ISSN: 2688-9323

Design and Implementation of Course Resource Management System based on Hadoop Private Cloud

Yongling Liu*, Taizhi Lv, Xuejun You

School of Information Engineering, Jiangsu Maritime Institute, Nanjing 211170, China

*594697472 @qq.com

Abstract

The performance of traditional storage technologies and the existing security vulnerabilities in data storage have made it increasingly challenging to meet the growing demands of educational informatization. Therefore, it has become imperative to leverage big data storage and cloud computing platforms to establish a virtualized data storage infrastructure as the foundation for storing educational resources in vocational institutions. In order to better manage course resources, this paper designs and implements a course resource management system based on a Hadoop private cloud. The system utilizes Hadoop to build a distributed storage platform, develops a cloud disk module based on Hadoop's distributed file system API, and implements the course resource management system using front-end and back-end separation technology.

Keywords

Hadoop; Course Resource Management; Private Cloud; Front-end and Back-end Separation; Distributed File System.

1. Introduction

The development of Internet technology continues to change people's lifestyles, and the process of informatization in the education sector is accelerating with the driving force of Internet technology. Cloud computing is a computing model that provides resources as services, offering dynamic scalability and the ability to centralize various types of resources on a cloud platform for collaborative work [1]. It is a service-oriented approach to providing access to business and storage functions externally. Many leading companies have invested significant human and financial resources in developing their own cloud computing applications. In the academic field, many experts and research institutions have also devoted considerable effort to cloud computing, producing numerous outstanding research results. The full application of cloud computing in the education sector can provide effective services for teaching, subject experiments, self-study, and post-class tutoring, among other activities. As cloud computing applications in education mature, the education sector is undergoing a transformation from basic, simple computer-assisted instruction to complex, interactive cloud-assisted instruction. Cloud-based educational platforms can not only improve teaching efficiency for teachers and facilitate interactive learning for students but also enhance students' critical thinking abilities and promote collective intelligence development, thus improving overall teaching quality [2-3].

Currently, the infrastructure of most higher vocational institutions' teaching resource centers still relies on traditional storage structures built around servers [4]. This means that an information system is installed and deployed on a central server, which serves as the basic equipment for running application systems and stores data generated by these information systems directly on its local disk. With the rapid development of information technology in higher education, data has experienced explosive growth, gradually exposing weaknesses and deficiencies in traditional data storage methods. These weaknesses can be summarized as follows:

Lack of security in centralized storage of teaching resources under traditional storage methods.

Issues with storage space wastage and the inability to achieve secure and straightforward sharing.

Lack of time-based teaching resource snapshot backups and data recovery.

Insufficient data backup and recovery to protect the entire data system.

Low server utilization due to continuous data growth, leading to an increase in the number of servers and reduced utilization rates.

The performance of traditional back-end data storage platforms and the security vulnerabilities in data storage have made it increasingly challenging to meet the growing demands of educational informatization. Therefore, it has become imperative to leverage big data storage technology and cloud computing platforms to establish a virtualized data storage infrastructure as the foundation for storing educational resources in higher vocational institutions.

This paper constructs a course resource management system based on a Hadoop private cloud. The implementation of this system effectively addresses the challenges of resource sharing for teachers, leading to improved teaching efficiency for educators and enhanced learning efficiency for students to some extent.

2. Requirement Analysis

The system is horizontally divided into three main modules: the cloud disk module, the course resource management module, and the course resource application module.

2.1. Cloud Disk Module

The cloud disk module is built on the Hadoop private cloud, providing online cloud storage functionality to meet users' needs for handling personal file data in the cloud. This cloud disk moudle enables the management of personal files, public files, and user accounts. Personal file management includes features such as file uploading, downloading, renaming, and deletion of files and folders. Public file management allows operations on shared files and the allocation of permissions. The user management module is responsible for managing user accounts and configuring personal spaces.

