Multimedia Tools and Applications
https://doi.org/10.1007/s11042-024-18763-2

Cloud media video encoding: review and challenges

Wilmer Moina-Rivera[1] · Miguel Garcia-Pineda[1] · Juan Gutiérrez-Aguado[1] · Jose M. Alcaraz-Calero[2]

Received: 19 September 2023 / Revised: 29 December 2023 / Accepted: 24 February 2024

Abstract

In recent years, Internet traffic patterns have been changing. Most of the traffic demand by end users is multimedia, in particular, video streaming accounts for over 53%. This demand has led to improved network infrastructures and computing architectures to meet the challenges of delivering these multimedia services while maintaining an adequate quality of experience. Focusing on the preparation and adequacy of multimedia content for broadcasting, Cloud and Edge Computing infrastructures have been and will be crucial to offer high and ultra-high definition multimedia content in live, real-time, or video-on-demand scenarios. For these reasons, this review paper presents a detailed study of research papers related to encoding and transcoding techniques in cloud computing environments. It begins by discussing the evolution of streaming and the importance of the encoding process, with a focus on the latest streaming methods and codecs. Then, it examines the role of cloud systems in multimedia environments and provides details on the cloud infrastructure for media scenarios. After doing a systematic literature review, we have been able to find 49 valid papers that meet the requirements specified in the research questions. Each paper has been analyzed and classified according to several criteria, besides to inspect their relevance. To conclude this review, we have identified and elaborated on several challenges and open research issues associated with the development of video codecs optimized for diverse factors within both cloud and edge architectures. Additionally, we have discussed emerging challenges in designing new cloud/edge architectures aimed at more efficient delivery of media traffic. This involves investigating ways to improve the overall performance, reliability, and resource utilization of architectures that support the transmission of multimedia content over both cloud and edge computing environments ensuring a good quality of experience for the final user.

Keywords Cloud computing · Multimedia · Coding · Encoding · Transcoding · Review · Survey

Wilmer Moina-Rivera, Miguel Garcia-Pineda, Juan Gutiérrez-Aguado and Jose M. Alcaraz-Calero contributed equally to this work.

✉ Miguel Garcia-Pineda
miguel.garcia-pineda@uv.es

Extended author information available on the last page of the article

Published online: 09 March 2024

1 Introduction

Internet traffic has skyrocketed over the past year with the increase in many digital activities due to the lockdowns and restrictions caused by the coronavirus pandemic. Teleworking, streaming, video calls, and online shopping are some of the fastest growing activities, which account for a large share of Internet traffic [1]. For example, in the first half of 2021, bandwidth traffic was dominated by video streaming, representing 53.72% of the overall traffic, with YouTube, Netflix, and Facebook video in the top three [2].

The multimedia sector is experiencing rapid growth as more businesses and organizations turn to video and other multimedia content to communicate with their customers, employees, and other stakeholders. To meet the increasing demand for multimedia content, businesses need a flexible and scalable platform that can handle the large amounts of data and traffic associated with streaming video and other multimedia [3]. The cloud provides an ideal solution for this, as it offers a range of features that are well-suited to the needs of the multimedia sector. Some key benefits of the cloud for the multimedia sector include:

- Pay-as-you-go pricing: Expenditure is based on the actual encoding time utilized, proving to be a notably more economical alternative compared to the acquisition and maintenance of on-premises encoding hardware and software.
- Scalability: Effortlessly adjust the amount of encoding power to align with fluctuating needs. This flexibility is advantageous for variable workloads or when encoding substantial batches of video content.
- Reduced IT overhead: Eliminate concerns related to provisioning, managing, or maintaining on-premises encoding infrastructure. This liberation allows IT personnel to concentrate on more strategic initiatives.
- Improved time to market: Accelerate the delivery of video content to the market through cloud-based encoding. This is attributed to the absence of delays associated with the provisioning or installation of hardware or software.

Overall, the power of the cloud makes it an attractive option for businesses in the multimedia sector looking to deliver high-quality content to their customers and users.

