Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 7.056

IJCSMC, Vol. 10, Issue. 9, September 2021, pg.38 – 45

Distributed Database System Optimization for Improved Service Delivery in Mobile and Cloud BigData Applications

O.T Jinadu[1]; O.V. Johnson[2]; M. Ganiyu[3]

[1]Department of Computer Science Rufus Giwa Polytechnic Owo, Nigeria
[2]School of Computer Sciences Universiti Sains Malaysia, Malaysia
[3]Department of Computer Science Federal Polytechnic Ile-Oluji, Nigeria
[1] yimikajinad@gmail.com; [2] olajohnson@fedpolel.edu.ng; [3] mutganiyu@fedpolel.edu.ng

DOI: 10.47760/ijcsmc.2021.v10i09.004

Abstract— Many issues associated with managing centralized database include data isolation, redundancy, inconsistency, and atomicity of updates, among others; however, distributed database implementation over high-performance compute nodes maximizes information value across the networks. Also, analysis of bigdata generated/consumed over the mobile Internet, Internet of Things (IoT) and cloud computations necessitates low-latency reads and updates over cloud clusters. Conventionally, services in distributed systems demand optimized transactions. This paper examines transaction generation over distributed storage pool using suggested reference architectures of fragmentation using hybrid semi-join operations to offer mobility transparency as an additional ingredient of integrity transparency offer of DDBMS. Distributed storage pool is simulated using configured WLAN to activate multiple file transfers concurrently, engaging mobile nodes and large file sizes. Major functionality desired in the storage pool is improvised by storage virtualization whereby a global schema query optimizer effects transaction management to characterized latency-driven throughputs achieved by joint optimization of network and storage virtualization. Measurements and evaluations gave the best overall performance of low-latency reads and updates using the provisioned mobile-transmission control protocol (M-TCP). Appreciable improvement in service delivery is offered using distributed storage pool (DSP) facilitated with hybridized RAID construction and copy mechanisms. Improved response-time and speed-up transmissions evidently showed low-latency read and update transactions, depicting improved service delivery. Evaluating the DDBMS model simulated in the DSP architecture, all complexity (overheads) associated with conventional shared systems were minimized.

Keywords— RAID, Replication, Distributed Database, Transaction, Remote-join, Virtualization

I. INTRODUCTION

A cloud computing platform facilitates distributed and parallel software systems. Deployed either as Software as a Service (SaaS) or Security as a service (SecaaS), it became expedient for cloud services (typically bigdata and mobile computations) to be offered with latency-driven throughput, rally around distributed storage pools [1], intelligently provisioned by virtualization technology. These coherent designs are a great catalyst in the economies of scale requirements of cloud computing (servers, storage) and cloud service (on-demand access) models. To provision backup and disaster recovery, several technologies including, cloning, snapshots, mirroring among others are implemented to extend Redundant Array of Independent Disks (RAID) technology for greater delivery [2]. RAID construction converts multiple physical storage (hard disks) infrastructures to one

logical volume and implements logical partitioning and network file system structure to provide for hot-swapping during disasters.

Storage pools as virtualized cloud resources enable distributed processing, fast deployment and rapid provisioning of data with reduced latency using combined technologies of virtualization and hybridized raid construction. To support mobile and bigdata applications, ubiquitous access to data stored in DDBS architecture is strongly facilitated with low-latency reads and updates to feature as Distributed Resource Pool (DRP). Virtualization promotes dynamic resource mobilization (DRM) to offer high availability (HA) and distinct power management (DPM) requirements of real-time cloud agents. DRPs, activated with storage virtualization technology are capable of meeting suggestive demand of high data availability with reduced-latency stemming from the extreme speeds of generated huge volumes of unstructured data in bigdata and mobile computing. Heterogeneous feature of Internet data also motivates desired features of bigdata application to include, robustness, high availability, fault-tolerance, low-latency reads and updates among others.

