

Received 7 June 2022, accepted 25 June 2022, date of publication 29 June 2022, date of current version 12 July 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3187119

## APPLIED RESEARCH

# Evaluating Erasure Codes in Dicoogle PACS

FABRICIO MARCILLO<sup>1</sup>, ANTONIO F. DIAZ<sup>1</sup>, RAUL H. PALACIOS<sup>2</sup>,  
AND WILMAR HERNANDEZ<sup>3</sup>, (Senior Member, IEEE)

<sup>1</sup>Department of Computer Architecture and Technology/CITIC, University of Granada, 18011 Granada, Spain

<sup>2</sup>Escuela Superior de Huejutla, Autonomous University of the State of Hidalgo, Pachuca, Hidalgo 42039, Mexico

<sup>3</sup>Facultad de Ingeniería y Ciencias Aplicadas, Universidad de Las Américas, Quito 170513, Ecuador

Corresponding author: Wilmar Hernandez (wilmar.hernandez@udla.edu.ec)

This work was supported in part by the Ministry of Science, Innovation and Universities (MICINN) of Spain, under Project PGC2018 098883-B-C44; in part by the European Regional Development Fund (ERDF); in part by the Programa para el Desarrollo Profesional Docente para el Tipo Superior (PRODEP) of Mexico; in part by the Corporación Ecuatoriana para el Desarrollo de la Investigación y la Academia (CEDIA) of Ecuador, under Project CEPRA XII-2018-13; and in part by the Universidad de Las Américas (UDLA), Quito, Ecuador, under Project IEA.WHP.21.02.

**ABSTRACT** DICOM (Digital Imaging and Communication in Medicine) is a standard for image and data transmission in medical purpose hardware and is commonly used for viewing, storing, printing and transmitting images. As a part of the way that DICOM transmits files, the PACS (Picture Archiving and Communication System) platform, Dicoogle, has become one of the most in-demand image processing and viewing platforms. However, the Dicoogle PACS architecture does not guarantee image information recovery in the case of information loss. Therefore, this paper proposes a file recovery solution in the Dicoogle architecture. The proposal consists of maximizing the encoding and decoding performance of medical images through computational parallelism. To validate the proposal, the Java programming language based on the Reed-Solomon algorithm is implemented in different performance tests. The experimental results show that the proposal is optimal in terms of image processing time for the Dicoogle PACS storage system.

**INDEX TERMS** Dicoogle PACS, fault tolerance, DICOM, Reed-Solomon erasure code.

## I. INTRODUCTION

In the medical field, it is necessary to establish norms or standards to share image formats between devices. Specifically, these standards cover aspects of messaging and communication between imaging equipment, such as fixed and portable X-ray equipment, ultrasound (US), single (CT) and multilayer computed tomography (MSCT), magnetic resonance imaging (MRI), positron tomography (PET), single photon emission tomography (SPET) and coronary angiography. In this regard, the National Electrical Manufacturers Association (NEMA) of the United States, along with the American College of Radiology (ACR) have contributed to the development of the norms and standards for medical image processing [1].

Transmission between medical image devices requires standards for reliable communication and storage. For

The associate editor coordinating the review of this manuscript and approving it for publication was Rongbo Zhu<sup>1</sup>.

this purpose, Digital Imaging and Communication in Medicine (DICOM) is a standard protocol for communication among information systems. In addition, DICOM is a medical image storage format that appears to be a solution for interoperability problems between various devices, including medical purpose hardware [2]. To determine the different applications or treatments of DICOM files, an analysis of the literature was carried out. As a result, several lines were found to, in some way, deal with DICOM files. In this sense, [3] and [4] focused on artificial intelligence (AI) and machine learning (ML), respectively. Likewise, [5]–[7] and [8] used deep learning (DL) techniques for to identify and process medical images. Regarding contrastive learning (CL), in [9] and [10], it was applied to medical image segmentation. Furthermore, in [11] and [13], the authors used a set of units called artificial neurons, which are connected and used to transmit signals to each other [13]. Finally, regarding high-quality image reconstruction, in [14], geometric parameters were applied to improve image quality.

Specifically, the essence of this work focuses on high-performance computing (HPC) and fault tolerance by applying the Reed-Solomon erasure code algorithm using encoding and decoding techniques. The reason for this research is that a fault-tolerant storage plugin has been developed as a part of Dicoogle PACS that enables the full recovery of any damaged, corrupted, or lost DICOM file. This plugin can recover the original file without errors and it is a faithful copy of the first one. Therefore, a hash certificate is issued stating that the faithful copy mentioned above is 100% identical to the original file. The originality check of the recovered file is performed through the application of SHA-512 (Secure Hashing Algorithm).

Based on the above, the main objective of the paper is to propose a solution to maximize the performance of the developed storage plugin. This solution is developed by applying parallel computing to the encoding and decoding techniques on DICOM files, considerably reducing the processing times. The DICOM standard integrates the Picture Archiving and Communication System (PACS), which has become the main storage approach for medical images, such as radiology, ultrasound imaging and nuclear medicine [15]. In this regard, the term medical information system arises, which refers to medical image communication and storage as a solution for the image processing technology management system of a hospital environment. An important advantage of PACS is that it substantially reduces both the radiation doses to patients and the operating costs of the radiology department [16].

A serious problem that exists in institutions performing diagnostic imaging (Dx) in the interconnection network or equipment is the blockage caused by the high data flow in the network. This flow is provided by the transfer of DICOM images to the devices that make up a DICOM network called Service Class Users (SCU) and Service Class Providers (SCP). In [17], the transactions of physicians with respect to DICOM archiving within a clinical image management system (CIMS) in support of patient care are described. In addition, the utilization of images in a neurosurgical oncology practice was audited, and there were 400 requests for 233 imaging studies during 297 sessions. Fifty percent were current studies, and 50% were historical studies. Every day, medical imaging modalities generate new data, and the growth in recent years has been exponential. The storage and distribution of the data is managed by PACS, but current technologies face new challenges, such as high availability. However, the massive amount of data requires a huge local infrastructure to handle the load and fault tolerance, which can be costly [18]. The system developed in our research offers features that fully match the DICOM standard in terms of data integrity [19], immutability and fault tolerance [20]. Moreover, an intelligent cache replacement policy suitable for PACS was proposed in [21]. This policy combines the logistic regression (LR) algorithm with the classical least recently used (LRU) cache replacement policy. Furthermore, in [21] the LRLRU algorithm is PACS-oriented and identifies

variables affecting file access probabilities by mining medical data.

The PACS market is heterogeneous, with several PACS vendors having deployed installations in healthcare centers. The DICOM query and retrieval interfaces provided by PACS have multiple variations related to implemented SOP (service object pair) classes, transfer syntaxes, extended negotiations, attributes and match types. These variations can integrate a new DICOM consumer with a PACS complex and make it time consuming. Although most PACS products provide a DICOM conformance statement with a description of their query and retrieval capabilities, there is no collective analysis describing the various query, retrieval capabilities and variations of imaging systems [22]. Additionally, another important aspect concerns system security. With respect to this, from a practical perspective, PACS-specific security measures should be considered along with measures applicable to the Information Technology (IT) infrastructure. The purpose of this is to strengthen the security of PACS systems that are accessible from the internet [23].

In addition, in [24], a computer component was developed to detect renal carcinomas in abdominal tomography images based on the Watersheds Transform. This component supports the diagnosis of renal cell carcinoma (RCC) made by specialists in health centers where the Medical Image Storage, Transmission and Visualization System (XAVIA-PACS) was deployed. Additionally, in [24], an experiment was developed comparing the results with images from an international database (KiTS 2019), and the diagnostic value was analyzed by leading medical specialists.

Perhaps the most difficult task for any new PACS administrator is to understand and gain experience with the DICOM standard. DICOM is a run-length encoded binary standard developed exclusively for medical imaging and may require hexadecimal editors to simply read it [25]. DICOM may be the most important area for a PACS administrator to master, and this mastery increases in difficulty as DICOM standards grow in number and complexity. The most effective way to learn DICOM is through a hands-on approach, which the open-source clinical image and object manager (DCM4CHEE) provide [25]. DCM4CHEE is a collection of open-source applications and utilities for healthcare institutions. These applications have been developed in the Java programming language for performance and portability, which supports implementation in the Java Development Kit (JDK) [25].

At this point, it is worth mentioning that in recent years, the increasing need for durability and efficiency in file storage has made erasure codes (ECs) a desirable new research target. To emphasize the importance of using erasure code algorithms, implementations of these algorithms in distributed file systems are mentioned below. For example, the Reed-Solomon (RS) erasure code ( $n; k$ ) algorithm [26] stands out among the abovementioned algorithms. The RS algorithm performs the processes of encoding given  $n$  original data blocks and decoding to recover lost data from  $k$

redundant blocks. According to the above, it is important to mention that erasure coding-based distributed storage systems are increasingly used by storage vendors for big data. These offer the same reliability as data replication, with a considerable decrease in the amount of storage to maintain data reliability. However, the problem arises when the storage system nodes are distributed over a geographically extensive area [27]. The storage systems with erasure codes are cost-effective and are deployed in large data centers to achieve high reliability with low storage cost [28], [29]. The Reed-Solomon erasure code [26] is part of a well-known family of erasure codes used in Google's ColossusFS [30], Hadoop Distributed File System (HDFS) [29], [31], and other storage systems [32], [33].