To achieve smooth and uninterrupted streaming playback, the technology used must be designed to transmit a high data flow and prepared to support variations in the demand for content. On the one hand, an elastic infrastructure based on Cloud Computing (CC) and its evolution to modern Mobile Edge Computing (MEC) where the Cloud is extended to the Edge of the network, allows the allocation of resources adapted to each need without excessive or inadequate provisioning, automatically. On the other hand, by processing the video, for instance, making different representations of the video for different devices or network conditions, we have more control over the broadcast quality from the client side [4]. In terms of computational capabilities, it should be taken into account that the growing importance of mobile devices such as smartphones or tablets often requires hardware capable of multi-device streaming, i.e. offering different bitrates and resolutions depending on the connected device, always with the ultimate goal of providing the best user experience.

The agility and elasticity of the Cloud and Edge make it possible to expand the capacity depending on the demand or even scale up resources compared to the rigidity of infrastructures not based on Cloud and MEC architectures.

Also, the relationship between cost and computational resources offered by Cloud and MEC infrastructures, together with the possibility of paying only for the resources used, avoiding overprovisioning problems, make these solutions the best option when it comes to supporting all tasks required by streaming-based content delivery projects [5].

In conclusion, concerning the economic ramifications of employing cloud services for video encoding, it can be affirmed that cloud video encoding constitutes a cost-effective and efficient method for encoding video content for delivery across diverse devices [6]. The particular economic advantages associated with the utilization of cloud video encoding encompass diminished capital expenditures (CAPEX), as there is no requisite investment in costly on-premises encoding hardware or software. Similarly, operational expenditures (OPEX) are curtailed, as the expenditure is incurred solely for the actual encoding time utilized, proving to be more economical than the continual maintenance costs associated with on-premises encoding infrastructure. Ultimately, these economies contribute to an enhanced return on investment (ROI). Cloud video encoding serves as a strategic means to augment ROI by mitigating overall video encoding expenses and expediting the market entry of video content.

Several review papers have explored the domain of multimedia cloud computing, yet none have specifically delved into the intricacies of encoding and transcoding processes in the cloud (see Section 3.1). Given their demand for substantial computational capacity, these processes are relocated to the cloud to optimize costs. This review paper is uniquely dedicated to this area due to the absence of comparable works. It explores a research frontier marked by innovations in enhancing the quality of experience for end-users and optimizing resource consumption to reduce the associated costs of these tasks.

The main goal of this paper is to carry out an exhaustive study of existing work related to video encoding on cloud and MEC architectures. Through this study, the reader will be provided with an analysis of the different published proposals that represent the current state of the art of video encoding on the cloud. Each work has been classified according to the type of cloud technology used, and according to the type of online video delivery. Besides, an analysis of the codecs used in each paper is carried out. Finally, an analysis of the challenges that researchers and industry will have to face to offer high-quality streaming services will be presented.

This can be a useful overview for researchers and practitioners who are not familiar with the topic, or who want to get up to speed quickly on the latest research. The reader can identify gaps in the current research that could suggest new directions for future research. This can be helpful for researchers who are planning their studies and want to know where there are opportunities to contribute new knowledge to the multimedia cloud. Moreover, it can be a good source of references for researchers who are looking for additional reading material on cloud-based video streaming and video encoding/transcoding over cloud/edge computing.

The rest of the paper is organized as follows. Section 2 presents a brief overview of video encoding over the cloud and edge architectures, and the main motivations to converge these two paradigms. Section 3 details the systematic literature review technique carried on to select the papers analyzed in Section 4. An exhaustive review of each selected article is provided in Section 4. All of them have been classified according to the type of technology and online video delivery used. An analysis of the relevance of the selected papers has been carried out according to the number of references received. In addition, future research directions are outlined in Section 5. Finally, we present the conclusions of this research work in Section 6.

2 State of the art

This section first provides a brief overview of video encoding, cloud and edge architectures, and the main motivations to converge these two paradigms. Then, a brief overview of relevant literature reviews and surveys in this domain is presented.

