


Design of a DFS to Manage Big Data in Distance Education Environments


Mahmut Ünver*

(Kırıkkale University, Department of Computer Technologies, Kırıkkale, Turkey
 <https://orcid.org/0000-0002-5882-2897>, munver@kku.edu.tr)

Atilla Ergüzen

(Kırıkkale University, Faculty of Engineering, Department of Computer Engineering,
Kırıkkale, Turkey
 <https://orcid.org/0000-0003-4562-2578>, atilla@kku.edu.tr)

Erdal Erdal

(Kırıkkale University, Faculty of Engineering, Department of Computer Engineering,
Kırıkkale, Turkey
 <https://orcid.org/0000-0003-1174-1974>, erdalerdal@kku.edu.tr)

*Author for correspondence

Abstract: Information technologies have invaded every aspect of our lives. Distance education was also affected by this phase and became an accepted model of education. The evolution of education into a digital platform has also brought unexpected problems, such as the increase in internet usage, the need for new software and devices that can connect to the Internet. Perhaps the most important of these problems is the management of the large amounts of data generated when all training activities are conducted remotely. Over the past decade, studies have provided important information about the quality of training and the benefits of distance learning. However, Big Data in distance education has been studied only to a limited extent, and to date no clear single solution has been found. In this study, a Distributed File Systems (DFS) is proposed and implemented to manage big data in distance education. The implemented ecosystem mainly contains the elements Dynamic Link Library (DLL), Windows Service Routines and distributed data nodes. DLL codes are required to connect Learning Management System (LMS) with the developed system. 67.72% of the files in the distance education system have small file size (≤ 16 MB) and 53.10% of the files are smaller than 1 MB. Therefore, a dedicated Big Data management platform was needed to manage and archive small file sizes. The proposed system was designed with a dynamic block structure to address this shortcoming. A serverless architecture has been chosen and implemented to make the platform more robust. Moreover, the developed platform also has compression and encryption features. According to system statistics, each written file was read 8.47 times, and for video archive files, this value was 20.95. In this way, a framework was developed in the Write Once Read Many architecture. A comprehensive performance analysis study was conducted using the operating system, NoSQL, RDBMS and Hadoop. Thus, for file sizes 1 MB and 50 MB, the developed system achieves a response time of 0.95 ms and 22.35 ms, respectively, while Hadoop, a popular DFS, has 4.01 ms and 47.88 ms, respectively.

Keywords: Distance Education, Big Data, Distributed File System, Dynamic Block Size, Serverless Architecture

Categories: C.1.4, C.2.4, C.4

DOI: 10.3897/jucs.69069

1 Introduction

In recent years, with the rapid development of the topic of Big Data and Internet technologies, DFS, data storage and management systems have also become the subject of important research [Chervyakov et al. 2019, Peng et al. 2020]. DFS systems have now become commercially attractive due to their ability to store large amounts of data compared to traditional data storage and management systems used in the past [Liao et al. 2015]. When data exceeds the size that one computer can manage, it is inevitably moved to other computers, and this process creates the need to manage more than one computer at a time, which is called DFS. In other words, DFSs are robust storage systems that allow files to be physically stored in different locations on the network, but users can access and edit their files as if they were stored on the local machine. The design and implementation of a DFS are more complicated than a traditional file system because of the physical distribution of users and storage devices. In DFS, the file system provides file services to clients. Clients can perform certain operations (create, delete, read, write, etc.) through client interfaces. These files are stored on the server's disk. A DFS can act as an intermediate layer or be part of the file system of a classical operating system environment [Ergün et al. 2013]. DFS is the only solution method for modern data management systems such as Hadoop, NoSQL and relational database management systems (RDBMS).

DFSs create numerous duplicates of the similar data block, which are stored as backups, and can store these copies on another computer in a network environment [Kaseb et al. 2019]. DFSs are used for data storage in many fields, such as commercial enterprises, banks, military and defense, healthcare, and educational systems. A 2018 study proposed a solution to the Big Data problem in healthcare, but the system created is not very efficient at storing small files due to its constant block size and bitmap structure [Ergüzen and Ünver 2018]. Systems that use block structures with a size of several megabytes are not very suitable for large data sets with small files due to internal fragmentation. Therefore, our study does not use a bitmap scheme, which negatively affects the performance and robustness of the data, and also uses a dynamic block structure that prevents internal fragmentation. Moreover, compression and encryption features are included to the platform.

Studies related to DFS first began in the [Alsberg and Day 1976]. The basic operational approach of DFS is to split big data into clusters and execute or store them on different devices. DFS systems are very secure in terms of data storage and consistency, as they use a replication method that allows data to be copied to other nodes in the system. One of the first studies in this area is ROE and was implemented because of the consistency of replicas, ease of setup, network transparency, and secure file authorization [Ellis and Floyd 1983, Perich et al. 2006]. Several studies have been conducted on the management of Big Data generated in education. In the study by [Kim et al. 2012], a system is recommended to be used in the Korean education system. The recommended system has proposed a recently used NoSQL system to manage Big Data instead of the classical RDBMS. In another study [Madani et al. 2017], Hadoop Distributed File System (HDFS) and MapReduce programming model were used for managing and analysing data to ensure that a student takes the most appropriate courses for them. [Khan et al. 2016] explores in detail how cloud-based Big Data analytics can be applied in Indian education.

The studies which are far from being unambiguous solutions for Big Data Management in distance learning are listed as follows.

1- HDFS

The use of HDFS is not a solution to big data in distance education for the reasons presented: i) the HDFS block size (128 megabytes by default) is not ideal for managing small files (especially smaller than 1 megabyte); ii) Hadoop requires skilled personnel for installation, management, and maintenance; iii) Hadoop is designed as a batch processing system.

2- Cloud service providers

The performance of the cloud service providers is insufficient for processing real-time data. Also cloud service providers have the disadvantages presented: i) charges for services rendered by the provider; ii) security issues; iii) one does not know exactly where the data is physically stored; iv) legal procedures become difficult in case of conflict; v) in Turkey, it is mandatory to store data of government institutions in local data centers.

3- NoSQL database systems

NoSQL systems also cannot provide a comprehensive solution to this problem due to the listed disadvantages: i) the need for qualified personnel; ii) requires high computer hardware requirements; iii) no universal language between NoSQL applications, unlike RDBMS.

Consequently, the characteristics of Big Data Management solutions for distance learning should have the features: i) cheap computer requirements and equipment; ii) simple design; iii) write once read many scheme; iv) easy horizontal scalability; v) data security. The proposed system was designed according to these characteristics.

2 Big Data

The term "Big Data" refers to a huge or composite data set where traditional data management systems are inadequate to process and manage the data [Ergüzen et al. 2018, Ünver et al. 2018]. Big data is usually explained with so-called V principles. One of these is known as 7V's as illustrated in Figure 1. These are velocity, variety, volume, value, veracity, variability, and visualization [Ciordas-Hertel et al. 2019]. Variety refers to all structured and unstructured data. Structured data is data that has a predefined length and format, such as a table form that is usually included in RDBMS, and is therefore easy to analyse, while unstructured data either has no predefined data model or has no length or type constraints. Velocity indicates the speed of incoming data or is the measure of how fast the data arrives. Data volume refers to the size of data sets, now larger than terabytes and petabytes, that need to be analysed and processed. Value criteria in data is expected to deliver a meaningful result that adds value to the business after the stages of data production and processing. In this way, they are used to directly influence the decision-making processes and make the right decision in a timely manner. Veracity means the data to be processed must come from a correct source and must be reliable. The accuracy of the results obtained by analysing the processed data is questionable. Such inconsistent and unclear data sets cannot be considered as Big Data. Visualization means after the data is analysed and visualized so that the end-users can better understand, evaluate, and act upon the results. Variability refers to the

inconsistency that data may exhibit from time to time which hinders the effective use and management of data.