In contrast to centralised systems, with a single CPU, distributed systems collect independent computers to implements distributed database access. A primary goal of Database Management System (DBMS) centers on the provision of environment, convenient and efficient in information storage and retrieval [3]. Major advances in information and communication technologies (ICT) trends sufficed, implementing distributed computing systems (DCS), which had called for 'embarrassingly-parallel' algorithm used in high-performance computing (HPC) architectures shown in Fig. 1. Capable of interconnecting many small/medium/large business systems using intelligent infrastructures, DCS enables resource sharing. Resources, including compute, storage, network, database, security, Enterprise Intelligence and Databases are made available via rapidly configurable shared pools in various cloud computing models. DDBMS plays a significant role in bigdata analysis using cloud software stack in SaaS cloud computing models [4]. As a complex piece of software, DBMS enables database creation and implementation with high adaptations, supporting location transparency for usage efficiency. Though DBMS provides data currency and consistency, offering integrity and enforces security, it has numerous issues. Featured as centralized technology, it collects information into a store but facilitates distribution/sharing via communication networks [5]. In addition, DBMS isolate programs from data format and as data changes programs need not change, a special feature attributed to flexible architecture of one-to-one, one-to-many or many-to-many logical relationships design structure of data items.

Unlike DBMS, which manages information as a single-site database, Distributed DBMS (DDBMS) manages several databases physically distributed over several storage sites and the architecture enables management of synchronized data just-in-time and/or periodically as if it were stored on the same computing node [6]. DDBMS model, depicted as Distributed Resource Pool (DRP) in this study is motivated to enhance the best overall performance of scalability by (i) offering storage at low costs, using storage virtualization technology and (ii) implementing as multi-backups within the architecture.

In mobile computing, as data and information location dynamically change, leading to increased volatility, and appreciable concurrency problems among multiple users, who may need to access the same data [7]. As a vast amount of information generated needs to be stored, limited storage on mobile hosts is a major factor that necessitates the provisioning of enhanced technology. A fragmented multi-database system, such as DDBMS, which supports replication is not sufficient to meet this challenge of rapidly increasing demands of data with low-latency reads and updates.

Therefore, this research suggests the implementation of virtualized storage construction using Redundant Array of Independent Disks (RAID) technology and associated copy mechanism to activate a distributed storage pool that meets the low-latency data reads of mobile (bigdata) computing. Integrating virtualized storage with the embedded technologies will offer replication to deliver required robustness and fault-tolerance characteristics offer of heterogeneous Local Schema (LS) of Multi-Database design framework. Each slave supports location transparency exhibited at data nodes while metadata, which is an aggregation of metadata, characterised with a Global Schema (GS) structure. Integrating all LS, while exhibiting data fragmentation and transparency [8], Restricted Global Schema (RGS) is offered as a union of all LS exhibiting data replication using semi-join operations.

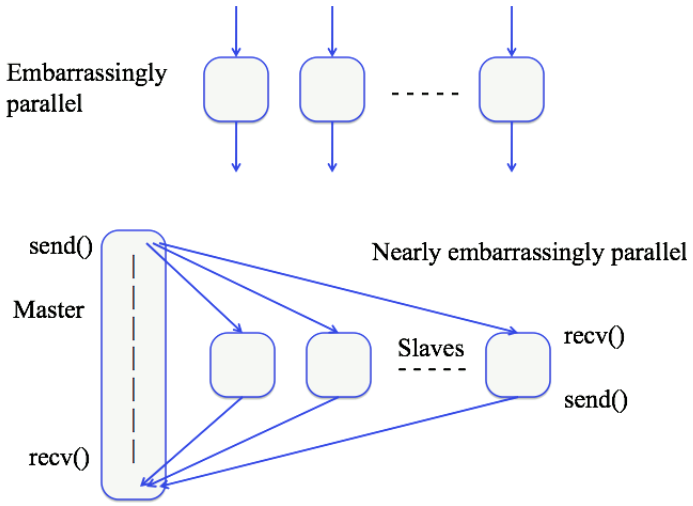

Fig. 1. Parallel algorithms for HPC based on DCS (Source: [9])