To highlight the importance of using erasure code algorithms, implementations of these algorithms in distributed file systems are mentioned below. An Erasure-Coded Data Archival System for Hadoop Clusters (aHDFS) leverages parallel and pipelined encoding processes. The encoding processes enable data acceleration of archival performance in the Hadoop Distributed File System (HDFS) on Hadoop clusters. aHDFS leverages three-way data replicas and MapReduce programming models in the Hadoop cluster to increase performance [34].

Furthermore, in [35], the authors evaluated the performance of the Jerasure [36] and GPU-accelerated Cauchy Reed-Solomon G-CRS [37] erasure code libraries. In the research presented in [38], algorithms combined with erasure codes and the Ant Colony Optimization-based Multiple Data Node Update (ACOUS) scheme were used. Moreover, [39] demonstrated that there is an increased demand for the scalability of the Bitcoin blockchain by applying erasure codes [39]. In addition, Facebook's HDFS-RAID system implements Reed-Solomon encoding and decoding [26] in the format of a distributed Redundant Array of Independent Disks (RAID) file system running on top of HDFS [31]. Moreover, previous versions of HDFS achieved fault tolerance by replicating multiple copies of data, such as RAID-1 on traditional storage arrays. Furthermore, the HDFS-EC (storage system implementing erasure code) system developed and implemented by Cloudera substantially reduces storage overhead while achieving similar or better fault tolerance by using parity cells such as RAID-5 [40].

Having exposed the above implementations and before the introduction to the erasure code algorithms, it is important to mention that HDFS exclusively used 3x replication for fault tolerance, which means that a 1 GB file uses 3 GB of storage space. With EC, the same level of fault tolerance for a 1 GB file can be achieved using only 1.5 GB of storage space. As a result, this feature is expected to considerably change the total cost of ownership for Hadoop usage [41], employing erasure coding to reduce storage cost compared to replication while maintaining high fault tolerance [42]. Moreover, there are technologies that can be exploited to improve performance and resource optimization in a medical imaging environment. For example, the plugin for fault-tolerant

storage uses the Reed-Solomon erasure code [26], [39], [43], [44] open source Backblaze library and the DICOM image files through the Dicoogle PACS system presented in this research.

Dicoogle is a PACS platform to develop new kinds of DICOM repositories, allowing us to make extensions such as storage in different technologies. It offers a full set of DICOM services and a web application to support the search, quick view and management of the PACS archive. Furthermore, some implementations using medical modalities are described in [45]. These implementations can receive patient data and they are related either to the study from the Hospital Information System (HIS) or to the Radiology Information System (RIS). The DICOM modality worklist [2] is the missing electronic link that transfers this critical information between the medical modalities and either to the HIS or the RIS.

For the development of this study, extensive research was carried out by locating the main contributions of Dicoogle PACS related to fault tolerance [20]. The results showed that this problem has not been widely studied, and the absence of studies applied to the Reed-Solomon erasure code was shown. The background described above is the basis for the motivation of this research. Specifically, our motivation is to guarantee the error-free retrieval of medical images in the Dicoogle PACS system in the shortest possible amount of time. In summary, in this research, a fusion of the technologies described above is performed, and for the encoding and decoding process, the Reed-Solomon erasure coding algorithm is used.

The organization of this paper is as follows. Section II presents a survey of the Dicoogle PACS architecture, Section III describes the Reed-Solomon erasure code algorithm, Section IV mentions the configuration of the proposal, Section V discusses the tests and experimental results, and finally, Section VI presents the conclusions.

## II. DICOOGLE PACS

PACS have been widely developed and implemented in hospitals and are now a commodity for healthcare professionals. However, their installation, maintenance and utilization remain challenging due to their heavy structures, which are usually supported by centralized computational solutions [46].

Dicoogle [46] is an open source PACS image storage and communications system; the code is free access and can be downloaded from [47], as developed primarily by UA. PT BIOINFORMATICS [48] and BMD Software [49]. Dicoogle can be used to support three different usage scenarios: production, research, and teaching. Its modular architecture enables the rapid development of new functionalities due to the availability of a Software Development Kit (SDK) [50]. Specifically, the Dicoogle SDK was created to simplify the development of new functionalities of specific use modules that are developed by other researchers. These researchers are known as the Third Party, and the goal is to ensure the

compatibility of new SDK developments with the Dicoogle platform.

In a nutshell, to develop a new functional module, developers must move the package they build to the Dicoogle plugin directory (see Fig. 1). After this process, Dicoogle automatically loads the new modules at startup. Next, the Dicoogle SDK makes all operations related to storage, querying, and indexing immediately available through its internal application programming interface (API) [50]. The overall architecture of Dicoogle (see Fig. 1) replaces the traditional relational database with a more agile process in the indexing and retrieval mechanism. Dicoogle was designed to extract, index, and store all metadata presented in DICOM files from patient studies, including private labels [50].

The paradigm introduced by the Dicoogle project empowers queries over a set of distributed repositories, which are logically indexed as a single agile unit. In [51], the authors described a cloud-based relay service [52] that acts as a communication bridge between different institutions, allowing the community to access, share and discover image records [51]. Dicoogle was proposed in [53] as an architecture for cloud [52]-based PACS archives that provides data privacy, integrity and availability. The solution proposed in [53] is independent of the cloud [52] provider and core modules. Operational metrics for various medical image modalities were tabulated and compared for Google Storage, Amazon S3 and LAN PACS [53]. In [54], the development and validation process of the Dicoogle system in two healthcare centers is described. This system represents a new approach capable of collecting and indexing information from different PACS archives, developed on top of Dicoogle, which allows the construction of multiple views on data repositories according to the DICOM standard in a flexible and fast way. The methodology developed in [54] can contribute to improving the use of DICOM metadata stored in the scattered PACS of radiology departments. If this improvement were not introduced, DICOM metadata would not be used in productivity, efficiency, and quality initiatives in radiology departments [54].

Another implementation based on Dicoogle is concerning [55], where the architecture and implementation of a query-based profiled content-based image retrieval (CBIR) system is proposed. In this solution, CBIR profiles allow the specification of both a distance function and the set of features that must be present for that function to operate. The CBIR retrieval system has been implemented as a mechanism to cope with increasing volumes of information present in medical image repositories [56].

A case study is presented in [57], which describes a process applied to the metadata held by 19 storage units of a PACS of a Portuguese Hospital Center composed of three sites. In total, more than 36 million images were collected in [57], coming from more than ten modalities. The fusion and reconciliation procedure presented in [57] has proven to be robust and efficient. It demonstrates that it is possible to merge metadata collected from multiple PACSs into a single unified

view, preserving the relationship among patients, studies, and images, which is essential for analyses performed on a large scale of medical image repositories from multiple institutions [57].

Recent research [56] describes Dicoogle as a software framework that has enabled developers and researchers to prototype and nimbly implement new functionality, leveraging integrated digital imaging and communications services in DICOM medicine. This complete implementation of a PACS archive is highly extensible due to its plug-in-based architecture and out-of-the-box functionality, thus enabling the exploration of large DICOM datasets and associated metadata. Clearly, these features make the proposed solution very interesting for prototyping, experimentation and bridging functionality with implemented applications [56].

In [58], the software architecture of Dicoogle based on different plugins is presented, which includes a storage plugin with two approaches, one aimed at local storage and the other at storage in the cloud [52]. Additionally, the Dicoogle software architecture is supported by a document-based indexing system and peer-to-peer (P2P) protocols. Furthermore, an architecture of a web pathology PACS fully compatible with DICOM standard data and communication formats is proposed in [59]. The solution includes a PACS archive responsible for storing whole slide image data in DICOM format for whole slide imaging (WSI) and provides a communication interface based on the latest DICOM web services. The other component is a zero-footprint viewer that runs in any web browser. It consumes data using standard PACS archiving web services [59].

In [60], the secure registry mechanism for medical image repositories was designed and instantiated as an extension of a well-known open-source archive. A new web services layer was implemented to provide a solution for DICOM Web protocols for storing, searching and retrieving medical image data.

In [61], the Dicoogle environment is described, with particular emphasis on its learning package, available resources, and the impact of the platform on research and academia. In [61], the authors began by presenting an overview of its architectural concept, the most recent research supported by Dicoogle, some observations obtained from its use in teaching, and statistics on the use of the software worldwide. Additionally, a comparison between the Dicoogle platform and the most popular open source PACS on the market is presented in [61]. Furthermore, current solutions to the problem of medical image storage and management are discussed. Moreover, a solution based on a private network of nodes is presented to overcome problems such as privacy, redundancy, and high availability.