Figure 1: 7Vs of Big Data

Traditional RDBMSs cannot provide solutions for Big Data due to data size and variety. DFSs and NoSQL databases are often preferred for persistent storage and management of large irregular data sets [Howard et al. 1988, Cattell, 2011]. Such programming frameworks have proven to be very successful in tackling clustering tasks, especially in ranking web pages. Various Big Data applications can be developed based on these innovative technologies or platforms [Chen et al. 2014]. Several challenges need to be overcome when dealing with Big Data. These challenges include data storage and analysis phases [Sin and Muthu 2015].

3 Related Works and Big Data in Distance Education

Distance learning using internet technologies (e-learning) is a method of education that has grown in popularity in recent years. E-learning is a formal method of instruction and communication between practitioners and students using specially designed Internet-based software and associated tools, all of which form a portal called an LMS, a web-based software that enables e-learning. It provides features, also known as modules, such as supporting a virtual classroom, presenting learning materials, uploading and downloading documents, collaborative discussion tools called forum or chat system on course topics, managing course catalogues, homework assignments, preparing and taking exams, providing feedback on homework and exams, organizing learning materials, tracking students and teachers, generating detailed statistical reports, providing performance monitoring tools, and other functionalities also supported by web technologies [Paulsen 2002, Ergüzen et al. 2021]. The available distance learning frameworks have two different specialized modules according to their

underlying software and hardware: i) LMS, where all training activities are gathered under one roof, as described above; ii) Virtual Classrooms Software (VCS) is a special type of live meeting room where participants listen to the instructor using a camera, microphone, and speaker over the Internet. These video files require a lot of storage space, and the file size increases as the recording time increases. According to records in the developed system, it occupies nearly 1.2 megabytes per minute. Online courses, education management systems, assignments, web-based training resources, and social media sharing also lead to an increase in educational data [Dwivedi and Roshni 2017]. A challenging problem that arises in this field is how to solve the Big Data problem effectively and efficiently. This is the core idea of this study. At the same time, educational institutions face a heavy workload managing and analyzing data obtained from educational resources to assess student achievement [Bamiah et al. 2018]. In the United States, in 2017, the number of students taking at least one distance education course increased by 3.9 percent from the previous year to 6,035,000, and in 2018, the number increased by 5.9 percent to 6,395,000 [Seaman et al. 2018]. In recent years, the data produced by the education sector has begun to increase the need for Big Data technologies and the tools used to manage them [Sin and Muthu 2015].

In 2016, Udupi and his colleagues presented a new paradigm for e-learning. In the study, an intelligent system was designed by integrating e-learning components. However, this framework is not a new system for data storage or file management [Udupi et al. 2016].

A web-based system was developed with a three-tier architecture running on a low-cost hardware configuration. The architecture created is used in the Moodle database source [Chaffai et al. 2017].

Technology-enhanced distance education has reached a large amount of data through course content, videos, images, messages, student assignments, transcripts, and supplemental documents (pdf, jpeg, docx, xlsx, pptx, etc.). In one of the recent studies, three levels of distance learning are defined: LMS, Content Management System (CMS), Virtual Learning Environments (VLE). The Hadoop system, which consists of a name node and two data nodes, was used to try to implement a solution for the Big Data generated by these layers [Dahdouh et al. 2018]. A total of 143 articles published between 2010 and 2018 were examined in a review study [Quadir et al. 2020]. The number of articles studied between 2001 and 2018 were 1, 7, 4, 17, 15, 17, and 42, respectively. This progressive growth in terms of number of publications shows an increasing interest in the field of Big Data in education. A study by Ciordas-Hertel et al. also searched the four databases IEEE, ScienceDirect, SpringerLink, and ACM, focusing on studies related to the keyword "Big Data and education" between 2015 and 2019. They examined 20 of these articles and found that the Hadoop cluster structure was the most used [Ciordas-Hertel et al. 2019]. In Wang and Zhao's study, modules were created on a Big Data cloud computing platform and user data was encrypted using the MD5 algorithm [Wang and Zhao, 2021]. Logica et. al. have tried to solve the fast growing Big Data problem in education using cloud technology. By analysing the Big Data, they have tried to find out its impact on the environment and provide it with a software to access it [Logica and Magdalena 2015].

Another study examined Big Data in education and emphasized that it can reveal students' abilities and help educational institutions make strategic decisions [Birjali et al. 2016]. Arthur and Zyl in their study found that the concept of Big Data is not clearly understood in higher education, so less efficient traditional methods are still used. They

examined 55 relevant articles from six computer science databases from 2007 to 2018, focusing on the progress of the Big Data framework and its applicability in education. As a result, they found that a comprehensive Big Data framework investigation is needed to create effective educational systems to improve teaching and learning quality [Otoo-Arthur and Van Zyl 2019]. Rodrigues et al. studied small file access and archiving in distributed file systems. Small files are accessed and stored in HDFS using the procedures Archive Files, New Hadoop Archive (New HAR), CombineFileInputFormat (CFIF) and Sequence File Generation. When evaluating the results obtained in the study, it was found that the storage performance for small files was improved, but there was no improvement in the read time performance [Rodrigues et al. 2021].

4 Kırıkkale University Big Data Projection

The data produced in distance education has reached a very large volume that is compatible with other Big Data oriented fields. In the example of Kırıkkale University in Turkey, Kırıkkale University Learning Management System (KULMS) has been in use since 2009. The research group that designed the structure is the same team that designed KULMS. Currently, 29780 students are active. 6394 courses in 252 departments have been opened and 826 faculty members are assigned. The data volume of the system is about 37.6 gigabytes for students' assignments only, 15.658 terabytes for virtual course videos and 62.2 gigabytes for supplementary materials. Soon the number of students and lessons will double, presenting us with the problem of how to organize, store, and archive the data generated.

Due to the pandemic crisis, face-to-face classes have been discontinued in all educational institutions around the world, especially in the first half of 2020, and complete distance learning has been introduced. In this challenging global world, technology-enabled distance education has become our salvation. As all courses have been converted to digital media, the amount of data produced and to be managed has increased tremendously. This process has once again shown that the system developed has its justification and that it is necessary to manage the data produced in distance education in the best possible way.

The larger the amount of data becomes, the more difficult it becomes to store, analyze and manage. As a result, data is perceived as big garbage that is not valuable material for e-learning. Unfortunately, institutions' decision makers have deleted the data instead of focusing on analyzing it. Until the Big Data problem can be solved, it is not possible to increase the level and quality of e-learning. For example, Table 1 shows that the total amount of data generated by the e-learning portal of Kırıkkale University has been steadily increasing. In particular, the increase of generated data during the pandemic period is very remarkable. This process has changed the way of education; now face-to-face teaching has been reinforced by e-learning. A major weakness of the portal is that the data generated in previous years has been deleted because the perception of Big Data is not well understood. The primary goal of LMS providers should be not only to capture Big Data, but also to ensure that the archived data can be retrieved quickly when needed. This is the focus of our study, to store data effectively and retrieve it quickly when needed.

The capacity of the e-learning system used at Kırıkkale University: i) 6394 lessons per week, 9591 hours of virtual classroom instruction; ii) archives are provided so that each lesson can be re-watched by students. Each virtual classroom on the platform generates almost 160 megabytes of data in one session.