II. DBMS ISSUES AND BIGDATA REQUIREMENTS

The database is viewed as a collection of logical data items [5], and arbitrary collection of data fields, [10] further described a distributed database (DDB) as a database of DCS and a special database, with storage location not attached to any computing node (physical compute node) but rather stored on multiple computing (storage) nodes within a cluster or across several zones [11]. As suggested in [12], databases can be categorised as centralised, NoSQL, relational, object-oriented, cloud, network, hierarchical, or distributed. Implementation is dependent on users' needs and application requirements. With the current trend in mobile and cloud computing, other categories of databases are made functional in a distributed approach considering the nature and desired features of bigdata analytics.

Though these database systems are capable of storing large bodies of information and providing for their safety (data security), they are characterized by the problems of :- data redundancy and inconsistency; difficulty in accessing data through programming; data isolation and one-format structure; integrity and atomicity of updates; and insecurity due to no standardised protocols. Sometimes, the increased complexity of database design under DBMS makes performance evaluation difficult as a process running at one node may impact the entire network. Therefore, with the emergence of DDBMS technology, with its several features of improved shareability, availability, reliability, and performance, data is located nearest the greatest site and is dispersed to match business requirements. With reduced operation costs ensured, even as new sites can be added without much compromise, the problems of integrity control and management complexities arising from security lapses remain an axe to grind. Furthermore, increased storage, duplicate infrastructural requirements and access path selection are all issues that must be addressed.

A. Requirements of Generalized DDBMS

Conceptually, centralized DDBMS manages several databases physically distributed to several sites, as depicted in Fig. 2. With the current trends in cloud and big-data requirements, the distributed database environment consists of multiple databases located at loosely coupled sites that share no physical components but belong to the same logical system [13]. Through an integrated synchronization technique offered as multiple users access the same data while the system ensures that updates, deletes, additions etc performed on such data at one location automatically reflects in the data stored elsewhere. This suggests mobility transparency characterized by implementing several servers in distributed systems. Each of the sites could communicate via wired or wired or wireless network. Each site features as a transaction and storage site where queries can be issued and replicated data stored respectively. As the DBMS operated at the various sites are autonomous, exhibiting some form of heterogeneity, they are referred to as multi-database systems.

B. The architecture of Distributed Systems

Although, from a user's perspective, a distributed database is logically a single database even if it is physically distributed and it consists of a possibly empty set of query sites (sites *1 to n*) and a non-interface and the applications, to facilitate data access but the query sites may not. There are many architectural modes for

developing distributed DBMSs. The most significant of these are client/server systems, *where the query sites correspond to clients while the data sites correspond to the server. Here, the server does most of the data management works-the query processing and optimizations, transaction and storage management; a* Peer-to-Peer System, *where no distinction is made among client machines and server machines, and all clients must have DBMS client module installed, for managing 'cached' data. This is in addition to the application and user interface.*

Distributed DBMS offered as a software system, permits the management of distributed databases storages, though the distributions appear transparent to users [11].

C. Technical Issues of Distributed Database System

Generally, the American National Standards Institution-Standard Planning and Requirements Committee (ANSI-SPARC) provides a standard basis for database architecture to consist of three levels of abstraction [8]. These include an external level of abstraction, *which models the different user and their global views, to incorporate all sets of data and their different representations in the database; c*onceptual level of abstraction, *which describes what data, is stored in the database and their entities, including the attributes, relationships, security and integrity; and i*nternal level of abstraction, *which describes how the data is stored in the database.*

The suggested architecture is a strategy to integrate the functions of all cluster components within any distributed system. As the need for database stability is extremely important, users need to access reliable data at all times. This in essence, as expressed by [12], is the ultimate goal of implementing a DDBMS model as DRP architecture. Copy mechanism (COW) enables data replication for accurate backup onto the master and slaves through replication, using the DRP architecture.

