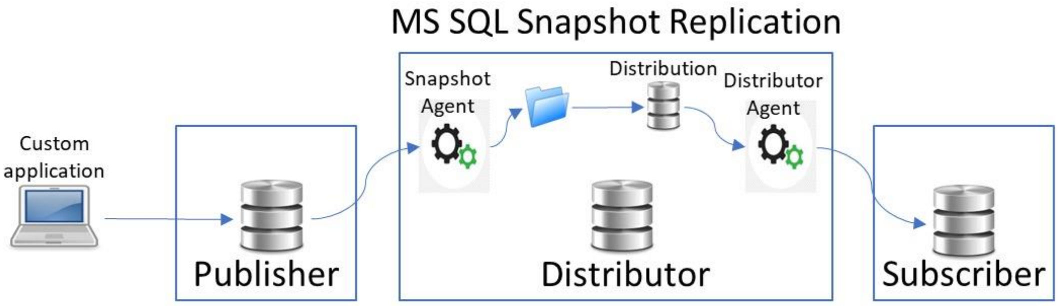


Figure 3 MS SQL Snapshot Replication

Designing replicated file systems can be done by utilizing a high-throughput distributed database system for metadata management. This approach offers improved scalability for the metadata layer of the file system by partitioning and replicating file metadata across a cluster of independent servers. Operations on file metadata can be transformed into distributed transactions, enabling better scalability and consistency [9]. One significant advantage of this approach is that the file system can support standard file system semantics, including fully linearizable random writes by concurrent users to any byte offset within the same file [10].

The analysis of the change of tracking possibilities in Active Directory Domain Services (ADDS) and their usage in the application, allows us to explore various techniques and mechanisms available in ADDS to monitor and track changes, such as event logging, replication, and directory synchronization. Additionally, code manageability is emphasized, indicating that the implementation of the application follows best practices for writing clean, modular, and maintainable code. This helps future users of the application understand its structure and configuration, making it easier to modify and adapt according to their specific requirements [11].

III. Distributed SYSTEM ARCHITECTURE

The proposed system design will include four different locations, which we call the main location (K) and the three other locations (B, T, and W). All locations are connected using microwave links with fixed IP addresses. Typically, all remote locations are connected to the main location (K) through the microwave link. Each location has its routing table. The implementation of such a distributed system will improve the performance and make it more scalable in terms of users and requests Figure (4).

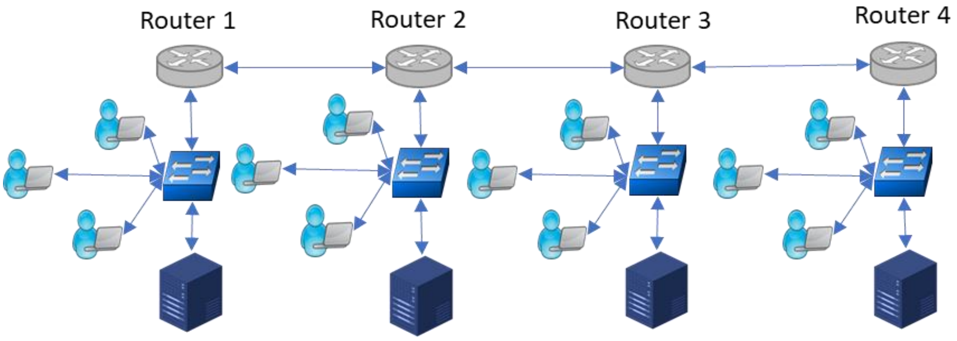


Figure 4 Four remote locations containing servers and routers

Replication, which is a storage process, will be applied to separate redundant copies and provide reliability. In case of database server failure, a redundant database will be activated to continue the functionality of the main one. Switching to another available and redundant database located at one of the three other locations allows for a reduction of network load and ensures system availability based on database local copies. Moreover, during peak hours, accessing the main database server could be inconvenient due to the huge load. The proposed system can monitor the load and ask other locations to use their local copies of the database instead.