Years	Assignment (MB-Midterm Exam)	Virtual Classroom (MB)	Course Content (MB)	Supplementary Materials (MB)
2020-2021 (fall semester-4 weeks)	16,720	7,067,200	24,800	17,000
2019-2020 (spring semester) (Pandemic crisis effect)	16,866	7,485,600	25,300	18,000
2019-2020 (fall semester)	866	336,000	15,100	12,500
2018-2019	1,604	388,000	15,100	14,700
2017-2018	1,588	382,000	15,100	removed
2016-2017	removed	removed	15,100	removed
Total Data Volume	37,644	15,658,800	110,500	62,200

Table 1: The volume of data stored in KULMS

In examining the general behaviour, it was found that at least one homework assignment was assigned by the teachers for each class. The average size of each assignment file is 40 kilobytes. Over the past four years, 37.644 gigabytes of homework files were stored in the system. Moreover, additional materials such as images, movies, PDF, and document files reached nearly 62.2 gigabytes. However, as shown in Table 1, due to the large amount of data, the institute has not archived important files including virtual classrooms, so only files from 2016. This data should also be considered Big Data.

5 Material and Method

- **TCP/IP:** This section deals with the most advanced technologies in this study. TCP/IP is one of them. Every computer or mobile device connected to a network has a unique Internet Protocol (IP) address that can be found individually on the same network. In fact, for each IP address, a device can be defined on the network.
- **Windows Service:** Windows services are a special type of application that runs in the background and has no user interface, usually serving the Windows Operating System function. Unlike an application, a service is a long-running

process until the computer is shut down and is started automatically when the computer is restarted. In this project, Windows service routines were used both server-side and client-side (DFS). These services also have server and client socket constructs for mutual data transfer.

- **Compression:** Compression uses an algorithm to reduce the size of one or more files to a smaller size. This makes storing and transferring files easier and faster. In this project, the .NET compression library has been used and GZipStream under System.IO.Compression used.
- **Programming Languages:** In this study, the application was developed using Visual Studio 2017 C# application development environment on .NET 4.6.1 framework.
- **Cryptography:** Cryptography is a set of techniques that attempt to provide information security such as privacy, authentication, and integrity. Data is exchanged from one point to another using the encryption key before transmission, and the receiver who receives the encrypted data converts this modified data into the original data using the appropriate decryption key. One of the fast symmetric block ciphers used in the literature is the Blowfish algorithm and has a key length between 23 and 448 bits [Schneier 1993, Sharma and Kaushik, 2019]. For these reasons, the Blowfish algorithm with a key length of 128 bits was preferred in the application. Asymmetric encryption is slower than symmetric encryption, hence the Blowfish algorithm was chosen.

6 Designed Distributed File System

Actively used LMSs have the following weaknesses.

- As the amount of data increases, the system becomes hard to manage and files that cannot be stored are deleted. This creates an archiving problem.
- As there is no optimized data management system, the performance decreases.
- The security of the data is the same as the used LMS. Since data storage and management is handled by the LMS, third-party applications that provide data security are not used. This may expose security vulnerabilities.
- Since the existing systems have a scaling problem, they are not suitable for growing data volumes. This is a barrier for distance education to reach larger audiences.

Students connect to the LMS and Virtual Classroom Applications (client applications) through the Internet and perform the required operations, and the applications store the collected data in their databases. These systems are comprehensive software that help students store, produce, share, and retrieve related information more efficiently and enable better tracking of learning outcomes. That is, e-learning applications use their databases and file servers to manage students and teachers, as shown in Figure 2.

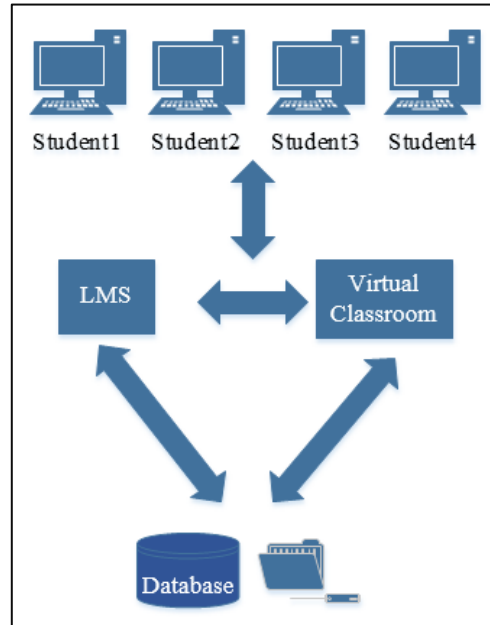


Figure 2: E-learning System Architecture

For data management, applications require a data layer that includes database engines such as MsSQL, ORACLE, MySQL, and the operating systems' file management utilities, as shown in Figure 3(i). The file system of the underlying operating system is used for basic operations such as inserting, deleting, searching, and copying files. The file system, which is the way files are organized on the hard disk, is the collection of methods and special data structures that an operating system must store and process files. Although the file system is sophisticated and specialized for file operations, it is not efficient in terms of horizontal scalability and data security. While the operating system ensures that the data is stored efficiently and successfully on only one computer, it does not contain the DFS features required to manage Big Data. The developed system has replaced the file management layer of the operating system and all the data storage and security services are provided by our system in this layer as shown in Figure 3(ii). At this point, all the data generated by e-learning applications are managed by our developed DFS which is the core of our work. The developed layer, named Remote Secure File System (RSFS), is a DFS-based system and therefore contains data and replication nodes. Client applications must use a specially developed Dynamic Link Library (DLL) file to access this layer. The current system, as described below, consists of i) DLL; ii) data layer nodes and iii) Windows service software (WSS) that manages these nodes (as shown in Figure 4).

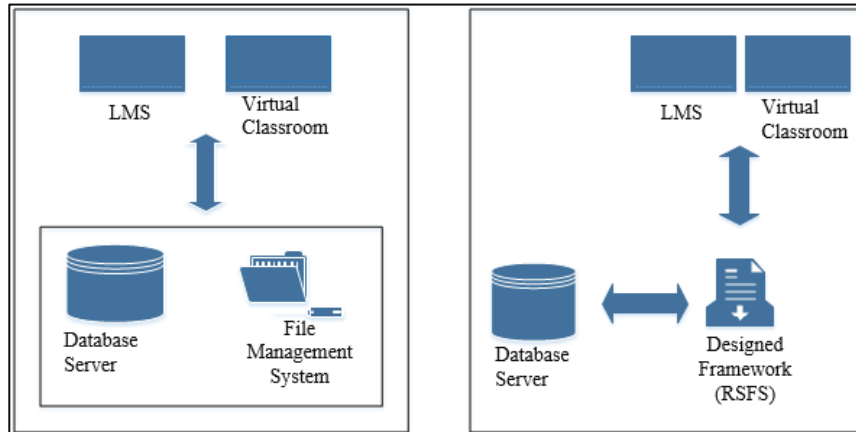


Figure 3: i) Current LMS Data Layer ii) Developed LMS Data Layer

Dynamic Link Library (DLL) is a library as shown in Figure 4(i) that contains code and data that can be used by programs without writing additional code. To use or enable DLL files, it must be embedded in the source code. That is, it is an intermediate layer that connects to client applications and RSFS. The DLL is integrated into the LMS system and performs read and write operations. This module is very important because it allows the LMS to access the data source and performs the following commands:

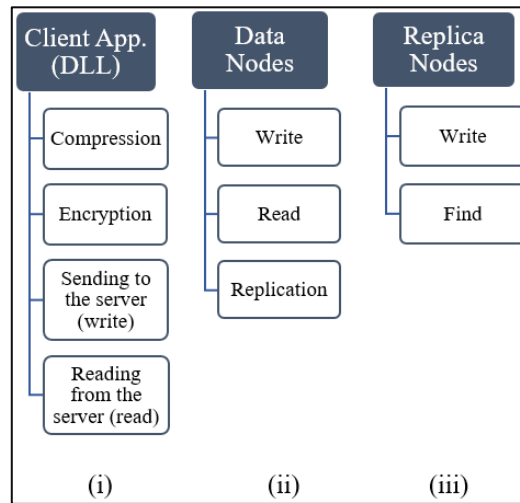


Figure 4: Frameworks of the designed system

- Write: Sends the file to the data node.
- Read: Reads the file through the data node.
- Read Replica: It reads the file through the Replica Node. To perform these operations, the information of the data nodes registered in the system is needed

first. They are uploaded to the client application using a special file and this file is reloaded when existing data node properties are changed, or a new data node is added to the system. The following tables show the structure of the registered data node information. The client application sends the files to the active data nodes one by one (node list in Table 2) so that the data does not pile up on a single node but is distributed to other nodes.