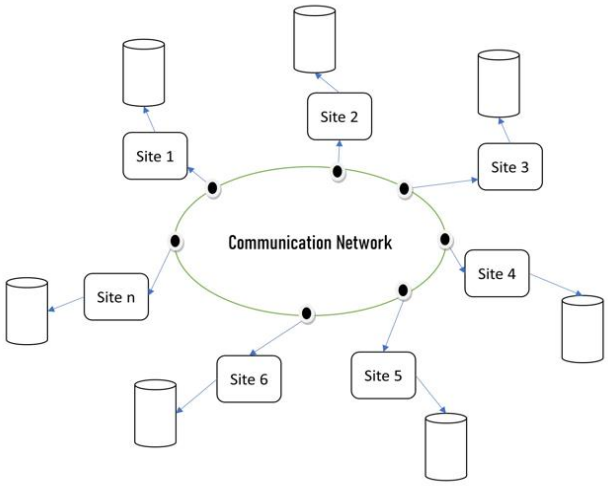

Fig 2. Distributed Database Environment (Adapted from [5])

Distributed database feature as Homogenous database, where every site runs same DBMS module and database structure constructed with same type relations. All sites have identical software and each site surrenders part of its autonomy (in terms of access right to change schemas or software) and all sites are aware of each other but agree to cooperate in processing user requests, making whole systems appear to users as a single system. This is the concept of location transparency or Heterogeneous database, where different sites run different DBMSs, supplied by various vendors and such databases are called multi-database. Different sites use different schemas (and software) where issues are resolved by query processing and optimization, as all sites are not aware of each other. A heterogeneous database system provides limited facilities for cooperation in transaction processing. The latter suggests many incompatibilities and increased overheads.

D. Storage Pool Construction using RAID Technology

With hybridized RAID constructed using RAID 3 and RAID 5, a storage pool is obtained with all chunks of all mounted physical storage servers are migrated at the level of virtualization. With RAID 3, (2D + 1P) and RAID 5 (3D + autogenerated and distributed P), where D represents the number of hard disks (HD) and P represents the parity.

RAID 5 is supplemented with RAID 3 because no extra overhead will be incurred on mounting extra hard disks to do parity but all represented HDs store the parity. This technology is characterised by no write penalty but better utilization due to distributed parity. Protection is provided with a logical unit number (LUN) created on the storage pool via the two RAIDs (3 and 5) and hot-spares (HS) created in LUN virtualization.

The two HSs depict replication and mirroring, cloning, snapshot creation and rollovers among other technologies that are easily facilitated for hot-swapping in any event of disasters. Read-write (R/W) performances increase as well and low-latency access of data is enhanced with protection.

E. Bigdata Requirements

Desired properties of a Bigdata system include robustness, fault-tolerance, low-latency-reads and updates, scalability, generalization, extensibility and minimal maintenance [14]. In addition, bigdata systems must support Debuggability, which is accomplished in suggested architecture and any batch layer is preferred to effect re-computation algorithms if desired. A Big Data system is capable of debugging the system if things go wrong. Fundamental assumptions for bigdata users are the ability to simultaneously/concurrently update replicated databases. DDBMS architecture therefore offers the provision of:

- optimized distributed transaction processing, to minimized query processing costs on each slave node;
- reduced communication to enhanced data integrity at different levels; and
- transportable/mobile communicable protocols, standardized to maximize asynchronous and concurrent processing techniques.

Technically, DBMS maintains a relationship between logical data items and provides integrity of the data in terms of data consistency (data correctly reflecting the state of the system); currency (a database containing recent information) and concurrency (data accessible by different users simultaneously. Despite these features, DDBMS remains inefficient in managing all databases, even as new sites are added to existing systems, especially as recorded in *teletraffic* of mobile computing.