3.1 System software components

The proposed architecture will include Windows Server 2012 R2, and Microsoft SQL Server. Windows Server was released on October 18, 2013. It contains storage tiering which allows storage and performance optimization. The work folders characteristic enables users to synchronize their work files between their devices and the corporate network, to enhance productivity and data availability. Windows servers also include Automatic Virtual Machine Activation (AVMA) which allows datacenter edition virtual machines to be automatically activated when running on a datacenter edition host without requiring a separate product key for each VM.

Microsoft SQL server is a database management system (RDBMN) designed to store, manage and retrieve data efficiently. SQL server is reliable and scalable and can handle thousands of concurrent users. Data integrity, security, high availability, and disaster recovery are all features of SQL servers. It allows Integration with Microsoft products including the .Net framework and Azure cloud platform.

3.2. Simulation components

Cisco Packet Tracer is a network simulation tool developed by Cisco Systems. It allows users to create virtual network topologies to simulate the behavior of computer networks [12]. Using this tool, we will design, configure, and troubleshoot our proposed distributed network in a risk-free virtual environment. VMware Workstation will be used to run multiple operating systems in a single physical device [13]. We use VMware Workstation to create virtual machines (VMs) that run a variety of guest operating systems, including Windows, Linux, and macOS. These VMs are isolated from the host operating system. VM allows us to test our software and run multiple applications without interfering with the host environment. Graphical Network Simulator (GNS3) is a network simulator used to design, simulate, and test complex network topologies. GNS3 is used to create the distributed virtual network and to simulate and troubleshoot it. It also allows creation and integration of hybrid networks with virtual and physical components.

3.3. System design

The proposed distributed system is distributed among four different locations. The main location (K) contains a main centralized server with an Active Directory (AD), a Distributed File System (DFS), and an SQL Server installed. The three other locations (B, T, and W) are considered remote locations and are connected to the main location (K). All three remote servers have a copy of the AD, a replication of the distributed files, and a replication of the SQL database installed and synchronized with location K as shown in Figure (5).

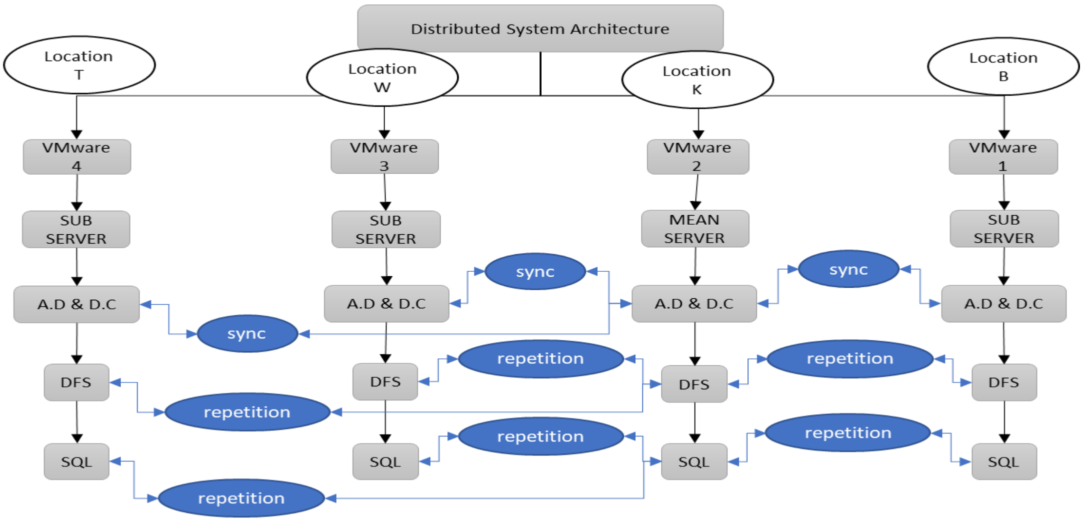


Figure 5 Synchronization and replication between servers between K and B, T and W

The remote locations are all connected to the main location using microwave links. The routers are responsible for routing all the requests and data from one location to another. All installed ADs and DCs at the main and the remote locations are synchronized. The AD and DC store information about the network resources including users, computers, and printers in a central database. Both AD and DC use a multi-master replication model, which saves and replicates all the changes made to a specific database over all other ADs and DCs to ensure synchronization between different remote locations [14]. The AD replication process is responsible for replicating changes made to the AD database on one domain controller to all other domain controllers in the same domain using the pull-based replication model. Different factors can affect the synchronization process including replication topology, network bandwidth, software configurations, domain controller hardware, and the amount of data being replicated. The proposed system installs additional Domain Controllers (ADC) and Child Domain Controllers (CHDC) at different remote locations.