ID	IP	Active/On-Off	Type
10101	192.255.***.***	1	Data
20101	192.255.***.***	1	Replica
20102	192.255.***.***	0	Data
30101	192.255.***.***	1	Data

Table 2: Registered node file structures

Windows services allow to create long-running executable applications that start automatically when the computer boots, can be paused and restarted, and do not display a user interface. Because these services are quick and easy to manage, they are used in all data nodes. These services work in all data nodes and have the following tasks: i) Establish a secure connection with the DLL installed on Client Applications; ii) send and receive data with the DLL; iii) store data by connecting to Windows Services on the other data nodes; iv) retrieve data from other data nodes. The DLL constantly connects to the WS installed on the data node and manages the data traffic.

Data Nodes: WSS has been installed on all nodes of the system as shown in Figure 4(ii). This software manages all the data stored on the computer. It is also the responsibility of the client application to ensure a secure connection, store the data on the storage device, provide feedback to the client, and write data to the replication nodes. The data nodes are responsible for managing the storage devices, in Figure 6. One of the most important features of the system is that has no constant block size. The file to be stored is written to a continuous area in the 2 TB main file of the system. This increases the read performance for large files, and small files can also be stored easily. There are three data nodes and two replication nodes in the system. The proposed and tested system has a serverless structure. The "serverless" scheme does not mean that no servers are used. This approach only means that the system does not need a main node to function since all data nodes in our system have the same priority.

Client Application means LMS and virtual classroom programs. These programs can send and receive data to RSFS using the developed DLL. By using the DLL, the applications are device-independent, they do not know where and how the data is stored. In the developed framework, both modules, LMS and video server in Figure 4(i), are connected to the platform through DLL files. Whatever task CA wants to perform, it uses the classes and methods available in the corresponding library. CA performs the following tasks using the DLL as shown in Figure 4(i):

- Compresses and encrypts files to be sent to nodes.
- Sends files to data nodes for storage (write).
- Reads files stored on the data node (read).
- Decompresses and decrypts the received files.

System integration: performed as described in 3 steps as shown in Figure 5 below.

- The DLL software is loaded into the client application.
- The WSS is installed on data nodes.
- The WSS is installed on the replication nodes.

Scalability: The issue should be examined from two points of view: vertical and horizontal. Vertical scalability is also called "scale up" and means changing the physical properties of the node. Horizontal scalability is also referred to as "scale out" and means adding more computers to the system.

The developed system fully supports horizontal scalability for the data nodes. Vertical stability is partially supported by instant CPU and RAM increase except for data storage. Hadoop supports both vertical and horizontal scalability for data nodes, but name node only supports vertical scalability. While many SQL databases support vertical scalability, NoSQL databases support horizontal scalability.

In the developed system, there are steps to be followed when a new data node is to be added to the system. These are in the order:

- Step 1. The new data node must be included in the ecosystem so that it can communicate with CA and write incoming file requests to disk. The WS software is constantly listening on the ports to respond immediately to requests from CA. Therefore, the WS software files that enable all these operations must be installed on this data node;
- Step 2. An IP address is a unique address used by devices connected to the Internet or other packet-switched networks that use the TCP/IP protocol to exchange data with each other over the network. A static IP address is the constant IP address of the device that is connected to the Internet. Therefore, each data node added to the ecosystem requires and is identified by a static IP address. Also, all IP addresses used in the ecosystem are stored in a file (as shown in Table 2);
- Step 3. The basic characteristics of each node in the ecosystem are listed in Table 2. There are static IP addresses that provide access to the nodes, the unique ID information of the node in the ecosystem, the activation status (1, 0) that contains the current status of the node, and the node type fields that indicate whether the node is a data node or replica node. For this reason, the static IP address specified in Step 2 is added to the file structure specified in Table 2 with the appropriate parameters.
- Step 4. CA provides data management using the information in Table 2. When a data node is added to the ecosystem, the list in CA must be updated to send data to it as well. For this reason, the updated Table 2 in Step 3 is sent to the CA.

Another phase of horizontal scalability is to easily adapt to the changing IP address of the nodes. Due to the nature of a static IP address, it is very rare for it to be changed once it has been defined for a device. However, if necessary, changes are supported by the system. Each data node in the ecosystem is identified by its unique ID value. Therefore, the changes that occur to the IP addresses of the nodes are updated in Table 2 and sent to the CA.

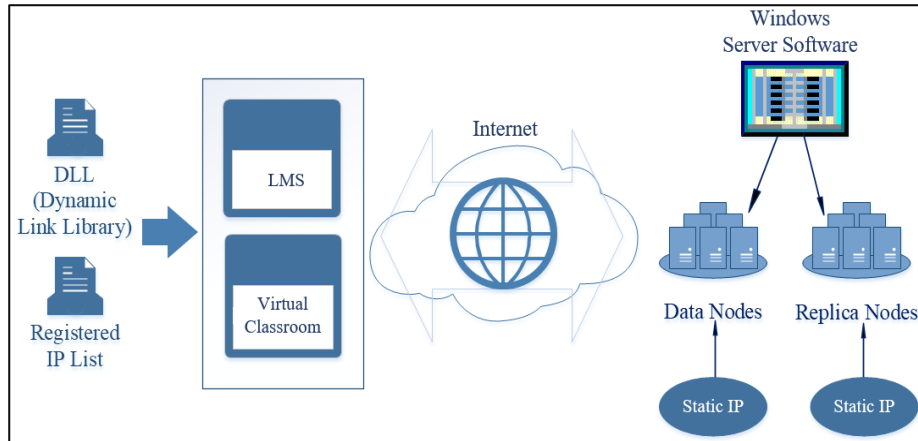


Figure 5: System integration overview

Disc 0 Header	Device ID 4 B.	Next File Pointer 8 B.	Size 8 B.	Replica1 Node ID 2 B. Replica1 Node IP 4 B. Replica2 Node ID 2 B. Replica2 Node IP 4 B.
---------------	-------------------	---------------------------	--------------	--

Figure 6: The structure of the storage device

Replica Node: These nodes are also managed by the WSS and write the specified files to their hard disk in Figure 4(iii). The file structure of the Windows operating system was used to store the files. These nodes are customized for writing and searching the desired file on the disk. These nodes are customized for writing and searching for the desired file on the disk. Files are saved and read on the underlying operating system, and no special file structure is designed. Replica node usage ratio is very low because CA needs replica nodes in the following cases:

- When network traffic is heavy and time out error occurs.
- When there is a maintenance on the target data node.
- When the target data node is physically inaccessible. The system has turned to replica nodes at a rate of only 3 per thousand, considering the total 1-year requests.