2.2. Course Resource Management Module

The course resource management module is based on the interfaces provided by the cloud disk module and is designed for course-specific educational resource management. Teachers can create websites for specific courses through the back-end management interface. Each course's website is created separately for different courses, utilizing existing primary data tables. This is achieved by duplicating the primary data tables, changing their names to create specialized data tables for the sub-sites, and establishing exclusive folders on the resource server to store resources. The front-end can specify the appropriate data table by transmitting different specialty names to the back-end. The resource classification module consists of two parts: the homepage tag module and the secondary tag module. The homepage tag module allows for the setting of tags, which will appear on the homepage of the corresponding specialty's website. Teachers can customize the names and quantities of these homepage tags, and automatically created websites come with commonly used tags for convenience. The secondary tag module, on the other hand, requires binding to the corresponding homepage tag when creating new secondary tags. These tags can be changed by teachers as needed within the teacher interface, but deleting a homepage tag will also remove its associated secondary tags. Secondary tags are

ISSN: 2688-9323

displayed on the left side of the webpage when students enter the corresponding homepage tag page. Clicking on a secondary tag displays the resources associated with that tag.

2.3. Course Resource Application Module

The course resource application module primarily caters to student users and anonymous users, enabling course browsing, downloading, viewing, and accessing relevant notifications. This fulfills the needs of students for online learning. The system restricts resource usage within the campus network and verifies student logins to ensure both the circulation of uploaded resources among internal users and the security of resources and websites. The rapid website creation feature allows non-specialized teachers to create course resource websites for their respective specialties.

3. System Design and Implementation

3.1. Module Design

As shown in Figure 1, the system's functionality is primarily divided into four main parts: Distributed File Platform, Cloud Module, Cloud Course Resource Management Module, and Cloud Course Resource Application Module.