Figure (6) illustrates the proposed domains at the main and the remote locations. The main server holds all permissions at the main Active Directory at K and all sub–Active Directory are installed on the sub-servers at B, T, and W. The process of replicating data across multiple nodes or servers in a distributed file system improves data availability, fault tolerance, and performance in the proposed distributed system [15]. Replication ensures that multiple copies of data are available on different locations, to allow users to access data locally or remotely in case of one location is disconnected. This allows network traffic reduction and improves performance by releasing loads on different locations [16].

Implementing SQL replication tools at different locations allows to create and maintain copies of a database in these locations. SQL Snapshot and Transactional replications were installed at all locations. they enable the synchronization of data between servers installed at K, B, T and W, and ensure availability [8].

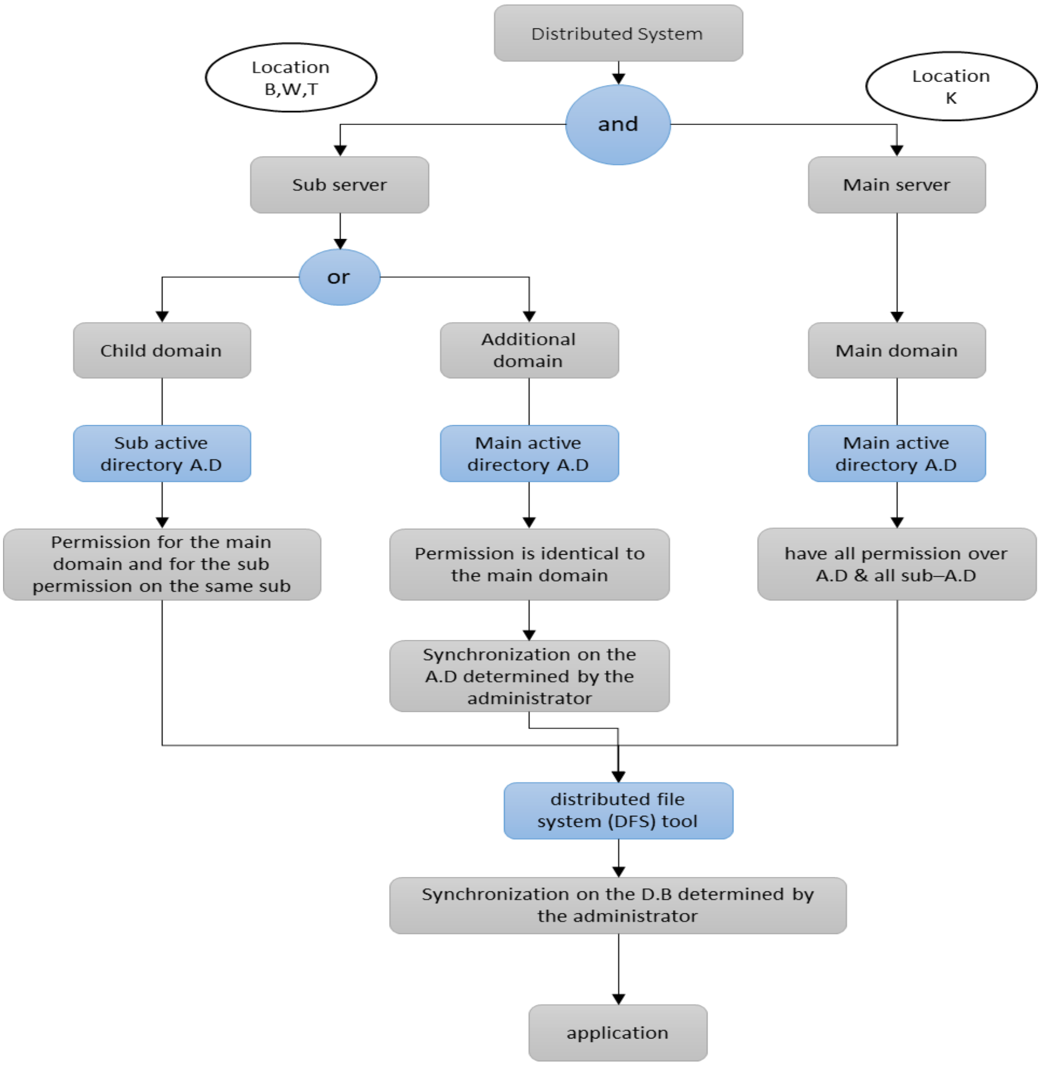

Distributed System
Location B,W,T