The flowchart of the implemented system is shown in Figure 7. The general structure of the system is described in the following sections.

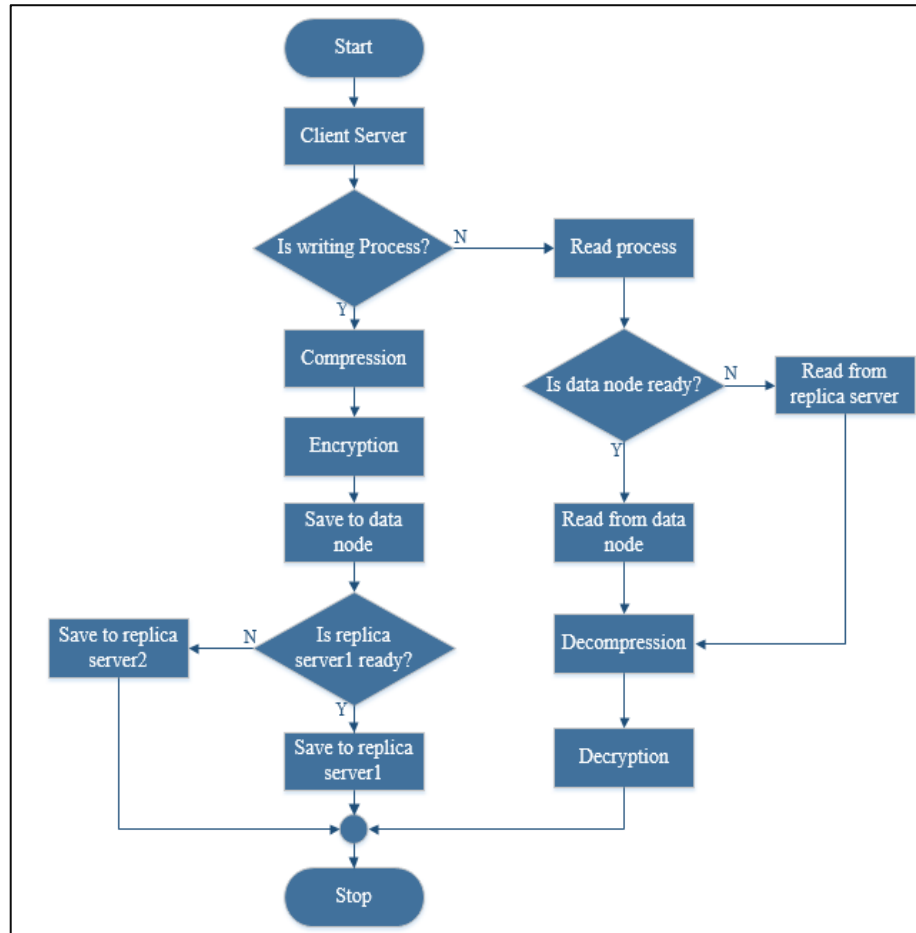


Figure 7: The flowchart of the implemented system

As shown in Figure 8, the LMS and video server software is connected to the server via DLL files. The client can perform read and write operations. The data is compressed and encrypted before writing. They are then sent to the server as a data stream for writing [Figure 7, part a]. For this operation, the client searches for the appropriate data node and sends the data packets. The data node in the server knows what to do by looking at the headers of the packets that reach it. In a write operation, the incoming data packets are managed by WSS on the data node. For write operations, WSS is responsible for organizing enough consecutive free space for the uploaded files.

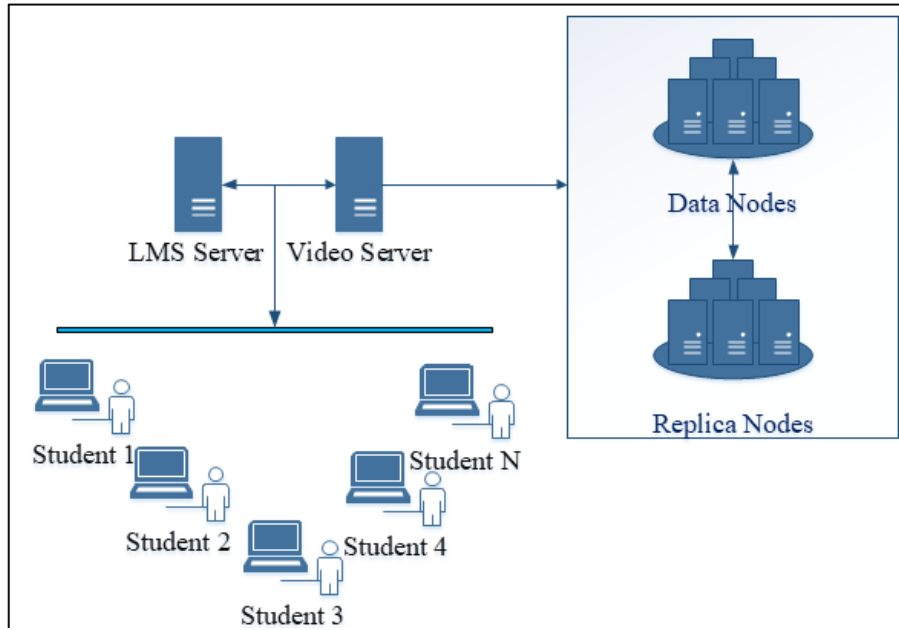


Figure 8: The structure of the implemented system

In other words, as shown in Figure 9, the file sent is stored as a single block in the developed file system. Here, the file is stored sequentially, i.e., all characters are written into a contiguous block with the necessary header information sent by the client [Figure 10]. In this way, unnecessary seek times (s) and rotation latencies (r) on disk are avoided. This process is returned as performance. Each data node contains 1 disk with a capacity of 2 terabytes. As shown in Figure 9, the title information of each file is fixed at 65 bytes. Each file header consists of (8 bytes) the date and time of file creation, (2 bytes) the data node identification number, (4 bytes) the IP value of the data node, (50 bytes) the stored file name, and (1 byte) the information whether the file is written to the replication node. In Figure 8, 100 megabytes of disk space is reserved. The title information of each file, which is 65 bytes, is written to this field. Thus, 1,613,193 files can be uploaded to the system ($104,857,600 \text{ B} / 65 \text{ B}$), which is the physical limit of the system per data node.

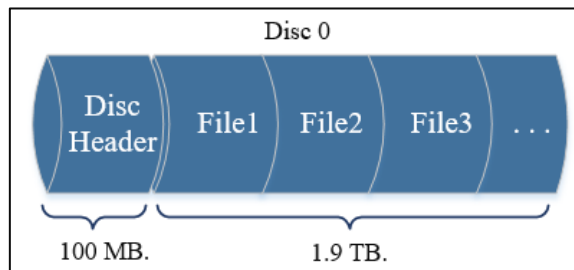


Figure 9: Storage device structure

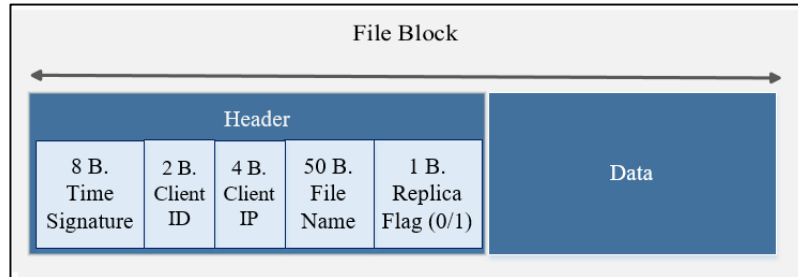


Figure 10: The file block structure of the designed system

7 System Statistics and Performance Analysis

All academic staff and students at our university can access the distance education system. The system files are divided into six different sizes as seen below. The total files uploaded to the LMS system during the Spring 2020 and Fall 2021 semesters were examined and presented in Table 3. As seen, 67.72% of the files in the distance learning system have a small file size (≤ 16 MB). In fact, 53.10% of the files are smaller than 1 MB. Distance Education produces a large amount of data because it contains a wide variety of educational tools. Volume, which is one of the characteristics of Big Data, refers to the speed and size of the data rather than the file size. The number of small files (50+ MB) in the distance education platform of Kırıkkale University is about 68 percent of the total files. So, one of the obstacles in Big Data processing is managing small files; systems that support large file structures, such as Hadoop, cannot be very useful here. One of the important goals of this study is to be able to efficiently solve Big Data that consists of small files.