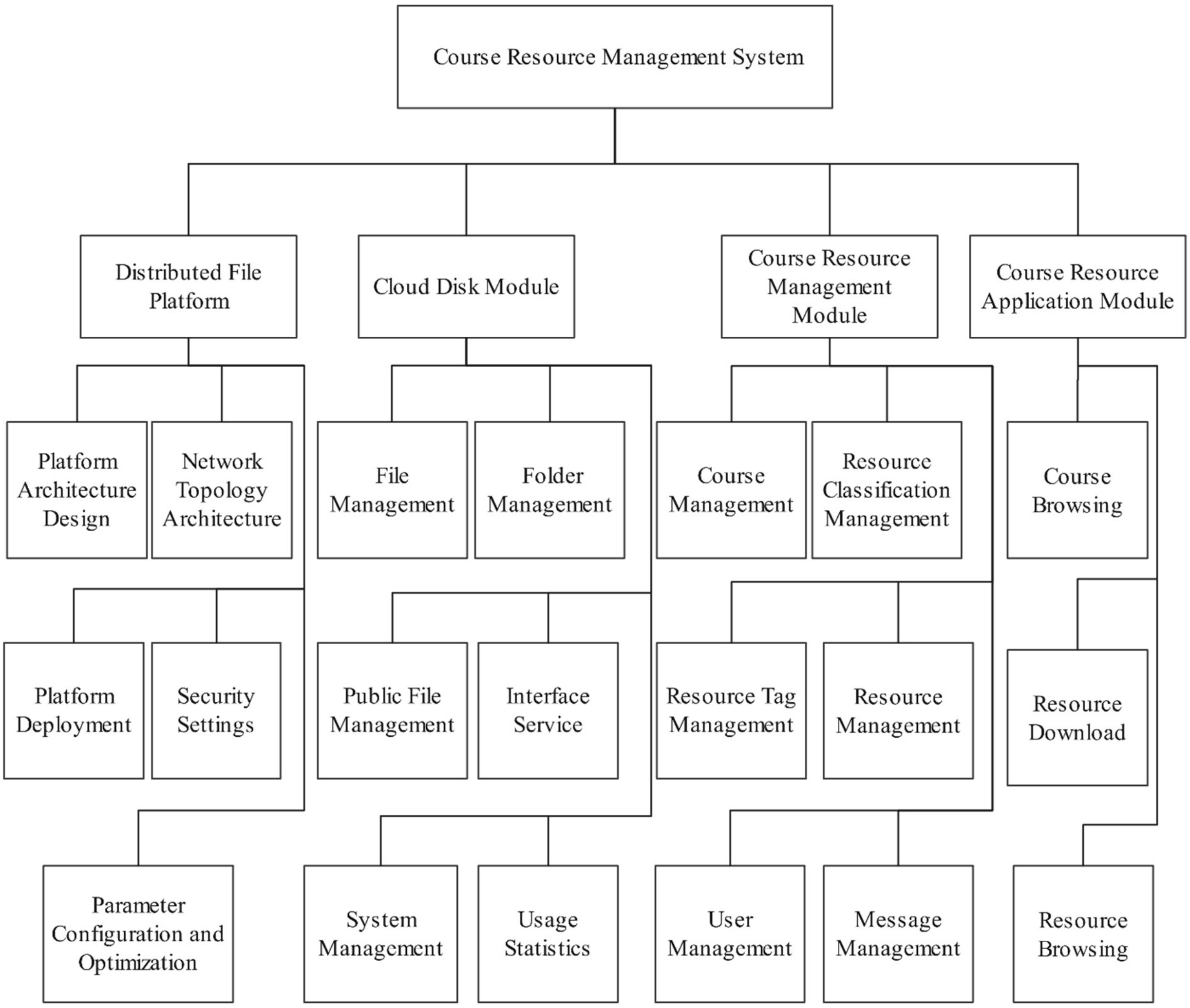

Figure 1. The modules of the system

3.2. Multi-Level Storage Architecture

As shown in Figure 2, the system employs a multi-level data storage structure.

ISSN: 2688-9323

Level 1: Redis Cache: This is the first level of storage, used for caching purposes. It stores information such as user login data, temporary files, and other ephemeral data. Redis is chosen for its high-speed, in-memory data storage capabilities, which enable quick access to frequently used information.

Level 2: MySQL Relational Database: The second level of storage is MySQL, a relational database management system. MySQL stores user data and facilitates rapid data retrieval. It provides structured storage for user-related information, making it easy for users to perform queries and transactions efficiently.

Level 3: Hadoop Distributed Platform: The third level of storage is the Hadoop distributed platform, which serves as a massive-scale data processing and storage system. It supports distributed storage, enabling the handling of large volumes of data, and offers high availability. Hadoop is well-suited for big data processing and provides robust fault tolerance and scalability for the system.