and
Location K
Sub server
Main server
or
Child domain
Additional domain
Main domain
Sub active directory A.D
Main active directory A.D
Main active directory A.D
Permission for the main domain and for the sub permission on the same sub
Permission is identical to the main domain
have all permission over A.D & all sub–A.D
Synchronization on the A.D determined by the administrator
distributed file system (DFS) tool
Synchronization on the D.B determined by the administrator
application

Figure 6 behavior of the main domain, Additional, and Child

IV. RESULTS AND CONCLUSION

The system was virtually implemented using GNS3, the locations were given fixed IP addresses. Each location is connected to the network using a microwave link. All locations included a router and a firewall in addition to a server as shown in Figure (7). All servers include a Replicated Active Directory, a Domain Controller, and Sub-Doman, an SQL Server with SQL Snapshot and Transactional replications.

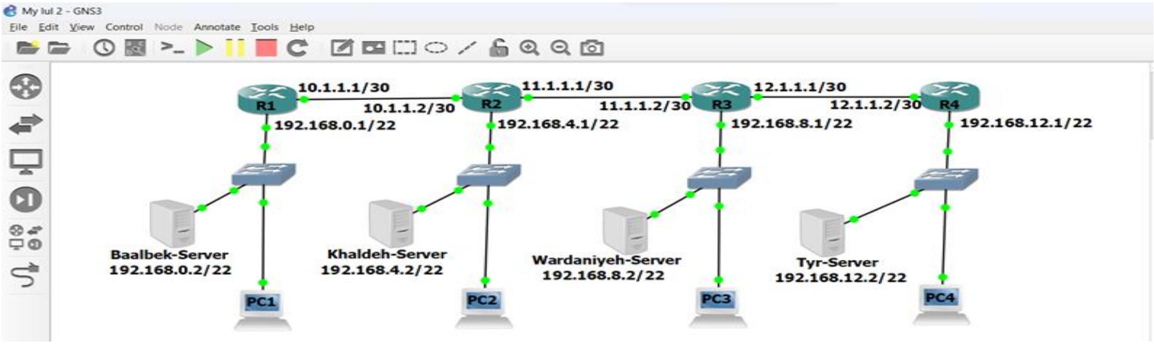
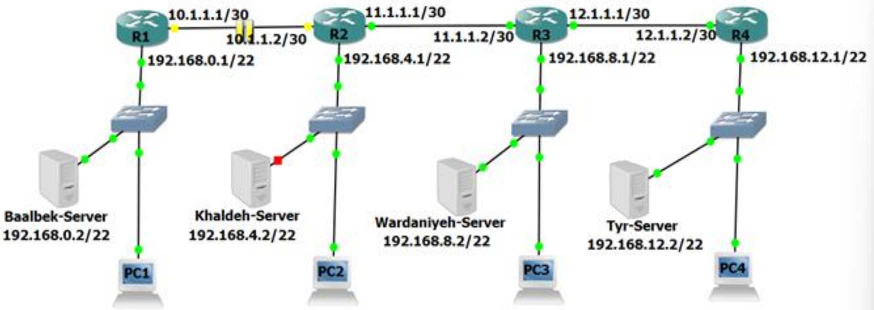
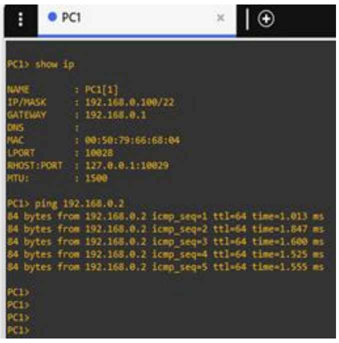