File Size	Number of Files	Total File Size (MB)	Percentage (%)
<10KB	3,936	16	0.85
10KB-100KB	135,555	1,980	29.19
100 KB-1MB	107,076	28,120	23.06
1 MB-16 MB	67,902	75,790	14.62
16 MB-50 MB	1,968	12,780	0.43
>50 MB	147,887	14,552,800	31.85

Table 3: Details of files on the system grouped by file sizes

The number of files written for the two semesters are grouped by their size and the month in which they were uploaded and are shown in Figure 11. As can be seen from the figure, the platform is used more frequently when the education and training activities are continued. According to the values obtained, there are 3,936 files with file size less than 10KB; 135,555 files between 10KB and 100KB; 107,076 files between 100KB and 1MB; 67,902 files between 1MB and 16MB; 1,968 files between 16MB and 50MB and finally 147,887 files with more than 50MB. In these two semesters, a

total of 464,324 files were written by the system, i.e., a monthly average of 46,432 files were uploaded.

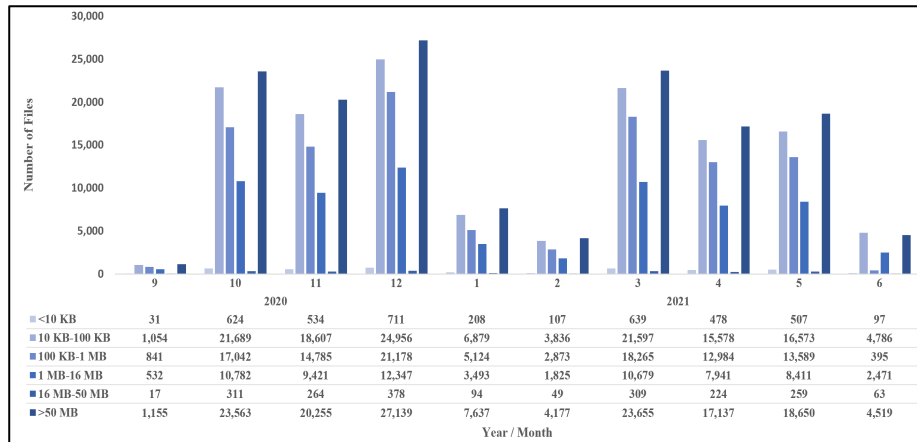


Figure 11: File write counts by month and file size

The number of files read for the two periods are grouped according to their size and the month in which they were read and are shown in Figure 12. According to the values obtained, there are 11,706 reads with a file size of less than 10KB; 349,581 reads between 10KB and 100KB; 272,804 reads between 100KB and 1MB; 196,687 reads between 1MB and 16MB; 5,637 reads between 16MB and 50MB and finally 3,099,444 reads of more than 50MB. During these two semesters, a total of 3,935,859 files were read by the system, so that a monthly average of 393,585 files were read. This means that each file written to the system was read 8.47 times. In addition, each video file in the system (recordings of lectures in the virtual classroom) was viewed an average of 20.95 times. This value proves the importance of the archive used on the platform.

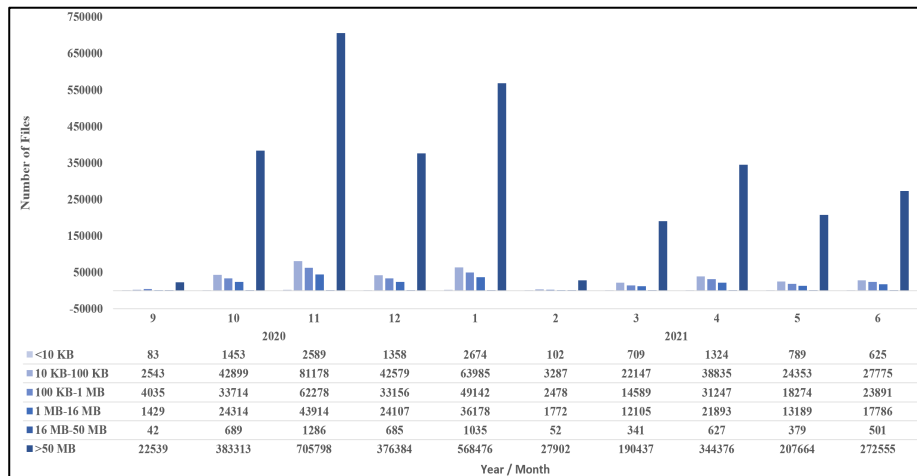


Figure 12: File read counts by month and file size

To evaluate the performance of the proposed system, comparisons of different file sizes on common data processing systems were performed [Table 4]. The technical characteristics of the compared systems are as follows [Ergüzen and Ünver 2018].

- The configuration of the Hadoop system includes a name node and three data nodes, Red Hat Enterprise Linux Server 6.0, Java-1.6.0, and Hadoop-2.7.2, installed on each node [Hadoop 2006].
- Couchbase, one of the most popular NoSQL databases, was chosen for testing [Couchbase 2011]. Each Couchbase bucket consists of 20 megabytes of clusters. The test machine has 6 gigabytes of RAM and 200 gigabytes of storage space.
- Microsoft SQL Server 2014, which we currently use in other projects, was preferred, and installed on the same machine.
- The DLL files of the system we are developing are also installed on this machine.
- Various files ranging in size from 30 kilobytes to 50,000 kilobytes were used.

All software except Hadoop (for name node) was run on the same machine to ensure that the comparison was made under the same conditions. The hardware configuration of the Hadoop nodes is identical to that of the other machine. The main physical server configuration consists of two Intel Xeon E5-2620 v4 2.10 GHz processors (8 cores) workstation, 64 gigabytes of RAM and 512 gigabytes of SSD and 1 terabytes of SATA SAS hard drives.

File Size (kilobytes)	Proposed System	OS	NoSQL	RDBMS	Hadoop
30	0.01	0.04	0.60	0.75	0.80
1,000	0.95	1.43	2.74	3.15	4.01
10,000	4.16	4.48	8.44	9.97	11.15
20,000	8.79	11.19	18.01	18.16	20.45
30,000	11.87	14.80	27.22	27.80	30.15
50,000	22.35	24.54	43.95	44.08	47.88

Table 4: The response times for different file sizes (s)

The proposed system shows that it performs better performance than other tools. The reasons are as follows:

- The proposed system has only one layer (data nodes only), while other systems have more than one layer depending on their own architecture.
- Low-level read and write operations (windows file API) are performed.
- The files are written sequentially to the disk, so the read time is significantly reduced as there is no fragmentation of the files. Since the operating system specializes in the file management, the second-best result belongs to it. NoSQL databases are known to be better than RDBMS by performance criteria [Ali et al. 2019, Yoo et al. 2018].