The section is divided into three subsections. First, we will discuss the evolution of streaming with a special focus on the latest streaming methods and the importance of encoding processes in today's streaming systems. Second, we will analyze the importance of cloud systems in multimedia environments and we will detail the architecture of the infrastructure used today to provide streaming services over the cloud. Third, we will analyze other existing surveys and tutorials on similar topics to allow the reader to differentiate how this manuscript goes a step forward in the state of the art.

2.1 Media streaming: focusing on encoding process

Streaming systems have evolved over the years as technology and network connections have improved. From the first streaming systems in the 90s to the current streaming systems, like YouTube or Twitch, all the technologies associated with these systems have significantly evolved. Two clear examples that will be discussed in this subsection are: a) streaming protocols and b) video encoding systems.

In the case of streaming protocols, the main ones worth to be discussed are:

- Real-time Transport Protocol (RTP) and Real-time Transport Control Protocol (RTCP) are described in RFC 3550 [7]. Both of them work over User Datagram Procotol (UDP). This means that the protocol should support by design packet losses as it is not guaranteed that packets will reach their destination. While RTP transports media streams (e.g., audio and video), RTCP is used to monitor transmission statistics and quality of service (QoS) and assists in the synchronization of multiple streams. Over the years, both protocols have been considered the main standard for real-time multimedia streaming in IP networks.
- Real-Time Streaming Protocol (RTSP) is defined in RFC 7826 [8]. RTSP is an application-level protocol, similar to Hypertext Transfer Protocol (HTTP) in terms of operation and syntax, but its goal is to transfer media data in real-time. It is used to establish and control media sessions between endpoints. Therefore, RTSP is a control channel protocol between the media client and the media server, while the multimedia stream is sent by RTP.
- Real-Time Messaging Protocol (RTMP) is an application-level protocol designed for multiplexing and packetizing multimedia transport streams (such as audio, video, and interactive content) over a suitable transport control protocol (TCP), see [9]. To deliver streams smoothly and transmit as much information as possible, it splits streams into fragments, and their size is negotiated dynamically between the client and server.

After looking at some of the most commonly used streaming protocols, there are some drawbacks to be considered. Currently, most content on the Internet is served to customers through content delivery networks (CDN) and most of them do not support RTP/RTCP. However, some work has continued to rely on the promotion of RTP for video delivery, in particular in the context of Web Real-Time Communication (WebRTC) [10, 11] and [12]. This framework was born in 2011 as another technological alternative for real-time transmissions. WebRTC is a collection of W3C and IETF standards that allows the transmission of content in real time between users, with an end-to-end latency of less than half a second.

Streaming video over HTTP has evolved over time. Initially, progressive download was used, where the video was downloaded and played simultaneously. However, this method had limitations in terms of playback control, as users could only watch the portion of the video that had already been downloaded. HTTP Pseudo-streaming emerged as an evolution of this approach, aiming to simulate on-demand streaming with the added benefit of forward and rewind playback capabilities. This method allows users to navigate through the video without downloading the entire file. However, it requires specific implementations on both the client and server sides.

Nowadays, HTTP Adaptive Streaming (HAS) is the number one video technology for most video streaming platforms. In HAS, see Fig. 1, a video is divided into smaller segments, and each segment, s_t, is encoded at several resolutions and bit rates, thus creating several representations, r_j. Each player runs an Adaptive Bitrate (ABR) algorithm to select the most suitable segment to create a seamless playout; the work developed by [13] describes a classification of some of the client-side adaptation techniques (Throughput-Based ABR, Buffer-Based ABR, and Hybrid/Control Theory-Based ABR). There are several HAS format specifications, but the most commonly used are HTTP Live Streaming (HLS) and Dynamic Adaptive Streaming over HTTP (DASH) [14]. These streaming systems require efficient video encoding systems to deliver content in as many resolutions as possible. In addition, to offer the highest possible quality of experience (QoE) [15].

Video encoding has been improved for years, but the increase in device resolutions and/or new types of content requires new codecs with greater compression capacity and minimal quality loss. When developing a codec, not only the final quality is important, but also its performance, resource consumption, encoding time, and final size.