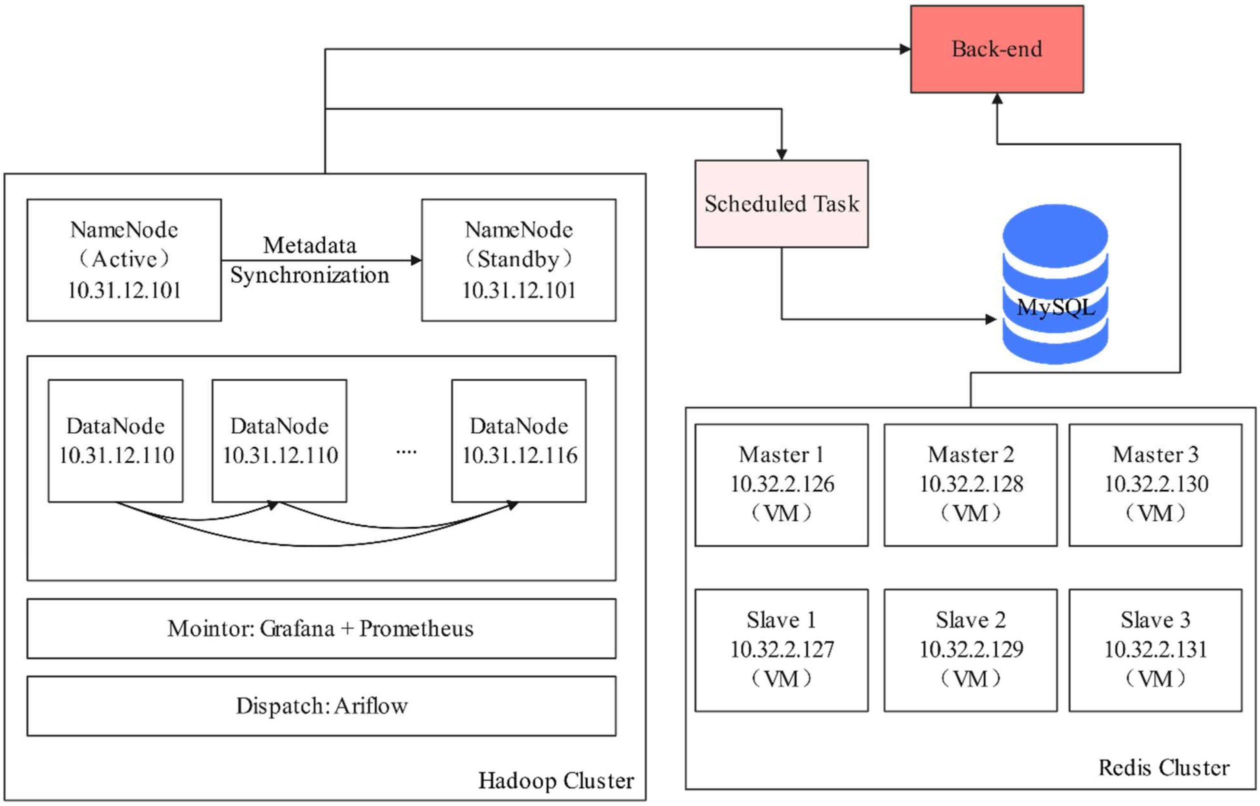

Figure 2. The multi-level data storage architecture

This multi-level storage architecture allows the system to efficiently manage different types of data based on their characteristics and access patterns, ensuring both speed and reliability in handling various data-related tasks.

Redis (Remote Dictionary Server), also known as a remote dictionary service, is an open-source, ANSI C language-written, network-enabled, in-memory, key-value database that can also be used with persistence and provides APIs for various programming languages [5]. The system employs a three-master-three-slave architecture built on virtual machines to create a Redis cluster. The three-master-three-slave configuration is a classic clustering mode in Redis, where three nodes act as masters, forming a highly available cluster, with each master having one slave. This results in a total of six nodes in the cluster.

Apache Hadoop's core modules consist of a storage module and a computation module. The storage module is known as the Hadoop Distributed File System (HDFS), while the computation module is referred to as the MapReduce computation model [6-7]. In the Hadoop architecture, files are initially divided into data blocks and distributed among computing nodes within the cluster. The code responsible for computation tasks is then transmitted to each node, enabling parallel processing of data. This approach effectively utilizes data locality, allowing each node to independently process the data it can access. Compared to traditional supercomputer architectures, this approach results in faster data processing and higher efficiency. In the system, the NameNode is the primary server responsible for managing the file system namespace, while the DataNode are responsible for storing data in blocks. Replication of file blocks is used for data backup to prevent data loss. To enhance the reliability of the file system, six-clustered DataNode are employed to store resource files.

3.3. The Back-end Architecture

The course resource management system employs a front-end and back-end separation architecture. The back-end consists of several layers, including the View, Controller, Service, and Mapper layers. Resource files are stored using the Hadoop cluster. The back-end utilizes the Spring Boot framework, while the front-end employs the Vue.js framework built using the Vue-cli scaffold to facilitate rapid Vue framework development. Vue.js is complemented with the Element UI from the Vue ecosystem, providing a user-friendly interactive experience. The overall architecture is illustrated in Figures 3.