The main problem of data storage in DICOM format is that a radiological examination can have a data volume of more than one gigabyte and consists of thousands of images [62]. In Moldova, in 2015, the DICOM Network project was launched [62], which aims at providing access to radiological studies to medical staff with appropriate access rights, as well



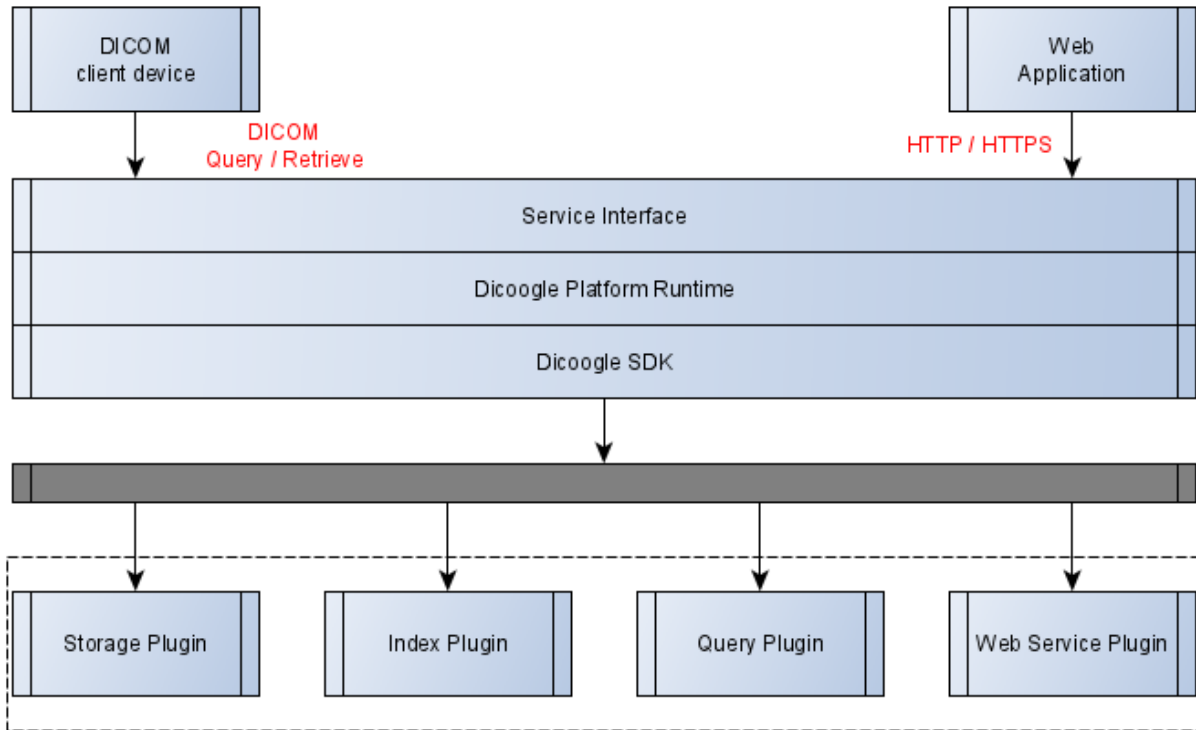


FIGURE 1. Dicoogle PACS architecture.

as giving patients access to their personal radiological examinations. The system now collects and processes more than 500 gigabytes of data each month [62].

In [63], an overview of the current state of the art of non-relational databases was also studied, discussing the strengths and weaknesses of this type of database in our use cases. In addition, the authors of [63] focused on some implementations of this type of database in the medical imaging use case. Furthermore, in [63], these implementations were presented in such a way that they were integrated into the Dicoogle open source PACS.

Finally, in [64], a framework for the integration of multiple microscopy image modalities into the PACS-DICOM universe, including numerous metadata elements, was proposed. In [64], the authors developed a proof-of-concept system for validation purposes that, integrated with the open source PACS Dicoogle, provides image storage, metadata indexing, and visualization.

### III. REED SOLOMON ERASURE CODE

The Reed-Solomon (RS) erasure code [39], [43], [44] is an erasure code algorithm with properties necessary for reliable file storage and retrieval. It is simple to implement while being a reliable technique. This guarantees the recovery of a complete data item, even when part or parts of the original stored data file are lost or unavailable.

In [65], a study of the operation of the encoding and decoding processes of the erasure code algorithm is carried

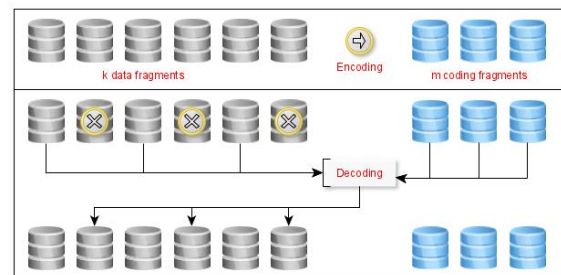


FIGURE 2. An erasure-coded storage system that encodes  $k$  data disks onto  $m$  coding disks.

out. In short, [65] proposed an erasure code storage system that encodes  $k$  data blocks or disks into  $m$  blocks or disks when up to  $m$  data blocks or disks fail. Their contents are decoded from the available blocks or disks to recover the total information (see. Fig. 2).

RS algorithms are used in distributed storage systems and provide efficient implementation and high fault tolerance capability for a given level of redundancy [66]. Storage systems have grown to the point where failures are inevitable, and those who design the systems must plan in advance so that information is not lost when failures occur. The main technology for protecting data from failure is erasure coding, which has a history of more than 50 years [65].

In [67], an analysis of erasure code replication is performed, and they define some situations where full file

replication is preferred. In addition, in [67], the switching point of preferring full file replication or erasure code replication is studied and characterized. Furthermore, considerations in building erasure code replication systems are also discussed in [67].

As mentioned in [68], several versions of RAID technology have been used in recent years aimed at providing file redundancy as well as reliability. Erasure coding is standard practice for systems that reliably store data, and many of them use Reed-Solomon erasure code [39], [43], [44]. Based on the above, centralized and networked storage systems have grown to the point where the fault tolerance provided by RAID-5 is no longer sufficient. In addition, RAID-6 storage systems protect  $k$  data disks with two parity disks so that the  $k + 2$  disk system can tolerate the failure of any two disks [69]. On the other hand, RAID [68] embedded in Linux uses Reed-Solomon erasure code and has a carefully tuned Reed-Solomon implementation in a high-level C language that is part of the RAID module. Additionally, the cloud computing service, Microsoft Azure and the Facebook storage system use Reed-Solomon [70].

In [71] and [72], several proposals for hardware-based EC schemes have been found to take advantage of the advanced computational capabilities of modern data centers. The Backblaze Reed-Solomon erasure code library is an open source [73]. Url's available download [74] has a software license created at the Massachusetts Institute of Technology (MIT) [74], which is a free software license that can be used in free and commercial projects.

**IV. IMPLEMENTATION**

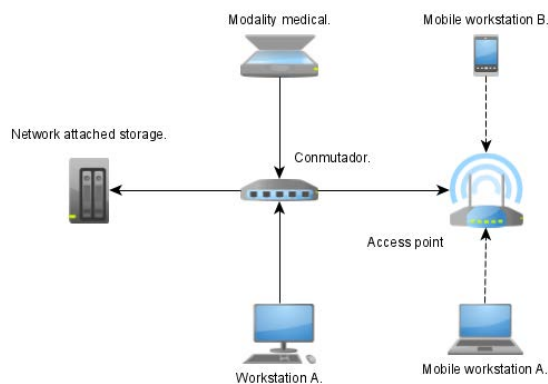
In this section, we describe two steps for the configuration process. First, the hardware configuration for the tests performed is described. Second, the software configuration of the native storage plugin implemented in Dicoogle PACS [60] is described.

In the software case, the Reed-Solomon erasure code algorithm from the Backblaze Library [73] was used, where values for  $k$  and  $m$  are assigned according to the encoding and decoding configuration (see Fig. 3). The following elements are observed: 1) Network Attached Storage (NAS) of the medical images, 2) Modality medical as the image generating element, and 3) Workstation A and Mobile Workstation A and B as devices for visualization of images generated by Modality medical and stored in the NAS.

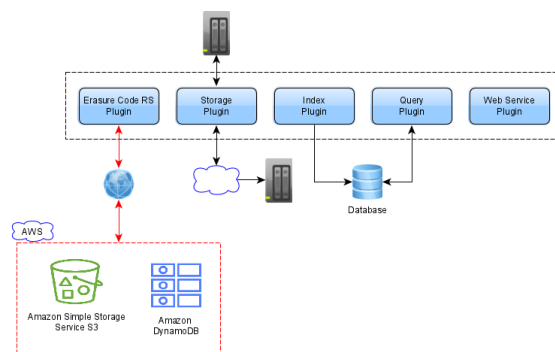
**A. HARDWARE**

The evaluation was performed on an HPE StoreEasy 1660 HPE PN NAS network attached storage system. Q2P71A, with Intel® Xeon-Silver® 4112 CPU @ 2.59 GHz, 128 GB RAM, 5100 PRO M.2 SSD 240 GB x 2 SATA 6 Gb/s HPE PN. 871627-001 and HPE 24TB SAS LFF Smart Carrier 4-pack HDD Bundle HPE PN. Q2P82A. Hard disk rotation speed of 7200 revolutions per minute (RPM).

For the connectivity of the architecture, a Cisco® Catalyst® 2960-XR Gigabit Ethernet Switch was used,



**FIGURE 3. Architecture used to validate the proposal.**



**FIGURE 4. Erasure Code Plugin in Dicoogle PACS.**

and for the wireless connectivity, a Cisco Aironet AIR-CAP1702I-A-K9 Access Point was used, applying the IEEE 802.11n protocol at the 2.4 GHz frequency.

On the other hand, tests were performed with DICOM files using FDR Smart X [75], which is the new Fujifilm X-ray system. Likewise, in this research, a Toshiba Tomographer model TSX-031A, ActivionTM16, was used. This equipment uses a 16-slice helical computed tomography (CT) system that supports the exploration of the entire human body [76]. The system generates a minimum of 32 slices per 1.5 seconds using the selectable slice thickness multiwire detector (SSMD). Regarding the architecture shown in Fig. 3, Workstation A and Mobile workstation A and B correspond to devices for displaying DICOM images.