H.264 [16], also known as Advanced Video Coding (AVC), is the most widely used codec for all types of transmissions. It was created in 2003 and was intended to replace the traditional MPEG-4, being named MPEG-4 AVC. This format is currently used for Blu-Ray movies, streaming services, satellite broadcasting, and even digital terrestrial television. However, with the new video resolutions of 4K and 8K, it was observed that H.264 required a large bandwidth. Therefore, new codecs were needed to lower the bitrate of this type of video while maintaining quality. As a result, H.265 and VP9 were born.

The H.265 codec [17], also known as High-Efficiency Video Coding (HEVC), was jointly developed by the Video Coding Experts Group (VCEG) and the Moving Picture Experts Group (MPEG) in 2013. It is a royalty-bearing standard, so it requires hardware manufacturers to pay a license fee to add support, and developers to pay a fee to implement it. The bandwidth required for video transmission is reduced by almost half compared to H.264 without loss of quality. Consider also that H.265 can support resolutions of up to 8 K UHD (8192x4320) and a frame rate of up to 300 fps.

VP9 [18] is the open-source alternative of H.265 and it is royalty-free and developed by Google as the successor to VP8. It was born in 2013 and has had a great impact due to Google's large deployment of its YouTube service and Android operating system. The design goals for VP9 included reducing the bit rate by 50% compared to VP8 while maintaining the same video quality, and aiming for better compression efficiency than the MPEG HEVC standard.

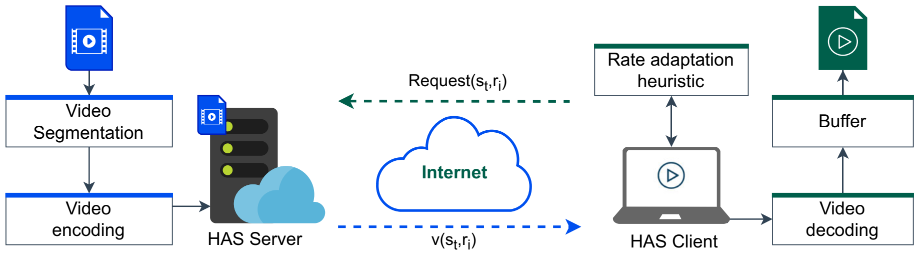


Fig. 1 HTTP Adaptive Streaming (HAS)

VP9 is well suited for video resolutions greater than 1080p until 8 K with a maximum rate of 120 fps.

AV1 was developed to improve VP9. The stable version of the AV1 encoder was released in March 2018 [19]. This is an open, royalty-free video compression format developed by the Alliance for Open Media (AOMedia), a consortium of the most important companies related to the Internet and streaming. This encoder has a compression capacity of more than 30% compared to its predecessors (VP9 and H.265) [20].

AV1, as its predecessors, is a traditional block-based frequency transform format featuring new techniques. Frame content is divided into adjacent same-sized blocks called superblocks (similar to macroblock). Superblocks are square-shaped and they can either be of size 128 × 128 or 64 × 64 pixels, but these superblocks can be divided into smaller blocks until 4 × 4 blocks. AV1 introduces the concept of T-shaped and can divide a superblock into horizontal or vertical splits into four stripes of 4:1 and 1:4 aspect ratio. It expands on non-directional intra-modes by adding 3 new smooth predictors. Moreover, AV1 extends the number of references for each frame from 3 in VP9 to 7.

H.266/VVC, Versatile Video Coding (VVC), [21] is the latest encoding standard offering 50% more compression compared to H.265/HEVC. It increases the block size in the temporal model to 128×128 with binary or ternary partitioned blocks compared to HEVC's quaternary partitioning. It allows different partitioning for the luminance and chrominance planes and enables hardware acceleration through parallel processing. For intra-frame prediction, 67 prediction modes are used instead of the 33 used in HEVC, in addition to enabling the use of rectangular blocks. Inter-frame prediction allows prediction from two reference images, as well as increasing the degrees of freedom of the motion vectors from 2 to 3 dimensions.