III. SYSTEM MODELING FOR TRANSACTIONS

Traffic modeling (over unified networks) connecting distributed databases is considered with the duration of File Transfer sessions incurred for 'transmit and idle' periods [15]. FTP simulations consider three identified sessions to include:

- U representing the number of FTP sessions over network-A, with a connection at the wireline network.
- U1 of these sessions are wireline sessions, both ends of the connection are wireline.
- U2 are wireless sessions with one end of the connection at the wireless side of the network, and the other end at the wireline side. Mathematically,

$$U2 = U - U1 \quad \text{(i)}$$

The traffic associated with U2 sessions encounters two transport protocols consisting of (a) transport protocol suggested for WLAN and (b) standard TCP-Reno protocol used in wired network-A. For a packet size of 1KB, approximating 8 TCP payloads (1TCP = 128Bytes), 2 TCP payload transmittable in 1 cycle. Two consecutive sessions equal a cycle [16].

The input traffic for FTP sessions for wireline traffic is reported in Table 1 while the wireless FTP sessions for second transport (network-A) are given in Table 2. The idle period (T2 or t2) after the transmission is assumed to be substantially larger than the fade duration of the channel [17], [18], [19]. Idle time is the intertransmit time for both channels. The fading time was small otherwise the mobile device closes the connection). For the wireline sessions, the average FTP transmission time is T1 representing the ON/OFF model for FTP sessions, T1 and T2 are exponentially distributed. Four cycles (I – IV) representing the 8 TCP payload of 1KB packet size was considered from the simulated 32 packets window size.

TABLE 1
WIRELINE TRAFFIC MEASUREMENTS FOR 1KB PACKET (APPROX. 8TCP PAYLOADS)

TCP Cycle	Transmitting *T1*(sec)	Idle *T2* (sec)	*T=T1+T2* (sec)
I	13.0	4.6	17.6
II	12.5	3.0	15.0
III	12.5	1.5	14.5
IV	12.0	1.0	13.0
TOTAL	50.0	10.1	60.1

TABLE 2
WIRELESS TRAFFIC MEASUREMENTS FOR 1KB PACKET (APPROX. 8TCP PAYLOADS)

TCP Cycle	Transmitting *t1*(sec)	Idle *t2* (sec)	*T=t1+t2* (sec)
I	2.4	1.0	3.1
II	2.2	1.2	3.2
III	2.3	1.1	3.5
IV	2.0	1.0	3.2
TOTAL	8.9	4.1	13.0

For U =8, U1=4 and U2=4 with BER set to 10-4, the throughput of network-A using the simulation is 384bps; giving 3.84% utilization. Wireline FTP traffic between S-1 and the WLAN follows a similar distribution. For the wireless FTP traffic leaving the WLAN, less link delay is suffered in transferring packets without errors due to RF link to enter network-A confirming given assertions.

Though the presence of white noise follows Gaussian distribution, and as the number of FTP sessions increases, improved TCP connection recorded 778bps in the simulation; suggesting 7.78% utilization of network-A (doubling with M-TCP) [20].

With t1 = average transmission time for wireless FTP sessions leaving the WLAN and entering network-A, when there is no fading over the wireless channel due to the filtering process of I-TCP over the WLAN, T1=t1. Let t2 = average fade duration for the wireless traffic transfer the WLAN; t1 and t2 are exponentially distributed random variables according to Jakes Model [21], [22], [23] and T2>> t2, denoting an improved throughput of (> 22%) for the unified network.

With t3 representing modified idle (inter-transmit) time for the wireless traffic over network-A, and N a random variable representing the number of inter-fade periods during T. Then,

$$N = \frac{T1}{T2} \tag{ii}$$

and

$$T1 + T2 = (N+1) \times t1 + Nt2 + t3 \tag{iii}$$

Therefore,

$$t3 = (T1 + T2) - (N+1) \times t1 + Nt2 \tag{iv}$$

Network Simulator version 2 (*Ns*-2) [24] was used for the simulation. Significant code modifications made to the original simulation script were done to fit the purpose of this study. Running test assumptions and parameters include:

- a WLAN configuration based on Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) technique;
- WLAN concurrent sessions established using TCP-Reno with LAN bandwidth of 10 Mbps;
- WLAN Bandwidth of 10Mbps with a link delay of 10 microseconds; and
- packet size of 128 Bytes as TCP's payload for window size of 32 packets with required design parameters set to: T1 = 50 sec; T2 = 10 sec; t1 = 9 sec; t2 = 4 sec. with an average bit error rate (BER) of 10-1 used.