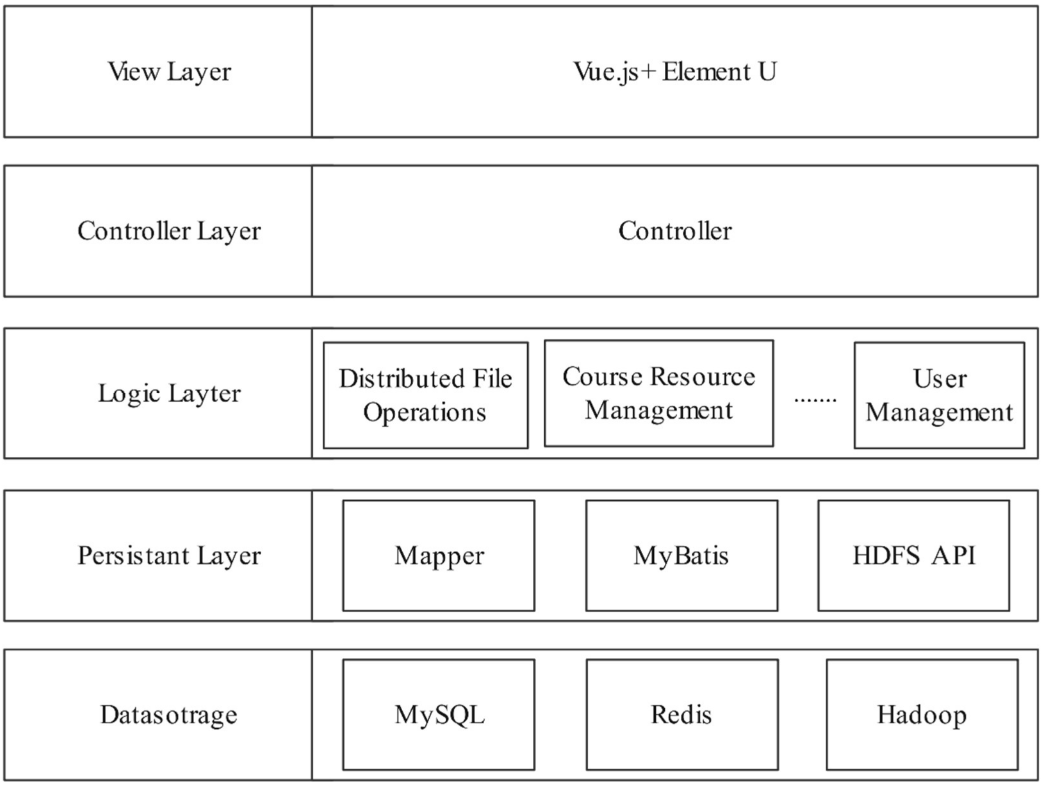

Figure 3. The back-end development architecture

Spring Boot is a Java web development framework that minimizes configuration files, emphasizes convention over configuration, simplifies complexity, and enables the rapid creation of standalone projects [8]. It offers four key features: grouping common dependencies into a single dependency, allowing easy addition to projects; automatic configuration; a

ISSN: 2688-9323

command-line interface for simplifying Spring application development; and Actuator for building application monitors. The specific coding process involves:

Step 1: Creating entity classes.

Step 2: Defining Service interfaces.

Step 3: Configuring XML files.

Step 4: Implementing Mapper interfaces.

Step 5: Creating ServiceImpl implementation classes.

Step 6: Implementing Controller layer to call interfaces.

Step 7: Displaying the final interface.

3.4. The Front-end Architecture

This system primarily follows a responsive system development approach and is designed and implemented as a responsive course resource management system based on Vue.js technology. The website is primarily constructed using Vue.js as the front-end architecture, employing a front-end and back-end separation approach. It leverages Vue.js for modular development, adjusts page layouts using Element UI, creates visualizations using ECharts, and refactors components with Element UI. These modules use concise charts to display changes in data, making it easier for users to understand [9]. The system rapidly builds the Vue front-end framework using the Vue-cli scaffold and manages and extends the front end using NPM. It adds Element UI and ECharts on top of the Vue ecosystem. The front-end structure is shown in Figure 4.

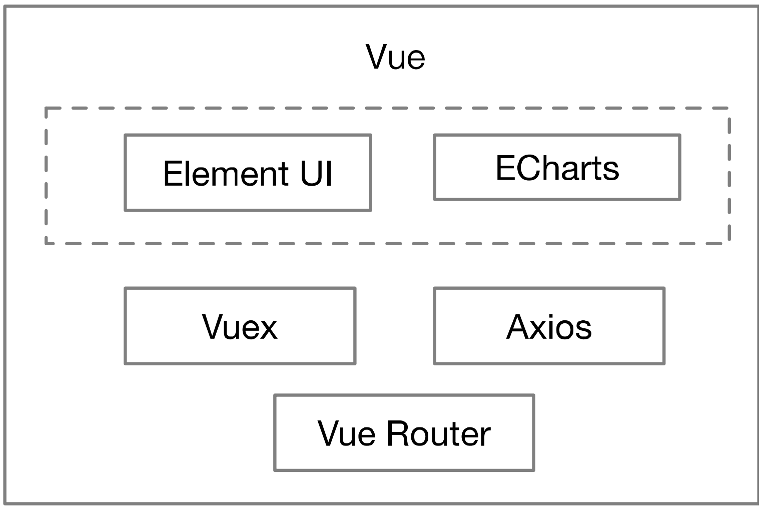

Figure 4. The front-end development architecture

3.5. Multi-Level Security Architecture

In the storage layer, Kerberos authentication is used to ensure that only authorized users can access the Hadoop cluster, including HDFS file access. Transparent Data Encryption (TDE) is used to encrypt files at the file level without requiring application changes [10]. In the service layer, the Spring Security framework is used to implement identity authentication, authorization, and attack defense functions. Spring Security ensures security in back-end services by implementing login, permission management, behavior analysis, and exception detection. The visualization layer ensures the use of the HTTPS encryption protocol during data transmission from the back-end to the front-end. Configuration is in place to prevent cross-site scripting attacks and mitigate the execution of malicious scripts in the user's browser, thus preventing XSS attacks.