**B. SOFTWARE**

The main proposal in this work lies in the implementation of a plugin for erasure code through the Reed-Solomon erasure code algorithm of the Backblaze library. In comparison with the Dicoogle PACS architecture [58], Fig. 4 shows the location of the plugin implemented in the proposal. Once the Backblaze Reed-Solomon erasure code library was evaluated, a native plugin was implemented in the Dicoogle PACS storage layer to guarantee fault tolerance. The main scheme of the algorithm is shown in Fig. 5. Java language was used for the development, and threads were used to improve

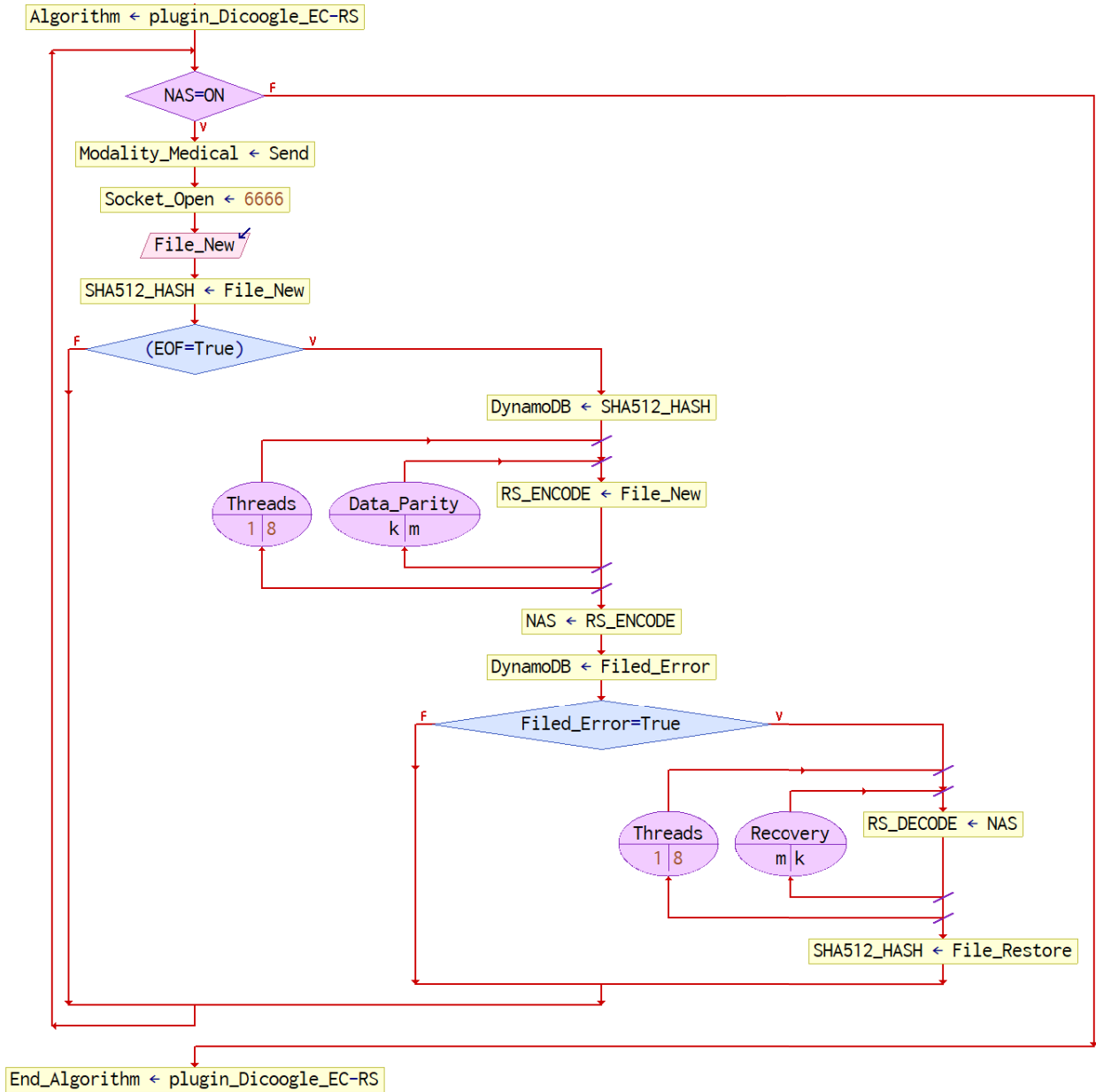


FIGURE 5. Erasure Code Plugin algorithm.

the performance of the algorithm. The characteristics of the development are mentioned below:

- The extension of the files used in the storage plugin was *.dcm*, which corresponds to the DICOM medical image format. These files are created by using the Fujifilm FDR Smart X-ray devices and the Toshiba Tomograph model TSX-031A.
- Regarding services and add-ons, it is possible to start and/or stop running services in real time. For this reason, a port is defined for use in Dicoogle PACS.
- For each DICOM file, a SHA512 hash function is used. This is done to perform SHA512 hash verification to avoid duplicity of information through a data structure.

- The SHA512 hash is stored in a NoSQL DynamoDB database. The implementation of DynamoDB in the storage plugin was provided as a java executable.jar file. The proposal can run on different operating systems, such as Windows, Linux, macOS and other Java compatible platforms.
- The protection provided by erasure coding can be represented in simple form by the following equation:  $n = k + m$ , where  $k$  is the original amount of data and  $m$  represents the redundant data added to provide protection against failures through erasure code.
- Backblaze Reed-Solomon is an erasure coding library for calculating fault tolerance and then using it to

TABLE 1. File size used in the test environment to Encode  $k = 4, m = 2$ .

Original size	Fragment size	Redundancy size
2 MB	512 KB	3 MB
4 MB	1 MB	6 MB
8 MB	2 MB	12 MB
16 MB	4 MB	24 MB
32 MB	8 MB	48 MB
64 MB	16 MB	96 MB
128 MB	32 MB	192 MB
256 MB	64 MB	384 MB
512 MB	128 MB	768 MB
1024 MB	256 MB	1536 MB

reconstruct files. When a DICOM file has been stored in Dicoogle PACS, the storage plugin can split it into 4, 6 and 10 fragments of the same size for each DICOM file. This process is called encode. Subsequently, 2, 3 and 5 additional fragments are created, which maintain parity. This results in a total of 6, 9 and 15 fragments for each configuration (i.e., 4, 6, 10 fragments). With this configuration, it is possible to reconstruct the original file from 4, 6 and 10 fragments. This process is called the decode of the 6, 9 and 15 fragments.

V. RESULTS

As part of the proposal put forward in this research, the built environment required the configuration of the Reed-Solomon erasure code storage plugin implemented in Dicoogle PACS. In this sense, two types of experiments were carried out using Encode and Decode. In the encoder process, settings regarding the original data block size  $k$  and the parity block size  $m$  were used for the recovery of any lost blocks. For the Decode process, data recovery is performed from the  $m$  parity blocks to ensure reliability. The main results for both processes are presented in Figs. 6, 7, 8, 9, 10, and 11. In all cases, the  $x$ -axis represents the data block size in MegaBytes (MB), and the  $y$ -axis represents the time consumption in milliseconds for each block size. To validate the recovery of lost blocks, we induced failures in the configuration to obtain the times and resource consumption (memory and processor) in the transmission/reception of DICOM images. Likewise, in all configurations from 1 to 8, threads are used. Having explained the above, different tests with respective configurations are presented below:

**Test 1.-** In this test, DICOM files of various sizes (see. Table 1) containing part of the lower extremity of the human body were used. The results of this configuration are shown in Fig. 6. In this figure, the time consumed by each strand in a ratio of 4 data blocks and 2 parity blocks (i.e., RS (4,2)) for complete data recovery in the case of loss of 1 or 2 data blocks can be observed. In the case of encryption, a consumption of approximately 0.9 seconds was observed using 8 threads with the most significant group of files.

**Test 2.-** In Fig. 7, reference is made to the 2-fault tolerance results of the DICOM file group of Test 1. In this figure, it is observed that the Decode process consumes approximately

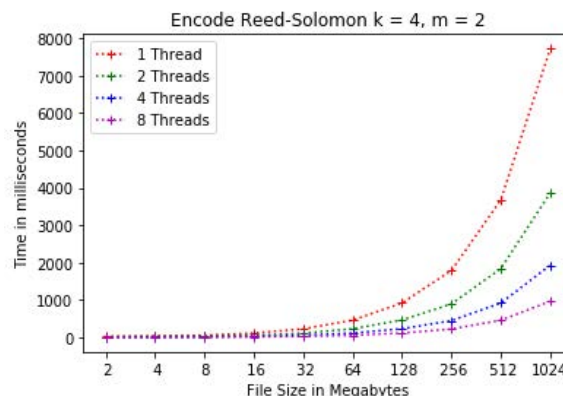


FIGURE 6. Encode configuration, RS(4, 2).

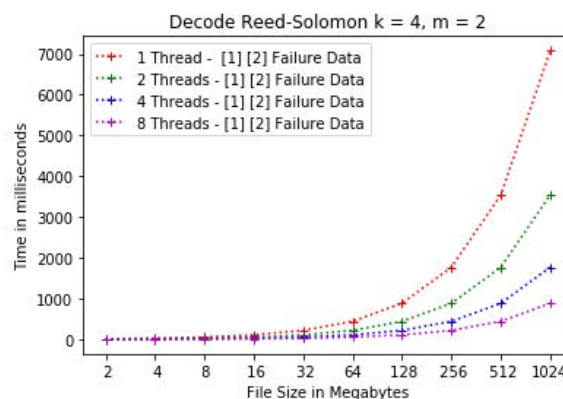


FIGURE 7. Decode configuration, RS(4, 2).