IV. NUMERICAL ANALYSIS OF RESULTS

The following cases were considered for the proposed architecture: (i) without M-TCP

- Case 1: Taking U=8, U1=4, U2=4 and the bit error rate (BER) for the wireless session set to 10-4. Throughput of Network-A using the simulation is 384Kbps; approximately 3.95% utilization for network-A for TCP/IP
- Case 2: U=4, U1=2, U2=2 and BER is changed 10-5. Throughput of network-A increased to 778Kbps; approximately 7.9% utilization of network-A.
- Case 3: pure wireline TCP with BER = 0 for U1=4, throughput increased to 2460Kbps; approximately 24.6% utilization arising from 790 x 4 TCP-Reno payloads with 128Bytes representing 1 TCP payload.
- (ii) with M-TCP
- Case 4: Taking U=4, U1=2, U2=2 and BER=10-4 throughput recorded is 2480Kbps, indicating a significant improvement of M-TCP over TCP-Reno; approximately 24.8% utilization of network-A
- Case 5: With more wireless FTP sessions, U2=4, U=8, throughput dropped due to randomness of wireless traffic.

V. CONCLUSION

This study examines a maximum achievable throughput for unified network transmission between distributed database system nodes engaged in multiple access transactions. Using the electronic Network Simulator Package (eNSP) and specified configurations to simulation a WLAN, equipped with various storage server nodes, that enables multiple FTP sessions, detectable packet loss was recorded. This was due to fading of wireless network segments. The loss was attributed to the hidden problem from the wireline side, envisaged to have been introduced by implemented protocol, the mobile Transmission Control Protocol (M-TCP).

A measurement of 220% Throughput was realized for the wireline than in the wireless (WLAN) configuration setup, which gave only 180%. It was also discovered that as the number of wireless FTP sessions increases, the overall throughput of network-A decreases without M-TCP. Network-A depicts the presence of multiple storage server nodes, that enable users to run various FTP sessions. Under these simulation specifications, maximum achievable throughput became appreciably higher in WLAN setup using the specified M-TCP in agreement with [25].

Using hybridized RAID technology to implement the DSP, enabled with abstracted copy-on-write (COW) mechanisms, improved services, characterised with high availability, fault-tolerant computing, reliable on-demand data services requirement of mobile and bigdata cloud applications was achieved. Improved response time and speed-up transmissions evidently confirmed the required low-latency read and update transactions. Service delivery evaluation using the DDBMS model, simulated in the DSP architecture minimizes the complexity (overheads) associated with conventional shared systems.

M-TCP was implemented on an integrated network with envisaged concurrent transactions on multi-database, featured as heterogeneous database sites. Required data transfer was effectively transmitted among multiple users, with the offer of improved throughput. M-TCP accounted for maximum achievable throughput performance than with standard TCP. Using M-TCP, WLAN configuration depicted a mobile-computing scenario while the multiple FTP sessions simulated the use of multiple-slave storage nodes as replicated, distributed database nodes.

REFERENCES

[1] Pati, S. P., and Pattnaik, P. K., "Criterions for Databases in Cloud Computing Environment. Intelligent Computing, Communication and Devices", 385–392, 2014, doi:10.1007/978-81-322-2012-1_4.

[2] Huawei Technologies Inc., Cloud computing training material (PDF), 2020.

[3] S. Tarun, R.S. Batth and S. Kaur, "A Review on Fragmentation, Allocation and Replication in Distributed Database Systems", 2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE), IEEE Xplore, 2019.

[4] M.K Gupta, R.K Arora and B.S. Bhati, "Study of Concurrency Control Techniques in Distributed DBMS", International Journal of Machine Learning and Networked Collaborative Engineering, 2(04) pp 180-187, 2018, https://doi.org/10.30991/IJMLNCE.2018v02i04.005.