So far, the evolution of codecs has always been similar. Quality is improved while maintaining or reducing bitrate at the cost of having encoders that require higher computational capacity. But, what happens if these next-generation encoders want to run on Edge or IoT devices? For this purpose, MPEG/ISO has developed the MPEG-5 Part 2 LCEVC (Low Complexity Enhancement Video Coding) standard [22]. MPEG-5 Part 2 LCEVC is the first internationally accredited enhancement standard for any existing and future video compression scheme. It improves the compression performance of any basic video codec (e.g. AVC, HEVC, AV1, or VVC) and offers improved picture quality with up to 40% lower bit rate for both live and video-on-demand (VoD) transmission.

Figure 2 depicts the current status of the most important codecs H.264, H.265, VP9, AV1, and VVC by carrying out a small comparison among them based on the following four features (see [23] and [24]):

- Encoding Performance: Time to encode a video with similar encoding parameters.
- Decoding Performance: Speed for a given bitstream when played back on an end device.
- Compression efficiency: Subjective and objective quality measures for a given bitrate.
- Ecosystem adoption: Degree of implementation and updating of encoders and decoders.

2.2 Multimedia cloud and mobile edge computing architectures

Today's Over-The-Top (OTT) services require an agile infrastructure to scale instantly and cope with large fluctuations in demand of these services. Cloud and Mobile Edge Computing help streaming services to solve these challenges. Cloud computing has been an evolution of the usage of virtualization to offer a service provider architecture that can be used simultaneously by multiple customers (tenants). Mobile Edge computing has been the evolution of

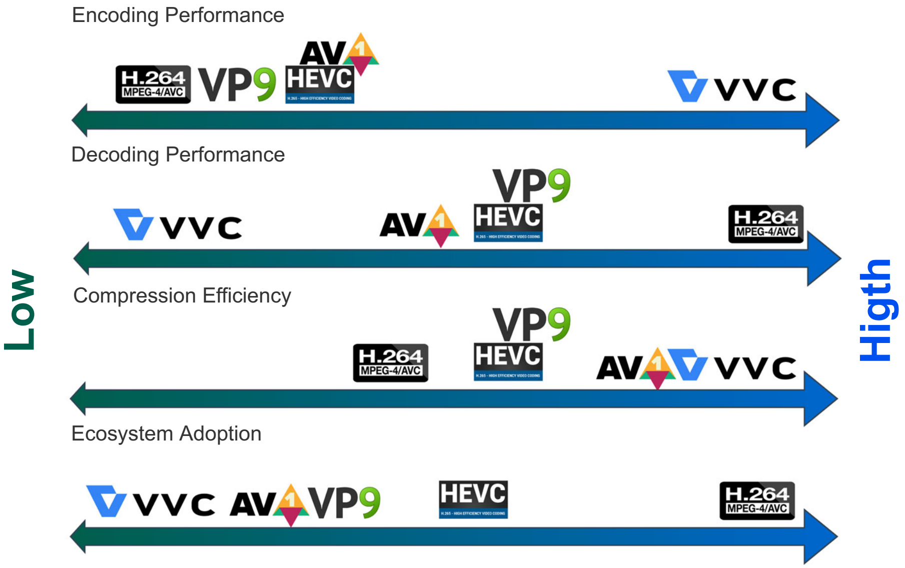


Fig. 2 Current status of codecs: a small comparision

Cloud infrastructure to cover not only the data center network segment but also the last mile close to the final customers.

Multimedia cloud and edge computing share similarities with general-purpose cloud computing [4], but they face additional challenges due to the diversity of multimedia services and the heterogeneity of devices. To effectively support multimedia applications, the infrastructure needs to adapt to various devices in terms of their processing capabilities and to provide QoS guarantees. Two approaches can be employed to achieve QoS provisioning: adding QoS capabilities to the existing infrastructure or introducing a QoS middleware layer. For example, [25] proposes a middleware application layer solution to increase the reliability and flexibility of the network for real-time multimedia applications using scalable video encoding (H.264/SVC).