4. Conclusion

With the rapid development of digitalization in higher education teaching resources, data has witnessed an explosive growth, revealing the shortcomings and inadequacies of traditional data storage methods gradually. In order to effectively manage educational resources and harness the power of big data and cloud computing, this paper designs and implements a course resource management system based on a private Hadoop cloud. The system constructs a private cloud platform architecture based on Hadoop to achieve efficient retrieval and reliable storage of course resources. By introducing the Hadoop cluster architecture into the private cloud platform, it realizes elastic allocation of data and computational resources. Data replication and backup enhance the redundancy and disaster recovery capabilities of course resources. The system improves access efficiency and reliability through a multi-level storage architecture and ensures security through a multi-level security architecture. The system has been applied in the big data technology major at the Jiangsu Maritime Institute, demonstrating its feasibility.

During the application process, the system still needs further improvement in the following aspects:

Further optimization is required for the storage of distributed resources. Uneven distribution of storage was observed during usage, necessitating adjustments to parameters such as storage blocks and replication numbers.

The user interface requires further refinement. Interface errors were identified when using certain web browsers, necessitating style adjustments to complete the interface upgrade.

Acknowledgments

This work was financially supported by the funding the Philosophy and Social Science Research Project of the Jiangsu Higher Education Institutions of China (2020SJA0730), the 13th Five-Year Plan Project of Education Science in Jiangsu Province(B-a/2018/03/22), the Fundamental Computer Education Research Project (2021-AFCEC-285) of the Association of Fundamental Computer Education Research Project of China, and the Excellent Teaching Team for QingLan Project of the Jiangsu Higher Education Institutions of China (Big Data Technology Teaching Team with Shipping Characteristic).

References

[1] Zhang, Li, and Taizhi Lv. "Research on Cloud Disk System Based on Hadoop." World Scientific Research Journal 8.1 (2022): 187-192.

[2] Siddiqui, Shams Tabrez, et al. "Cloud-based e-learning: using cloud computing platform for an effective e-learning." Smart innovations in communication and computational sciences: proceedings of ICSICCS-2018. Springer Singapore, 2019: 335-346.

[3] Papadakis, Stamatios, et al. "Revolutionizing education: using computer simulation and cloud-based smart technology to facilitate successful open learning." CEUR Workshop Proceedings. Vol. 3358. 2023:1-18.

[4] Judas, Brandon Lee. "Implementation of Cloud Computing Technology for Vocational Higher Schools." International Journal of Information Technology and Education 1.3 (2022): 130-134.

[5] Srai, Aziz, and Fatima Guerouate. "MDA approach for generating the PSM model for the NoSQL key-value database, application on Redis." 2023 3rd International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET). IEEE, 2023: 1-5.

[6] Rajeh, Wahid. "Hadoop distributed file system security challenges and examination of unauthorized access issue." Journal of Information Security 13.2 (2022): 23-42.

ISSN: 2688-9323

[7] Xiang, Lei, et al. "Design and Implementation of Hadoop-Based Campus Cloud Drive." International Conference on Artificial Intelligence and Security. Cham: Springer International Publishing, 2022:331-339.

[8] Lv, Taizhi, Peiyi Tang, and Juan Zhang. "A Real-Time AIS Data Cleaning and Indicator Analysis Algorithm Based on Stream Computing." Scientific Programming 2023 (2023): 1-11.

[9] Bielak, Konrad, Bartłomiej Borek, and Małgorzata Plechawska-Wójcik. "Web application performance analysis using Angular, React and Vue. js frameworks." Journal of Computer Sciences Institute 23 (2022): 77-83.

[10] McGiffen, Matthew. "Introducing Transparent Data Encryption." Pro Encryption in SQL Server 2022: Provide the Highest Level of Protection for Your Data. Berkeley, CA: Apress, 2022. 17-29.