TABLE 2. File size used in the test environment to Encode  $k = 6, m = 3$ .

Original size	Fragment size	Redundancy size
2 MB	341.33 KB	3 MB
4 MB	682.00 KB	6 MB
8 MB	1.33 MB	12 MB
16 MB	2.66 MB	24 MB
32 MB	5.33 MB	48 MB
64 MB	10.66 MB	96 MB
128 MB	21.33 MB	192 MB
256 MB	42.66 MB	384 MB
512 MB	85.33 MB	768 MB
1024 MB	170.66 MB	1536 MB

0.8 seconds when 8 threads are used for the recovery of 1 or 2 data blocks.

**Test 3.-** In this test, a second set of DICOM files was used (see. Table 2). Figure 8 shows the time consumed by the RS (6,3) configuration as a function of the number of threads for the encoder. In this case, it is observed that the time consumption is close to 1.5 seconds using 8 threads for the original maximum data block size of 1024 MB.

**Test 4.-** In Fig. 9, reference is made to the results of tolerance of up to 3 failures of the group of DICOM files of test 3. It is observed that the Decode process consumes approximately 1.2 seconds when 8 threads are used for the recovery of 1, 2 and 3 data blocks.



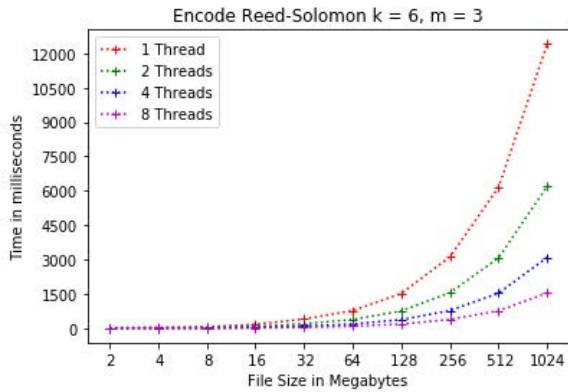


FIGURE 8. Encode configuration, RS(6, 3).

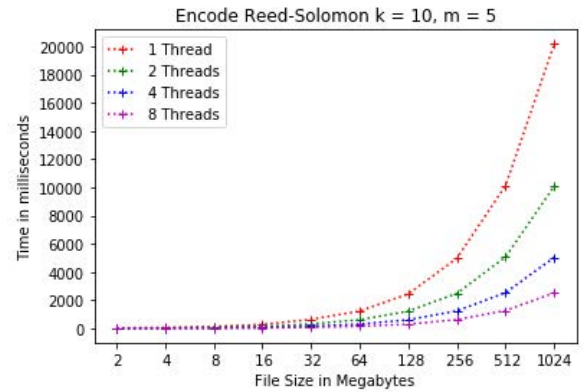


FIGURE 10. Encode configuration, RS(10, 5).

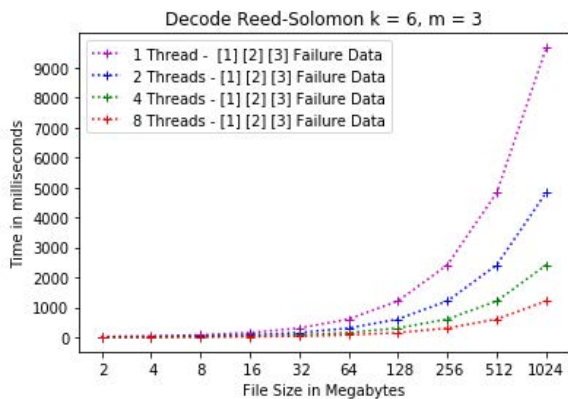


FIGURE 9. Decode configuration, RS(6, 3).

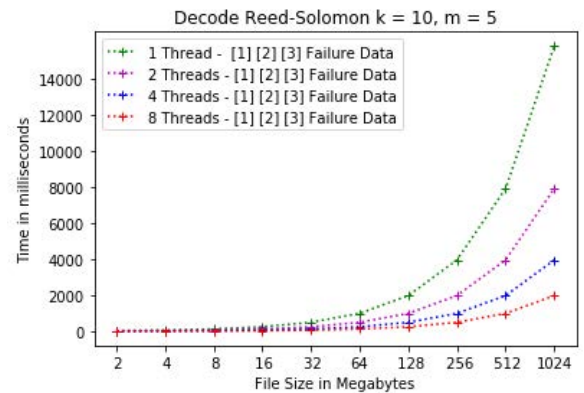


FIGURE 11. Decode configuration, RS(10, 5).

TABLE 3. File size used in the test environment to Encode  $k = 10, m = 5$ .

Original size	Fragment size	Redundancy size
2 MB	204.80 KB	3 MB
4 MB	409.60 KB	6 MB
8 MB	819.20 KB	12 MB
16 MB	1.60 MB	24 MB
32 MB	3.20 MB	48 MB
64 MB	6.40 MB	96 MB
128 MB	12.80 MB	192 MB
256 MB	25.60 MB	384 MB
512 MB	51.20 MB	768 MB
1024 MB	102.40 MB	1536 MB

**Test 5.-** In this test, a third set of DICOM files was used (see Table 3). Figure 10 shows the time consumed by the RS configuration (10,5) as a function of the number of threads for Encode. This configuration allows loss of 1 and 5 data blocks of the complete DICOM file. In this case, it is observed that the time consumption is close to 2.5 seconds using 8 threads for the maximum block size of 1024 MB.

**Test 6.-** Finally, Fig. 11 shows the results of tolerance from 1 to 5 failures of the group of DICOM files of Test 5. It is observed that the Decode process consumes approximately 1.9 seconds when 8 threads are used for the recovery of up to 5 data blocks.

Before ending this section, it is important to highlight the research results. In this sense, first, the approaches presented in **Test. 1** are taken as a reference. These allow us to reduce the data processing time for  $k = 4$  and the redundancy  $m = 2$ , reaching reductions in both encoding of 0.9 seconds and decoding of 0.8 seconds. This shows that the research results of the developed storage plugin are optimal in terms of processing time. Moreover, the use of different threads with the same characteristics has been established for all cases (see **Tests. 1-6**, and Figs. 6, 7, 8, 9, 10 and 11)  $k = 4, 6, 10$  and  $m = 2, 3, 5$ , revealing a storage efficiency of 66.66 %.

Second, the results of this research show that the Reed-Solomon erasure code algorithm can be successfully applied to Dicoogle PACS. In other research, the applications of Reed-Solomon erasure code are for environments other than the one in this paper, showing a gap in its applications to Dicoogle PACS. In this regard, it is worth mentioning that the study carried out in [77] used values of  $k = 2, 4, 8$  and  $m = 2, 2, 2$ . Likewise, in [78], the authors used  $k = 6, 10$  and  $m = 3, 4$ . Moreover, in [79], the authors used the following values: 1)  $k = 8, 9, 10$  and  $m = 2, 2$ , 2)  $k = 8, 9, 10$  and  $m = 3$ , and 3)  $k = 8, 9, 10$  and  $m = 4$ . Finally, in [80], the authors used  $k = 12$  and  $m = 4$ . Until now, the use of the  $k = 10$  and  $m = 5$  techniques had not been considered. However, this paper does introduce its use to establish equal storage efficiency conditions for the developed plugin.

## VI. CONCLUSION

The development of the research is based on parallelization through the use of threads in the execution of encoding and decoding. It is observed that the use of threads reduces the execution time in the encoding and decoding processes. In addition, the implementation of the Reed-Solomon algorithm in the different processing configurations in the Dicoogle PACS storage plugin is performed.

The main contribution of this research is that a considerable improvement in fault tolerance is achieved through erasure code algorithms in the Dicoogle PACS medical image storage system. This means supporting a fault tolerance configuration of  $m = 2, 3, 5$ , which is superior to the proposal made in the native FlexProtect technology of the OneFS operating system [81]. Simply put, the OneFS operating system uses Reed-Solomon coding to provide redundancy and data availability. FlexProtect technology supports data protection on up to four simultaneous failures of entire nodes or individual units of the configuration. This means that FlexProtect is more efficient than the RAID-6 commonly used today when the aim is to provide data availability [82]. This comparison is made in terms of the maximum level of RAID-6.

Furthermore, in the proposal presented in this paper, a hash function defined in [83] has been incorporated into the storage plugin, which is a method to generate keys to univocally represent a DICOM file or a set of data of a DICOM file. In this case, this process operates mathematically as a SHA-512 compression function that operates on a dataset of any length, and as a result, generates a digital key of a fixed size that is independent of the size of the original DICOM file.

As future research work, the current proposal inspires us to explore other implementations using dedicated graphics processing technologies such as GPUs [84] to take advantage of the GPUDirect technology supported by the Nvidia GPU [85].