In essence, the high demands of multimedia data on the Internet could clog a general-purpose cloud/Edge architecture if it has not been properly empowered with support for multimedia capabilities. Today's cloud and Edge architecture have mostly concentrated on providing compute and storage resources. Now, such architectures are being enhanced including techniques and improvements to reach QoS requirements such as increasing bandwidth, minimizing latency, and jitter, etc., and providing novel mechanisms to achieve QoS requirements such as network slicing [26].

Regardless of the IT infrastructure used, streaming processes can be divided into 4 steps (see Fig. 3). In the **Capture Zone**, two tasks can be performed depending on the type of streaming (VoD or live). In the case of VoD, this is where the tasks of content creation and production would be performed. These tasks are performed with raw or very high-quality videos in order not to lose image quality. For live systems, this area performs the ingestion of multimedia content coming from broadcast mobile units and also the production, to select the camera and generate a single stream, in real-time. In both scenarios, the multimedia stream in ultra-high quality is sent to the next step (Video Processing Zone). An example of how

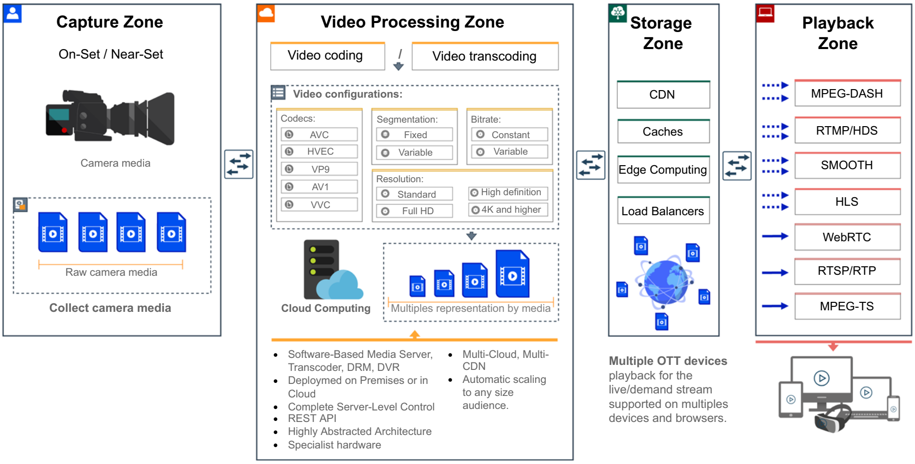


Fig. 3 Multimedia cloud and edge computing architecture

Netflix delivers cloud-based technology and infrastructure to help production crews create and exchange media during production and post-production stages can be seen in [27].

In the **Video Processing Zone**, encoding and transcoding tasks are performed according to the type of stream coming from the Capture Zone. The encodings can be performed for different codecs (AVC, HEVC, VP9, AV1, etc.), for different bit rates, different resolutions, etc. All this is designed for the type of streaming protocol to be used (e.g. DASH or HLS). These processes must be carried out in parallel in order to reduce the delay and be able to offer all versions of the content to the end-user and thus improve the perceived quality.

The **Storage Zone** is key to offering a good service to the end user. Today's cloud systems have distributed storage systems to bring content closer to users. CDN architectures can also be used to offer multimedia content with a reduced delay and without saturating the main content servers, specially making use of the advantage of the Edge infrastructures that allow proactive dynamic allocation of content geographically sited in the last mile to the final user and thus reducing significantly the latency required for the delivery.

Finally, the **Playback Zone** has to be adapted and compatible with the transmission protocols (DASH, HLS, RTSP, RTP, etc.). According to the type of transmission, protocol, or device, the users in this zone must use a specific type of application to access the video content. As was mentioned in Section 2.1, for instance, in a transmission using HAS, the client application controls the ABR algorithm by requesting video bit rates based on observed network conditions and device resolution, thereby regulating the rate of transmission between the client and the server [13].

Following the architecture shown in Fig. 3, nowadays several Cloud Service Providers (CSPs) like Azure, Amazon Web Services (AWS), Google Cloud, and Alibaba Cloud, offer video encoding services. These services are designed to handle the encoding of video content for streaming and broadcasting over the Internet. Below is a brief description of the services offered by these CSPs related to video encoding.