Figure 7 Virtual implementation for a distributed system using GNS3

Disconnecting Location B from the network (the connection between K and B is no more available) showed that all other locations were informed about the disruption that happened at B, and they continue working as normal. All AD, DC, and SQL continue to work normally serving their clients locally and remotely without disruption or lowering the quality of service even after completely shutting location B down as shown in Figure (8).





```
PC1> show ip

NAME        : PC1[1]
IP/MASK     : 192.168.0.100/22
GATEWAY     : 192.168.0.1
DNS         :
MAC         : 00:50:79:66:68:04
LPORT       : 10028
RHOST:PORT  : 127.0.0.1:10029
MTU:        : 1500

PC1> ping 192.168.0.2
84 bytes from 192.168.0.2 icmp_seq=1 ttl=64 time=1.013 ms
84 bytes from 192.168.0.2 icmp_seq=2 ttl=64 time=1.847 ms
84 bytes from 192.168.0.2 icmp_seq=3 ttl=64 time=1.600 ms
84 bytes from 192.168.0.2 icmp_seq=4 ttl=64 time=1.525 ms
84 bytes from 192.168.0.2 icmp_seq=5 ttl=64 time=1.555 ms

PC1>
PC1>
PC1>
PC1>
```

Figure 8 Disconnecting the connection between the (B) branch and the (K) branches and shut down the main server

The distributed system showed great performance, scalability, and security.

All locations have their servers, the servers contain AD, DNS, and DHCP. Users logging in to a specific server will be served from the same location unlike the central system, which forces users to retrieve information from the central branch. The system was configured as follows:

1. Synchronization is at the AD level, and the synchronization policy is set at 3:00 AM.

2. Replication at the file level via DFS Replication determine the replication policy at 12:00 am

3. Replication at the database level and defining the replication policy at 2:00 am and for two types (Snapshot and Transactional)

To compare efficiency and speed between the centralized and the distributed systems. We put into test two scenarios

1- A user at B is requesting a service from a centralized system at K.
2- A user at B requesting a service from a distributed system.

The user at B requests a service, (PDF report or Excel file), Suppose that the size of the file is 1MB, and the average data transfer speed is 25 Mbps (between B and K). Supposed location B has ten active users, the bandwidth will be divided by 10 (25Mbps/10 users) or 2.5Mbps available for each user. Requesting a 1Mbps file will take a minimum time of 0.4 seconds to reach user at location B. Sending a file of 50Mbps will take a minimum time of 20 seconds as shown in table (1).

Using the distributed system, a user at location B is no longer required to as a request from the main location K. The user will be served locally. Suppose that the data transfer rate at B network is 100Mbps using (Twisted cable CAT5 and NanoBeam M5). The user requests a file of 1Mbin size, suppose we have 10 active users onsite, the available bandwidth will be (100Mbps/10 users) or 10Mbps for each user. Requesting a 1Mbps file will be served in 0.1 seconds and the one with 50Mbps will be served in 5 seconds as shown in Table (1).

Table 1 A comparison between the two systems in the data transfer speed

Data Size System	**1Mb**	**20Mb**	**50Mb**	**100Mb**
Central System	0.4s	8s	20s	40s
Distributed System	0.1s	2s	5s	10s

Comparing the response times for both centralized and distributed systems showed a great advancement in functionality, availability, and quality of service when using the distributed systems as shown in Figure (9).