## REFERENCES

- [1] O. S. Pinykh, "Digital imaging and communications in medicine (DICOM): A practical introduction and survival guide," *J. Nucl. Med.*, vol. 50, p. 1384, Aug. 2009.
- [2] L. R. Alvarez and R. Vargas Solis, "DICOM RIS/PACS telemedicine network implementation using free open source software," *IEEE Latin Amer. Trans.*, vol. 11, no. 1, pp. 168–171, Feb. 2013, doi: 10.1109/TLA.2013.6502797.
- [3] C. You, Y. Yang, S. Hongming, G. Lars, L. Guang, J. Shenghong, Z. Zhuyang, Z. Zhen, Z. Yi, C. Wenxiang, and W. Ge, "Structurally-sensitive multi-scale deep neural network for low-dose CT denoising," *IEEE Access*, vol. 6, pp. 41839–41855, 2018.
- [4] C. You, R. Zhao, F. Liu, S. Dong, S. Chinchali, U. Topcu, L. Staib, and J. S. Duncan, "Class-aware generative adversarial transformers for medical image segmentation," 2022, *arXiv:2201.10737*.
- [5] C. You, W. Cong, M. W. Vannier, P. K. Saha, E. A. Hoffman, G. Wang, G. Li, Y. Zhang, X. Zhang, H. Shan, M. Li, S. Ju, Z. Zhao, and Z. Zhang, "CT super-resolution GAN constrained by the identical, residual, and cycle learning ensemble (GAN-CIRCLE)," *IEEE Trans. Med. Imag.*, vol. 39, no. 1, pp. 188–203, Jan. 2020, doi: 10.1109/TMI.2019.2922960.
- [6] Q. Lyu, C. You, H. Shan, and G. Wang, "Super-resolution MRI through deep learning," 2018, *arXiv:1810.06776*.
- [7] L. Yang, R. P. Ghosh, J. M. Franklin, S. Chen, C. You, R. R. Narayan, M. L. Melcher, and J. T. Liphardt, "NuSeT: A deep learning tool for reliably separating and analyzing crowded cells," *PLOS Comput. Biol.*, vol. 16, no. 9, Sep. 2020, Art. no. e1008193.
- [8] I. Guha, S. A. Nadeem, C. You, X. Zhang, S. M. Levy, G. Wang, J. C. Torner, and P. K. Saha, "Deep learning based high-resolution reconstruction of trabecular bone microstructures from low-resolution CT scans using GAN-CIRCLE," *Proc. SPIE*, vol. 11317, Feb. 2020, Art. no. 113170U, doi: 10.1117/12.2549318.
- [9] C. You, R. Zhao, L. Staib, and J. S. Duncan, "Momentum contrastive voxel-wise representation learning for semi-supervised volumetric medical image segmentation," 2021, *arXiv:2105.07059*.
- [10] C. You, Y. Zhou, R. Zhao, L. Staib, and J. S. Duncan, "SimCVD: Simple contrastive voxel-wise representation distillation for semi-supervised medical image segmentation," *IEEE Trans. Med. Imag.*, early access, Mar. 23, 2022, doi: 10.1109/TMI.2022.3161829.
- [11] L. Qing, Y. Chenyu, S. Hongming, Z. Yi, and W. Ge, "Super-resolution MRI and CT through GAN-CIRCLE," *Proc. SPIE*, vol. 11113, Sep. 2019, Art. no. 111130X, doi: 10.1117/12.2530592.
- [12] Y. Chenyu, Y. Linfeng, Z. Yi, and W. Ge, "Low-dose CT via deep CNN with skip connection and network-in-network," *Proc. SPIE*, vol. 11113, Sep. 2019, Art. no. 111131W, doi: 10.1117/12.2534960.
- [13] Y. Chenyu, J. Yang, C. Julius, and D. S. James, "Unsupervised Wasserstein distance guided domain adaptation for 3d multi-domain liver segmentation," in *Interpretable and Annotation-Efficient Learning for Medical Image Computing*. Cham, Switzerland: Springer, 2020, pp. 155–163, doi: 10.1007/978-3-030-61166-8.
- [14] G. Li, L. Shouhua, Y. Chenyu, G. Matthew, Z. Liang, W. Ge, and G. Ning, "A novel calibration method incorporating nonlinear optimization and ball-bearing markers for cone-beam CT with a parameterized trajectory," *Med. Phys.*, vol. 46, no. 1, pp. 152–164, 2019.
- [15] L. Yan, Y. Yu, and H. Ma, "DICOM standard and its application in PACS system," *Med. Imag. Process Technol.*, vol. 1, no. 1, pp. 34–41, Feb. 2018, doi: 10.24294/mipt.v1i1.221.
- [16] M. Modrák and V. Modrák, "The effect of a PACS on patient radiation doses and operating costs in a radiology department: A practical study," *Proc. Technol.*, vol. 9, pp. 1282–1287, Jan. 2013, doi: 10.1016/j.protecy.2013.12.143.
- [17] B. L. Guthrie, C. Price, J. Zaleski, and E. Backensto, "The digital imaging and communications in medicine (DICOM) archive is a dynamic component of a clinician image-related workflow solution," *J. Digit. Imag.*, vol. 14, no. S1, pp. 190–193, Jun. 2001.
- [18] R. Lebre, L. B. Silva, and C. Costa, "Decentralizing the storage of a DICOM compliant PACS," in *Proc. IEEE Int. Conf. Bioinf. Biomed. (BIBM)*, Dec. 2021, pp. 2749–2756, doi: 10.1109/BIBM52615.2021.9669902.
- [19] L. Yao, J. Lu, J. Liu, D. Wang, and B. Meng, "A secure and efficient distributed storage scheme SAONT-RS based on an improved AONT and erasure coding," *IEEE Access*, vol. 6, pp. 55126–55138, 2018.
- [20] W. Pu, N. Chen, and Q. Zhong, "SDCUP: Software-defined-control based erasure-coded collaborative data update mechanism," *IEEE Access*, vol. 8, pp. 180646–180660, 2020.
- [21] Y. Wang, Y. Yang, C. Han, L. Ye, Y. Ke, and Q. Wang, "LR-LRU: A PACS-oriented intelligent cache replacement policy," *IEEE Access*, vol. 7, pp. 58073–58084, 2019.
- [22] A. Boufahja, S. Nichols, and V. Pangon, "Quantitative evaluation of PACS query/retrieve capabilities," *J. Digit. Imag.*, vol. 34, no. 5, pp. 1302–1315, Sep. 2021.
- [23] M. Eichelberg, K. Kleber, and M. Kämmerer, "Cybersecurity in PACS and medical imaging: An overview," *J. Digit. Imag.*, vol. 33, no. 6, pp. 1527–1542, Oct. 2020.
- [24] A. Orellana Garcia, L. M. Garcia Portal, Y. Feijoo Ortiz, G. Ceruto Marrero, and E. E. Cardero Alvarez, "Component for renal carcinoma detection in abdominal tomography applying watersheds transform in the XAVIA PACS system," *IEEE Latin Amer. Trans.*, vol. 19, no. 4, pp. 542–550, Apr. 2021, doi: 10.1109/TLA.2021.9448536.
- [25] M. Warnock, J. Toland, C. Evans, D. Wallace, and P. Nagy, "Benefits of using the DCM4CHE DICOM archive," *J. Digit. Imag.*, vol. 20, no. 1, pp. 125–129, Oct. 2007.
- [26] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *J. Soc. Ind. Appl. Math.*, vol. 8, no. 2, pp. 300–304, Jun. 1960.
- [27] L. J. Mohan, P. I. S. Caneleo, U. Parampalli, and A. Harwood, "Geo-aware erasure coding for high-performance erasure-coded storage clusters," *Ann. Telecommun.*, vol. 73, nos. 1–2, pp. 139–152, Jan. 2018.