To better evaluate system performance, a test plan was created (a file of 10 KB to 100 KB) and measurements were performed according to this scenario. For each test to be performed under the test plan, previously determined six different file sizes were

used and their average values were calculated using a third-party application. The following system tests were performed under the scenario:

- 1 Read: Average time for 1 read per file size.
- 1 Write: Average time for 1 write operation per file size.
- 20 Read (sequential): Average time of 20 sequential reads per file size.
- 20 Write (sequential): Average time of 20 sequential writes per file size.
- 20 Writes (mixed): Average time of 20 random reads and 20 random writes per file size.

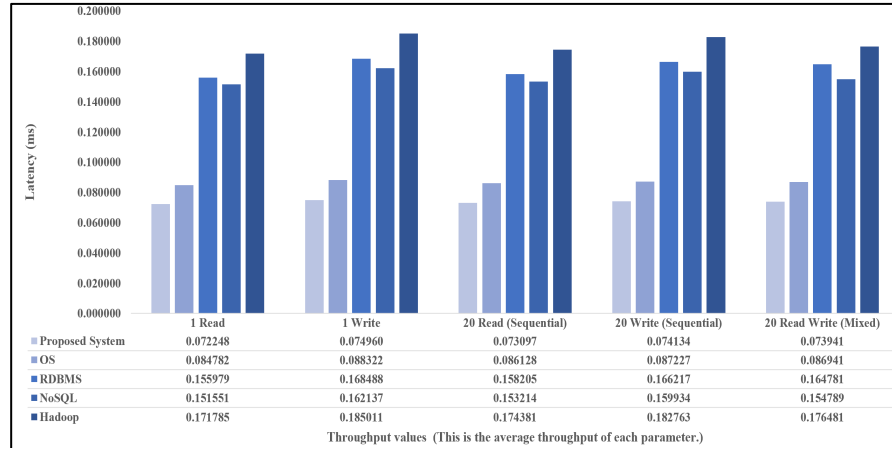


Figure 13: System performance comparison in different scenarios

The performance of the developed distance learning system was significantly better on all criteria, as can be seen in Figure 13. However, if we compare the reading and writing performances of the other systems with each other, we find that the reading performance is better. The reason for this result could be the cache structure in the systems. Also, since the developed system uses dynamic block size, the read and write operations are more efficient. On the other hand, other systems divide the files into blocks and write them to different regions on the hard disk, which negatively affects their performance.

Therefore, the most important features of the developed framework are the following:

- Small file size: most files created in distance learning have a small file size (Table 3). For this reason, there is a need for an ecosystem to store small files efficiently. The developed system has provided a solution to this problem.
- Dynamic block structure: systems that manage large amounts of data are often designed to have a large file size. A large file size leads to internal fragmentation for small files. In systems with dynamic block structure, there is no fragmentation problem. The larger the file, the larger the block size, so each file is written in a single block regardless of its size.
- Serverless architecture: in DFS systems, a master node usually works (there is also 1 secondary master node). If this node fails in any way, the entire system

is affected. Therefore, in the developed system all the nodes work independently, the nodes are accessed through the DLL installed in the client application.

- Compression and encryption: after the data is compressed at the client-side, it is encrypted and sent to the server. In this way, i) the data is sent faster; ii) less space is required on the server; iii) the data is kept secure.

8 Conclusion

With ever-evolving technological devices, much more data is being produced that is unimaginable compared to previous generations. The defense industry, pharmaceutical industry, meteorological processes, IoT devices, all the devices and servers on the internet, smart and mobile devices that are changing the world of education are all producing a huge amount of information in various formats at a high rate called Big Data, and at an accelerating pace. Big Data is one of the difficult problems in computer science that needs to be solved and overcome. Big Data can be broken down into three parts: small, medium, and large data sets to manage them more efficiently and easily. So far, no separate solution has been developed for each data set. Methods or tools developed for managing and storing large data sets are used for others, resulting in an inefficient and unmanageable system for medium-sized systems. Since Big Data platforms have unique characteristics and problems, solutions should be created according to these needs for an efficient and economical solution.

The requirements and characteristics of Big Data scaled distance learning are i) no need for hundreds or thousands of data nodes to store data as Hadoop and other systems can support; ii) large number of files with small volumes, which is an obstacle for alternatives like Hadoop (version 2.0) with a default block size of 128 megabytes; iii) the need for many data storage systems with write once and read many; iv) limited budget and resources; v) few skilled personnel. If these problems are solved, the distance education system will be more efficient, easier to manage and economical. This is the main objective of our study. Thus, a DFS was developed to manage and efficiently use the big data produced by the distance education ecosystem. In this way, i) archiving, the most important problem of the distance education system, has been solved; ii) course documents, lecture content, homework, auxiliary resources and video files can be easily stored; iii) no backup operations are required; iv) data security and confidentiality has been maximized as data is stored encrypted using Blowfish algorithm; v) data transfer time has been reduced by compressing and transmitting it to the server and data storage area has been reduced by about 30%.

This study was one of the first attempts to thoroughly investigate and address the Big Data issues that arise in the technology-based ecosystems of distance education. Unfortunately, the study did not consider the parallel processing of data nodes. This is because the scope of this study was limited to solving Big Data storage problems rather than developing a Big Data analytics framework. How Big Data is stored and processed is equally important in today's digital world [Romeike 2019]. More needs to be done to develop Big Data analytics tools. The result of this study has many important implications for future practice, as described above.

Furthermore, it is necessary to work on the issues of handling multi-threading capabilities for data nodes and fully vertical scalability for data storage devices in the future.

References

- [Ali et al. 2019] Ali, W., Shafique, M. U., Majeed, M. A., & Raza, A. (2019). Comparison between SQL and NoSQL databases and their relationship with big data analytics. *Asian Journal of Research in Computer Science*, 1-10.
- [Alsberg and Day 1976] Alsberg P. A., & Day J. D. (1976). A Principle For Resilient Sharing Of Distributed Resources. In *ICSE '76 Proceedings of the 2nd international conference on software engineering*, San Francisco, California, USA.
- [Bamiah et al. 2018] Bamiah M. A., Brohi S. N., & Rad B. B. (2018). Big Data Technology In Education: Advantages, Implementations, And Challenges. *Journal of Engineering Science and Technology Special Issue on ICCSIT*, p. 229 – 241.
- [Birjali et al. 2016] Birjali, M., Beni-Hssane, A., & Erritali, M. (2016). Learning with big data technology: The future of education. In *International Afro-European Conference for Industrial Advancement* (p. 209-217). Springer, Cham.
- [Cattell 2011] Cattell R. (2011). Scalable SQL and NoSQL data stores. *ACM Sigmod Record*, 39(4), p. 12-27.
- [Chaffai et al. 2017] Chaffai A., Hassouni L., & Anoun H. (2017). Real-Time Analysis of Students' Activities on an E-Learning Platform based on Apache Spark, in (IJACSA) *International Journal of Advanced Computer Science and Applications*, 8 (7), p. 101-109.
- [Chen et al. 2014] Chen M., Mao S., & Liu Y. (2014). Big data: A survey. *Mobile networks and applications*, 19(2), p. 171-209.
- [Chervyakov et al. 2019] Chervyakov N., Babenko M., Tchernykh A., Kuchеров N., Miranda-López V., & Cortés-Mendoza J. M. (2019). AR-RRNS: Configurable reliable distributed data storage systems for Internet of Things to ensure security. *Future Generation Computer Systems*, 92, 1080-1092.
- [Ciordas-Hertel et al. 2019] Ciordas-Hertel, G. P., Schneider, J., Ternier, S., & Drachsler, H. (2019). Adopting Trust in Learning Analytics Infrastructure: A Structured Literature Review. *J. UCS*, 25(13), 1668-1686.
- [Couchbase 2011] Couchbase. (2011). Retrieved from: <https://www.couchbase.com>
- [Dahdouh et al. 2018] Dahdouh, K., Dakkak, A., Oughdir, L., & Messaoudi, F. (2018). Big data for online learning systems. *Education and Information Technologies*, 23(6), 2783-2800.
- [Dwivedi and Roshni 2017] Dwivedi S., Roshni V. K. (2017). Recommender system for big data in education. In *2017 5th National Conference on E-Learning & E-Learning Technologies (ELELTECH)*, IEEE. p.1-4.,
- [Ellis and Floyd 1983] Ellis C. A. & Floyd R. A. (1983). The ROE File System, In *3rd Symposium on Reliability in Distributed Software and Database Systems*, Clearwater Beach, FL, USA.
- [Ergün et al. 2013] Ergün U., Eken S., & Sayar A. (2013). Güncel Dağıtık Dosya Sistemlerinin Karşılaştırmalı Analizi. In *6. Mühendislik ve Teknoloji Sempozyumu*, p. 213-218.