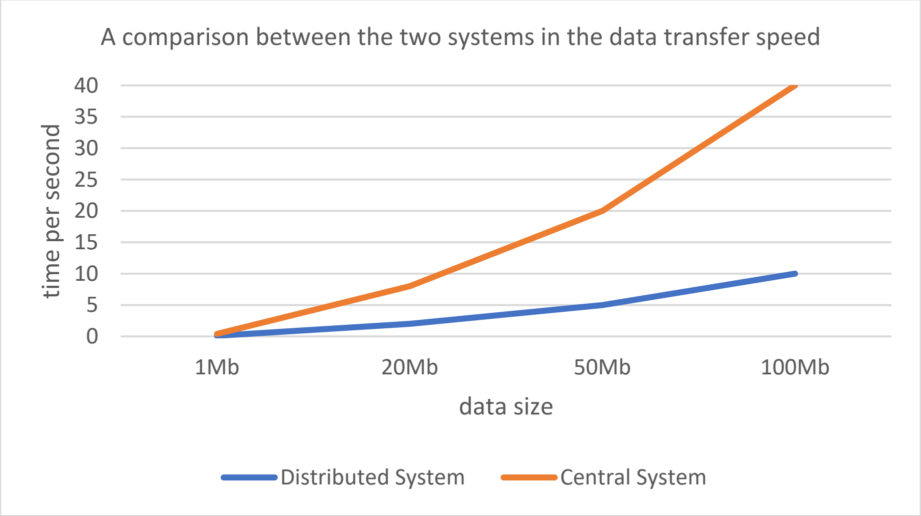


Figure 9 A comparison between the two systems in the data transfer speed

V. REFERENCES

[1] Stankovic, John A. "Research directions for the internet of things." *IEEE internet of things journal* 1.1 (2014): 3-9.

[2] Kopetz, Hermann, and Wilfried Steiner. Real-time systems: design principles for distributed embedded applications. Springer Nature, 2022.

[3] Benelallam, Amine. Model transformation on distributed platforms: decentralized persistence and distributed processing. Diss. Ecole des Mines de Nantes, 2016.

[4] Verissimo, Paulo, and Luis Rodrigues. Distributed systems for system architects. Vol. 1. Springer Science & Business Media, 2001.

[5] Okardi, Biobele, and O. Asagba. "Overview of distributed database system." *International Journal of Computer Techniques* 8.1 (2021): 83-100.

[6] Binduf, Afnan, et al. "Active directory and related aspects of security." 2018 21st Saudi Computer Society National Computer Conference (NCC). IEEE, 2018.

[7] Mazilu, Marius Cristian. "Database replication." Database Systems Journal 1.2 (2010): 33-38.

[8] Larson, P-A., Jonathan Goldstein, and Jingren Zhou. "MTCache: Transparent mid-tier database caching in SQL server." Proceedings. 20th International Conference on Data Engineering. IEEE, 2004.

[9] Thomson, Alexander, and Daniel J. Abadi. "{CalvinFS}: Consistent {WAN} Replication and Scalable Metadata Management for Distributed File Systems." 13th USENIX Conference on File and Storage Technologies (FAST 15). 2015.

[10] Buyya, Rajkumar, Rodrigo N. Calheiros, and Amir Vahid Dastjerdi, eds. Big data: principles and paradigms. Morgan Kaufmann, 2016.

[11] Hrubý, Tomáš. "Active Directory and Kentico CMS Network Synchronization."

[12] Rashid, Nazre Abdul, et al. "Cisco packet tracer simulation as effective pedagogy in Computer Networking course." (2019): 4-18.

[13] Rosenblum, Mendel, and Tal Garfinkel. "Virtual machine monitors: Current technology and future trends." Computer 38.5 (2005): 39-47.

[14] Minasi, Mark, et al. Mastering Windows Server 2012 R2. John Wiley & Sons, 2013.

[15] Borthakur, Dhruba. "The hadoop distributed file system: Architecture and design." Hadoop Project Website 11.2007 (2007): 21.

[16] Berkouwer, Sander. Active Directory Administration Cookbook: Actionable, proven solutions to identity management and authentication on servers and in the cloud. Packt Publishing Ltd, 2019.