- [28] C. Huang, H. Simitci, Y. Xu, A. Ogus, B. Calder, P. Gopalan, J. Li, and S. Yekhanin, "Erasure coding in windows azure storage," in *Proc. USENIX Annu. Tech. Conf. (USENIX ATC)*, Boston, MA, USA, 2012, pp. 15–26.
- [29] A. Thusoo, Z. Shao, S. Anthony, D. Borthakur, N. Jain, J. Sen Sarma, R. Murthy, and H. Liu, "Data warehousing and analytics infrastructure at Facebook," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, Indianapolis, IN, USA, Jun. 2010, pp. 459–470.
- [30] E. Ma. (2012). *Colossus: Successor to the Google File System (GFS)*. SysTutorials. Accessed: Apr. 10, 2022. [Online]. Available: <https://www.systutorials.com/colossus-successor-to-google-file-system-gfs/>
- [31] M. Sathiamoorthy, M. Asteris, D. Papailiopoulos, A. G. Dimakis, R. Vadali, S. Chen, and D. Borthakur, "XORing elephants: Novel erasure codes for big data," 2013, *arXiv:1301.3791*.
- [32] K. Rashmi, N. Shah, G. Dikang, K. Hairong, D. Borthakur, and K. Ramchandran, "A 'hitchhiker's' guide to fast and efficient data reconstruction in erasure-coded data centers," in *Proc. ACM Conf. SIGCOMM*, 2014, pp. 331–342, doi: [10.1145/2619239.2626325](https://doi.org/10.1145/2619239.2626325).
- [33] F. Schmuck and R. Haskin, "GPFS: A shared-disk file system for large computing clusters," in *Proc. 1st USENIX Conf. File Storage Technol.*, Berkeley, CA, USA, 2002, p. 19.
- [34] Y. Chen, Y. Zhou, S. Taneja, X. Qin, and J. Huang, "AHDFS: An erasure-coded data archival system for Hadoop clusters," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 11, pp. 3060–3073, Nov. 2017.
- [35] L. Liang, H. He, J. Zhao, C. Liu, Q. Luo, and X. Chu, "An erasure-coded storage system for edge computing," *IEEE Access*, vol. 8, pp. 96271–96283, 2020.
- [36] J. S. Plank, S. Simmerman, and C. D. Schuman, "Jerasure: A library in C/C++ facilitating erasure coding for storage applications-version 1.2," Univ. Tennessee, Knoxville, TN, USA, Tech. Rep. CS-08-627, 2008. Accessed: Apr. 10, 2022. [Online]. Available: <http://wiki.osll.ru/lib/exe/fetch.php/plank-paper.pdf>
- [37] C. Liu, Q. Wang, X. Chu, and Y.-W. Leung, "G-CRS: GPU accelerated Cauchy Reed–Solomon coding," *IEEE Trans. Parallel Distrib. Syst.*, vol. 29, no. 7, pp. 1484–1498, Jul. 2018.
- [38] Y. Hu, Q. Li, W. Xie, and Z. Ye, "An ant colony optimization based data update scheme for distributed erasure-coded storage systems," *IEEE Access*, vol. 8, pp. 118696–118706, 2020.
- [39] M. Jin, X. Chen, and S.-J. Lin, "Reducing the bandwidth of block propagation in bitcoin network with erasure coding," *IEEE Access*, vol. 7, pp. 175606–175613, 2019.
- [40] A. Chiniah and A. Mungur, "Dynamic erasure coding policy allocation (DECPA) in Hadoop 3.0," in *Proc. 6th IEEE Int. Conf. Cyber Secur. Cloud Comput. (CSCloud)/5th IEEE Int. Conf. Edge Comput. Scalable Cloud (EdgeCom)*, Jun. 2019, pp. 29–33, doi: [10.1109/CSCloud/EdgeCom.2019.00015](https://doi.org/10.1109/CSCloud/EdgeCom.2019.00015).
- [41] K. Nansai, X. Chen, S. Chen, and J. Zhang. (2019). *HDFS Erasure Coding in Production*. CLOUDERA Blog. Accessed: Apr. 10, 2022. [Online]. Available: <https://blog.cloudera.com/hdfs-erasure-coding-in-production/>
- [42] M. Xia, M. University, M. Saxena, M. Blaum, and A. David, "A tale of two erasure codes in HDFS," in *Proc. 13th USENIX Conf. File Storage Technol. (FAST)*, Santa Clara, CA, USA, 2015, pp. 213–226.
- [43] R. Elagooz, A. Mahran, S. Gasser, and M. Aboul-Dahab, "Efficient low-complexity decoding of CCSDS Reed–Solomon codes based on Justesen's concatenation," *IEEE Access*, vol. 7, pp. 49596–49603, 2019.
- [44] P. Liu, Z. Pan, and J. Lei, "Parameter identification of Reed–Solomon codes based on probability statistics and Galois field Fourier transform," *IEEE Access*, vol. 7, pp. 33619–33630, 2019.
- [45] M. E. Gale and D. R. Gale, "DICOM modality worklist: An essential component in a PACS environment," *J. Digit. Imag.*, vol. 13, no. 3, pp. 101–108, Aug. 2000.
- [46] C. Costa, C. Ferreira, C. Bastião, L. Ribeiro, A. Silva, and J. L. Oliveira, "Dicoogle—an open source peer-to-peer PACS," *J. Digit. Imag.*, vol. 24, no. 5, pp. 848–856, Oct. 2010.
- [47] *Dicoogle PACS Source Code*. Accessed: May 15, 2022. [Online]. Available: <https://dicoogle.com/downloads>
- [48] Universidade de Aveiro, Portugal. *UA. PT Bioinformatics*. A Computational Biology and Biomedical Informatics Research Group. Accessed: May 15, 2022. [Online]. Available: <https://bioinformatics.ua.pt/>
- [49] BMD, Software. Open Source Projects. *Strong Collaboration With Universities, R&D Institutes and Other Companies to Improve the Processes and Change Healthcare*. Accessed: May 15, 2022. [Online]. Available: <https://www.bmd-software.com/research/>
- [50] R. Lebre R, L. Bastião, and C. Costa, "Shared medical imaging repositories," *Stud. Health Technol. Inform.*, vol. 247, pp. 411–415, 2018, doi: [10.3233/978-1-61499-852-5-411](https://doi.org/10.3233/978-1-61499-852-5-411).
- [51] C. Viana-Ferreira, C. Costa, and J. L. Oliveira, "Dicoogle relay—A cloud communications bridge for medical imaging," in *Proc. 25th IEEE Int. Symp. Computer-Based Med. Syst. (CBMS)*, Jun. 2012, pp. 1–6, doi: [10.1109/CBMS.2012.6266402](https://doi.org/10.1109/CBMS.2012.6266402).
- [52] Y. Xiao, S. Zhou, and L. Zhong, "Erasure coding-oriented data update for cloud storage: A survey," *IEEE Access*, vol. 8, pp. 227982–227998, 2020.
- [53] L. A. B. Silva, C. Costa, and J. L. Oliveira, "A PACS archive architecture supported on cloud services," *Int. J. Comput. Assist. Radiol. Surg.*, vol. 7, no. 3, pp. 349–358, Jun. 2011.
- [54] M. Santos, S. de Francesco, L. Silva, A. Silva, C. Costa, and N. Rocha, "Multi vendor DICOM metadata access a multi site hospital approach using Dicoogle," in *Proc. 8th Iberian Conf. Inf. Syst. Technol. (CISTI)*, Jun. 2013, pp. 1–7.
- [55] F. Valente, C. Costa, and A. Silva, "Dicoogle, a PACS featuring profiled content based image retrieval," *PLoS ONE*, vol. 8, no. 5, May 2013, Art. no. e61888.
- [56] F. Valente, L. Silva, T. Godinho, and C. Costa, "Anatomy of an extensible open source PACS," *J. Digit. Imag.*, vol. 29, no. 3, pp. 284–296, Oct. 2016.
- [57] M. Santos, J. Pavão, T. Godinho, and N. Pacheco, "DICOM metadata aggregation from multiple healthcare facilities," in *Proc. IEEE 5th Portuguese Meeting Bioeng. (ENBENG)*, Feb. 2017, pp. 1–4, doi: [10.1109/ENBENG.2017.7889472](https://doi.org/10.1109/ENBENG.2017.7889472).
- [58] E. Pinho, T. F. Godinho Valente, and C. Costa, "A multimodal search engine for medical imaging studies," *J. Digit. Imag.*, vol. 30, no. 1, pp. 39–49, Aug. 2016.
- [59] T. Marques Godinho, R. Lebre, L. B. Silva, and C. Costa, "An efficient architecture to support digital pathology in standard medical imaging repositories," *J. Biomed. Informat.*, vol. 71, pp. 190–197, Jul. 2017.
- [60] R. Lebre, L. Bastião, and C. Costa, "A cloud-ready architecture for shared medical imaging repository," 2019, *arXiv:1904.05795*.
- [61] R. Lebre, E. Pinho, J. Silva, and C. Costa, "Dicoogle framework for medical imaging teaching and research," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jul. 2020, pp. 1–7, doi: [10.1109/ISCC50000.2020.9219545](https://doi.org/10.1109/ISCC50000.2020.9219545).
- [62] A. Golubev, P. Bogatencov, and G. Secrieru, "Optimal methods of storage, transfer and processing of DICOM data in medical information systems," in *Proc. Int. Conf. Distrib. Comput. Commun. Netw.*, 2017, pp. 269–280, doi: [10.1007/978-3-319-66836-9](https://doi.org/10.1007/978-3-319-66836-9).
- [63] R. Lebre and C. Costa, "A non-relational approach for distributed medical imaging databases," in *Proc. IEEE EMBS Int. Conf. Biomed. Health Inform. (BHI)*, Jul. 2021, pp. 1–6, doi: [10.1109/BHI50953.2021.9508612](https://doi.org/10.1109/BHI50953.2021.9508612).
- [64] Y. Gupta, C. Costa, E. Pinho, and L. A. Bastiao Silva, "A DICOM standard pipeline for microscope imaging modalities," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Sep. 2021, pp. 1–6, doi: [10.1109/ISCC53001.2021.9631529](https://doi.org/10.1109/ISCC53001.2021.9631529).
- [65] J. S. Plank, "Erasure codes for storage systems: A brief primer," in *Proc. Mag. USENIX & SAGE*, Washington, DC, USA, 2013, pp. 44–50.
- [66] W. Ryan and S. Lin, *Channel Codes: Classical and Modern*. Cambridge, U.K.: Cambridge Univ. Press, 2009. Accessed: Apr. 10, 2022. [Online]. Available: <https://www.cambridge.org/core/books/channel-codes/79D1F8D9B618C2CF306C539F29A53199>
- [67] W. K. Lin, D. M. Chiu, and Y. B. Lee, "Erasure code replication revisited," in *Proc. 4th Int. Conf. Peer–Peer Comput.*, Aug. 2004, pp. 90–97, doi: [10.1109/PTP.2004.1334935](https://doi.org/10.1109/PTP.2004.1334935).
- [68] D. A. Patterson, G. Gibson, and R. H. Katz, "A case for redundant arrays of inexpensive disks (RAID)," in *Proc. ACM SIGMOD Int. Conf. Manage. Data (SIGMOD)*, 1988, pp. 109–116, doi: [10.1145/50202.50214](https://doi.org/10.1145/50202.50214).
- [69] J. S. Plank, "The raid-6 liber8tion code," *Int. J. High Perform. Comput. Appl.*, vol. 23, no. 23, pp. 242–251, Jun. 2009.
- [70] J. van der Krogt, "Comparing the reed solomon code to two recently found codes for distributed storage," B.Sc. thesis, Delft Inst. Appl. Math., Delft Univ. Technol., Delft, The Netherlands, 2019. [Online]. Available: <https://repository.tudelft.nl/islandora/object/uuid:beb59701-5506-45a3-8dd7-7d38465342ab?collection=education>
- [71] H. Shi, X. Lu, D. Shankar, and D. K. Panda, "High-performance multi-rail erasure coding library over modern data center architectures: Early experiences," in *Proc. ACM Symp. Cloud Comput. (SoCC)*, Carlsbad, CA, USA, New York, NY, USA: Association for Computing Machinery, 2018, pp. 530–531, doi: [10.1145/3267809.3275472](https://doi.org/10.1145/3267809.3275472).