- [Ergüzen and Ünver 2018] Ergüzen, A., & Ünver, M. (2018). Developing a file system structure to solve healthy big data storage and archiving problems using a distributed file system. *Applied Sciences*, 8(6), 913.
- [Ergüzen et al. 2018] Ergüzen, A., Erdal, E., & Ünver, M. (2018). Big Data Challenges and Opportunities in Distance Education. *International Journal of Advanced Computational Engineering and Networking*, 6(2), 35-38.,
- [Ergüzen et al. 2021] Ergüzen, A., Erdal, E., Özcan, A., & Ünver, M. (2021). Improving Technological Infrastructure of Distance Education through Trustworthy Platform-Independent Virtual Software Application Pools. *Applied Sciences*, 11(3), 1214.,
- [Hadoop 2006] Hadoop. (2006). Retrieved from: <http://hadoop.apache.org/>.
- [Howard et al. 1988] Howard J. H., Kazar M. L., Menees, S. G., Nichols D. A. Satyanarayanan M, Sidebotham RN, & West MJ. (1988). Scale and performance in a distributed file system. *ACM Transactions on Computer Systems (TOCS)*,; 6(1), p. 51-81.
- [Kaseb et al. 2019] Kaseb, M. R., Khafagy, M. H., Ali, I. A., & Saad, E. M. (2019). An improved technique for increasing availability in big data replication. *Future Generation Computer Systems*, 91, 493-505.
- [Khan et al. 2016] Khan S., Shakil K. A., & Alam M. (2016). Educational intelligence: applying cloud-based big data analytics to the Indian education sector. In 2016 2nd International Conference on Contemporary Computing and Informatics (IC3I), IEEE, p. 29-34.
- [Kim et al. 2012] Kim T., Cho J. Y., & Lee B. G. (2012). Evolution to smart learning in public education: a case study of Korean public education. In IFIP WG 3.4 International Conference on Open and Social Technologies for Networked Learning, Springer, p. 170-178.
- [Liao et al. 2015] Liao B., Yu J., Zhang T., Binglei G., Hua S., & Ying C. (2015). Energy-efficient algorithms for distributed storage system based on block storage structure reconfiguration. *Journal of Network and Computer Applications*, 48, p. 71-86.
- [Logica and Magdalena 2015] Logica B., & Magdalena R. (2015). Using Big Data in the Academic Environment, *Procedia Economics and Finance*, 33, p. 277-286
- [Madani et al. 2017] Madani Y., Bengourram, J., Erritali M., Hssina B., & Birjali M. (2017). Adaptive e-learning using genetic algorithm and sentiments analysis in a big data system. *International Journal of Advanced Computer Science And Applications*, 8(8), p. 394-403.
- [Otoo-Arthur and Van Zyl 2019] Otoo-Arthur, D., & Van Zyl, T. (2019). A systematic review on big data analytics frameworks for higher education-tools and algorithms. In *Proceedings of the 2019 2nd International Conference on E-Business, Information Management and Computer Science* (p. 1-9).
- [Paulsen 2002] Paulsen F. (2002). Online Education Systems: Discussion and Definition of Terms, NKI Distance Education. [http://www.porto.ucp.pt/open/curso/modulos/doc/Definition %20of%20Terms.pdf](http://www.porto.ucp.pt/open/curso/modulos/doc/Definition%20of%20Terms.pdf). Accessed November, 3.
- [Peng et al. 2020] Peng R., Xiao H., Guo J., & Lin C. (2020). Optimal defense of a distributed data storage system against hackers' attacks. *Reliability Engineering & System Safety*, 106790.
- [Perich et al. 2006] Perich, F., Joshi, A., & Chirkova, R. (2006). Data management for mobile ad-hoc networks. In *Enabling technologies for wireless e-business* (pp. 132-176). Springer, Berlin, Heidelberg.

- [Rodrigues et al. 2021] Rodrigues ,A., Fernandes, R., Vijaya, P., & Chander, S., (2021). Performance Study on Indexing and Accessing of Small File in Hadoop Distributed File System. *Journal of Information & Knowledge Management*. 2150051. DOI:10.1142/S0219649221500519.
- [Romeike 2019] Romeike, R. (2019). The Role of Computer Science Education for Understanding and Shaping the Digital Society. In *International Conference on Sustainable ICT, Education, and Learning*. Springer, Cham. p. 167-176.
- [Schneier 1993] Schneier, B. (1993). Description of a new variable-length key, 64- bit block cipher (Blowfish). In *Fast Software Encryption Second International Workshop*, Leuven, Belgium, December 1993, Proceedings, SpringerVerlag, ISBN: 3-540-58108-1, pp.191- 204, 1994.
- [Seaman et al. 2018] Seaman J. E., Allen I. E., & Seaman J. (2018). *Grade Increase: Tracking Distance Education in the United States*. Babson Survey Research Group.
- [Sharma and Kaushik, 2019] Sharma, S., & Kaushik, B. (2019). A survey on internet of vehicles: Applications, security issues & solutions. *Vehicular Communications*, Volume 20, 2019, 100182, ISSN 2214-2096, <https://doi.org/10.1016/j.vehcom.2019.100182>.
- [Sin and Muthu 2015] Sin K., & Muthu, L. (2015). Application Of Big Data In Education Data Mining And Learning Analytics-A Literature Review. *ICTACT Journal on Soft Computing*, 5(4).
- [Udupi et al. 2016] Udupi, P. K., Malali, P., & Noronha, H. (2016, March). Big data integration for transition from e-learning to smart learning framework. In *2016 3rd MEC International Conference on Big Data and Smart City (ICBDSC)* (p. 1-4). IEEE.
- [Ünver et al. 2018] Ünver, M., Erdal, E., & Ergüzen, A., (2018). Big Data Example In Web Based Learning Management Systems. *International Journal of Advanced Computational Engineering and Networking*, 6(2), 39-42.,
- [Quadir et al. 2020] Quadir, B., Chen, N. S., & Isaias, P. (2020). Analyzing the educational goals, problems and techniques used in educational big data research from 2010 to 2018. *Interactive Learning Environments*, 1-17.
- [Yoo et al. 2018] Yoo, J., Lee, K. H., & Jeon, Y. H. (2018). Migration from RDBMS to NoSQL Using Column-Level Denormalization and Atomic Aggregates. *Journal of Information Science & Engineering*, 34(1).
- [Wang and Zhao, 2021] Wang, J., & Zhao, B. (2021). Intelligent system for interactive online education based on cloud big data analytics. *Journal of Intelligent & Fuzzy Systems*, (2021), 1-11.