[5] A. H. Al-Sanhani, A. Hamdan, A. B. Al-Thaher and A. Al-Dahoud, "A comparative analysis of data fragmentation in distributed database", 2017 8th International Conference on Information Technology (ICIT), 2017, pp. 724-729, doi: 10.1109/ICITECH.2017.8079934.

[6] C. George, D. Jean and K. Tim, Distributed Systems: Concepts and Design, 3rd ed. Harlow: Addison-Wesley, 2001. xiii, 772. ISBN 0-201-61918-0, 2001.

[7] T. Connolly and C. Begg, *Database System: A Practical Approach to Design, Implementation & Management*, Sixth Edition ISBN-13978-0132943260 Publisher Pearson, 2015.

[8] M.T. Özsu and P. Valduriez, *Principles of Distributed Database Systems*, doi:10.1007/978-3-030-26253-2, 2020.

[9] Nearly Embarrassingly parallel algorithm in High-Performance Computing, retrieved on 3rd September, 2021. https://www.cs.iusb.edu/~danav/teach/b424/b424_23_embpar.html.

[10] A. M. Lourdes, *Learn based Optimisation of Distributed Queries. Databases* [cs.DB], Université Grenoble Alpes, 2014. English. ⟨tel-01586508⟩.

[11] O. Biobele and A.O. Prince, "Overview of Distributed Database System", International Journal of Computer Techniques, Vol. 8(1): 83-100, ISSN: 2394-2231, 2021.

[12] P.O. Orlunwo and O.A Prince, "Distributed Database Management System (DBMS) Architectures and Distributed Data Independence. IJCSMC, 10(1), 23–48, 2021, doi.org/10.47760/ijcsmc.2021.v10i01.004.

[13] A.S. Tanenbaum, *Distributed Operating Systems (Computers)*, Pearson Education 606 pages, 1995.

[14] N. Marz, *Database: Desired Properties of Bigdata Systems*. Big Data. First Edition. Manning Publications.

[15] A. Bruno and O. Katia, "A Framework for Modelling Spatial Node Density in Way-point based Mobility", Issues of Wireless networks, Springer Links. Vol 21 Issue 4, 2012

[16] C.F. Chiasserini and M. Meo, "Improving TCP over Wireless Through Adaptive Link Layer Setting" IEEE GLOBECOMM Symposium on Internet Performance, 2001.
[17] J. Geetha and G. Gopinath, "Ad-Hoc Wireless Routing Protocols – A review", Journal of Computer Science. Science Publications, India, 2007.
[18] H Lee, S. Lee and D. Cho, "Performance Analysis of Wireless Multiuser VoIP System with Adaptive Modulation and Coding" Issues of Wireless networks, Springer Links. Vol 21 Issue 4, 2012.
[19] K. Radha and H. Martin, "Dynamic Connectivity and Path Formation Time in Poisson Networks", Issues of Wireless networks, Springer Links. Vol 22 Issue 4, 2013.
[20] J.M. Wozencraft and I.M Jacobs, *Principles of Communication Engineering*, Wiley, New York, 2006.
[21] W. Jakes W, *Microwave Mobile Communications*, IEEE Press, 1993.
[22] H.T Chaskar T. Lakshman and U. Madhow, "The Design of Interfaces For TCP/IP Over Wireless", In Proc. IEEE Milcom. 2006.
[23] J. Yoon, J. Choi, K. Kee and S. Yang, "A Fully Distributed Replica Allocation Scheme for an Opportunist Network", Issues of Wireless networks, Springer Links. Vol 20 Issue 4, 2011.
[24] T. Issariyaku and E. Hossain, "Introduction to Network Simulator 2 (NS2). Introduction to Network Simulator NS2", 21–40, 2011. doi:10.1007/978-1-4614-1406-3_2.
[25] E. Quaddoura and A. Daraiseh, "TCP Optimal Performance in Wireless Network Applications", International Journal of Computer Science, Science Publications, Saudi Arabia, 2006.