- [72] H. Shi and X. Lu, "Designing high-performance erasure coding schemes for next-generation storage systems," in *Proc. Int. Conf. High Perform. Comput., Netw., Storage, Anal.* Denver, CO, USA: Colorado Convention Center, Nov. 2019. Accessed: Apr. 10, 2022. [Online]. Available: [https://sc19.supercomputing.org/proceedings/src\\_poster/poster\\_files/spostg143s2-file2.pdf](https://sc19.supercomputing.org/proceedings/src_poster/poster_files/spostg143s2-file2.pdf)
- [73] B. Beach. *Open-Sources Reed-Solomon Erasure Coding Source Code*. Backblaze. Accessed: Apr. 10, 2022. [Online]. Available: <https://www.backblaze.com/blog/reed-solomon/>
- [74] *Backblaze Reed-Solomon Source Code*. Accessed: May 15, 2022. [Online]. Available: <https://www.backblaze.com/open-source-reed-solomon.html>
- [75] B. Snaith. *Expanding the Diagnostic Imaging Workforce: How Assistants are Supporting U.K. Service Delivery*. Multiparametric MRI Maps of the Liver. Accessed: Apr. 10, 2022. [Online]. Available: <https://www.dieurope.com/site/wp-content/uploads/2019/03/DIEurope-FebMarch-2019.pdf#page=64>
- [76] C. Manopoulos, A. Raptis, W. Krishan, C. Mavratzas, M. Drandakis, S. Astraka, I. Kouerinis, and N. M. Vaxevanidis, "Reconstruction, processing and smoothing of surface geometry of a patient specific ascending aortic aneurysm," in *Proc. IOP Conf., Mater. Sci. Eng.*, Dec. 2020, vol. 1037, no. 1, Art. no. 012020.
- [77] J.-J. Kim, "Erasure-coding-based storage and recovery for distributed exascale storage systems," *Appl. Sci.*, vol. 11, no. 8, p. 3298, Apr. 2021.
- [78] C. Aatish and M. A. U. Einstein, "Performance enhancing secure erasure code (PESEC) for cloud storage using random n blocks encryption," in *Proc. 8th IEEE Int. Conf. Cyber Secur. Cloud Comput. (CSCloud)/ 7th IEEE Int. Conf. Edge Comput. Scalable Cloud (EdgeCom)*, Jun. 2021, pp. 13–17, doi: [10.1109/CSCloud-EdgeCom52276.2021.00013](https://doi.org/10.1109/CSCloud-EdgeCom52276.2021.00013).
- [79] Y. Uezato, "Accelerating XOR-based erasure coding using program optimization techniques," in *Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal.*, Nov. 2021, pp. 1–14, doi: [10.1145/3458817.3476204](https://doi.org/10.1145/3458817.3476204).
- [80] Y. Fang, S. Wang, H. Tan, X. Zhang, and J. Zhang, "CLRC: A new erasure code localization algorithm for HDFS," in *Proc. Int. Conf. Comput. Eng. Artif. Intell. (ICCEAI)*, Aug. 2021, pp. 62–65, doi: [10.1109/ICCEAI52939.2021.00012](https://doi.org/10.1109/ICCEAI52939.2021.00012).
- [81] Dell. EMC. *A Technical Overview*. ONEFS, DELL EMC ISILON. Accessed: Apr. 10, 2022. [Online]. Available: [https://cdn-prod.scdn6.secure.raxcdn.com/static/media/DAM\\_cf898bc8-c67e-4be2-9d01-33374e80a748.pdf](https://cdn-prod.scdn6.secure.raxcdn.com/static/media/DAM_cf898bc8-c67e-4be2-9d01-33374e80a748.pdf)
- [82] P. Q. Koziol, *High Performance Parallel I/O*. Boca Raton, FL, USA: CRC Press, 2014. Accessed: Apr. 10, 2022. [Online]. Available: <https://libribook.com/author/prabhat>
- [83] H. N. Bhonge, M. K. Ambat, and B. R. Chandavarkar, "An experimental evaluation of SHA-512 for different modes of operation," in *Proc. 11th Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT)*, Jul. 2020, pp. 1–6, doi: [10.1109/ICCCNT49239.2020.9225559](https://doi.org/10.1109/ICCCNT49239.2020.9225559).
- [84] C. Perez and F. Piccoli, "Estimación de los parámetros de rendimiento de una GPU," *Mecánica Computacional*, vol. 29, no. 31, pp. 3155–3167, Nov. 2010.
- [85] C. J. Chang, J. Chou, Y.-C. Chou, and I.-H. Chung, "ECS2: A fast erasure coding library for GPU-accelerated storage systems with parallel & direct IO," in *Proc. IEEE Int. Conf. Cluster Comput. (CLUSTER)*, Sep. 2020, pp. 349–358, doi: [10.1109/CLUSTER49012.2020.00045](https://doi.org/10.1109/CLUSTER49012.2020.00045).



**ANTONIO F. DIAZ** received the M.S. and Ph.D. degrees in electronic physics from the University of Granada, Spain, in 1993 and 2001, respectively. He is currently an Associate Professor at the Department of Computer Architecture and Technology, University of Granada. His research interests include HPC, communication networks, distributed systems, security, and neutrino telescopes.



**RAUL H. PALACIOS** received the bachelor's degree in systems engineering from the Universidad Autónoma de Tamaulipas, Mexico, in 2003, and the M.S. degree in computer and network engineering and the Ph.D. degree in information and communication technologies from the University of Granada, Spain, in 2007 and 2016, respectively. From September 2012 to November 2012, he was a Visiting Researcher with the Research Centre for Information and Communications Technologies (CITIC), University of Granada (UGR) under the CUMEX fellowship Mexican programme. From 2013 to 2016, he was with the Department of Computer Architecture and Technology and CITIC-UGR under the PRODEP fellowship Mexican programme. From May 2017 to September 2018, he was a Professor at the Universidad Técnica Estatal de Quevedo, Ecuador. From 2018 to 2020, he was with the Sistema Nacional de Investigadores (SNI), Consejo Nacional de Ciencia y Tecnología (CONACYT), Mexico. Since 2010, he has been a full-time Professor at the Universidad Autónoma del Estado de Hidalgo, Mexico.



**WILMAR HERNANDEZ** (Senior Member, IEEE) received the degree in electronics engineering and the specialist degree in microelectronics from the Instituto Superior Politecnico Jose Antonio Echeverria (ISPJAE), Havana, Cuba, in 1992 and 1994, respectively, and the M.S. degree in signal treatment and the Ph.D. degree in electronic engineering from Ingeniería La Salle, Universitat Ramon Llull, Barcelona, Spain, in 1997 and 1999, respectively. From 1992 to 1995, he was a Lecturer with the Electrical Engineering Faculty, ISPJAE, and a Researcher with the Microelectronics Research Center, ISPJAE. From 1999 to 2003, he was with the Department of Electronics and Instrumentation, University Institute for Automobile Research, Universidad Politécnica de Madrid (UPM), Spain, where he was the Technical Director of the Department, from January 2003 to January 2004. From January 2004 to March 2013, he was an Associate Professor of circuits and systems with the Department of Circuits and Systems, EUIT de Telecomunicación, UPM. From September 2014 to September 2015, he was a Researcher with SENESCYT, Ecuador, under the Prometeo Fellowship Program. From December 2015 to November 2017, he was a Professor with the Universidad Técnica Particular de Loja, Ecuador. Since January 2018, he has been a Professor with the Universidad de Las Américas, Ecuador.



**FABRICIO MARCILLO** received the bachelor's degree in systems and informatics engineering from the Autonomous Regional University of the Andes, Ambato, Ecuador, and the master's degree in connectivity and computer networks from the State Technical University of Quevedo, Quevedo, Ecuador. He is currently pursuing the Ph.D. degree with the Research Center for Information and Communication Technologies (CITIC), University of Granada, Granada, Spain. In work experience, from 2002 to 2015, he was a Network Communications and Infrastructure Analyst at Dr. Gustavo Domínguez Zambrano Hospital, Santo Domingo, Ecuador. From 2016 to 2019, he was a Professor at the State Technical University of Quevedo. From 2020 to 2021, he was a Professor at the Technical University of Cotopaxi, La Maná, Ecuador. Since 2019, he has been a half-time Professor at the "Japan" Technological Institute, Santo Domingo and Quito, Ecuador.