- Azure Media Services Encoding[1]: Azure Media Services offers highly secure encoding workflows suitable for web developers, broadcasters, and studios. It allows users to define their own encoding workflows using tools like Azure Media Explorer and a graphical

[1] Microsoft Azure: Services available at: https://azure.microsoft.com/en-us/products/media-services/encoding

Workflow Designer. The service supports a wide range of input file formats and provides standard MP4 multi-bitrate files to save on storage costs. Azure Media Services also offers on-the-fly conversion to various streaming protocols such as MPEG-DASH, Apple HLS, and Microsoft Smooth Streaming, with the option to add encryption dynamically. Adaptive bitrate streaming is used to enhance the viewer experience by adjusting video playback quality based on the user's connection speed.
- Amazon Elastic Transcoder and AWS Elemental MediaConvert[2]: Amazon offers two services: Amazon Elastic Transcoder and AWS Elemental MediaConvert. Amazon Elastic Transcoder is designed to be a highly scalable and cost-effective service for developers and businesses to transcode media files. It supports a wide range of devices and formats, including segmented files and playlists for HLS, Smooth, or MPEG-DASH streaming. AWS Elemental MediaConvert is a more advanced file-based video transcoding service with a comprehensive suite of features. It offers on-demand pricing and is designed to scale seamlessly with the user's needs. AWS also provides Accelerated Transcoding, which can significantly speed up the processing of file-based video encoding jobs.
- Google Cloud Transcoder API[3]: Google Cloud's Transcoder API allows users to submit, monitor, and manage transcoding jobs within Google Cloud. It supports a variety of features, including content encryption for DRM, ad keyframe insertion, thumbnail creation, and job templates for reusing configurations. The API is based on Google's Identity and Access Management for access control and supports a range of containers and encryption schemes for different streaming protocols.
- Alibaba Cloud ApsaraVideo Media Processing (MPS)[4]: Alibaba Cloud's ApsaraVideo Media Processing service supports all major formats and includes customizable transcoding services. It offers features like video encryption based on the AES128 algorithm, video snapshot capabilities, transcoding templates, and workflows for automatic transcoding after media uploads. The service also provides various features such as video editing, merging, snapshot capture, and watermarking, with the ability to create custom transcoding parameters.

Each of these services offers a unique set of features and pricing models, and the choice between them would typically depend on the specific requirements of the encoding workflow, the desired output formats, the level of control and customization needed, as well as integration with other cloud services and APIs.

2.3 Related work

In this subsection we are going to present other reviews or surveys related to Multimedia Cloud Computing (MCC) previously published, paying special attention to the differences between these previous papers and our review.

The paper [4] introduces the principal concepts of multimedia cloud computing. The authors approached multimedia cloud computing from two perspectives: multimedia-aware cloud and cloud-aware multimedia. They demonstrated how a multimedia-aware cloud may provide QoS support, distributed parallel processing, storage, and load balancing for diverse multimedia applications and services. They explored how cloud-computing resources might

[2] Amazon AWS: Services available at: https://aws.amazon.com/es/mediaconvert/

[3] Google Cloud: Service available at: https://cloud.google.com/transcoder/docs/concepts/overview?hl=es-419

[4] Alibaba Cloud: Services available at: https://www.alibabacloud.com/es/product/mts

be best utilized by multimedia services and applications such as storage and sharing, authoring and mash-up, adaptation and delivery, and rendering and retrieval.

In [28], the authors describe how to use mobile cloud computing to enable multimedia applications on mobile devices. They focus on the technical side and discuss existing challenges and solutions from different perspectives. In this paper, the authors pay attention to showing customers' energy consumption and how the cloud could help reduce it. Potential challenges must take into account the quality of experience, security, and privacy. [29] is another work that surveys the paradigm of cloud mobile media. This paper studies this research area ranging from resource management and control, media platform services, cloud systems, and applications. It is the only survey where the authors analyze the encoding/transcoding process in a subsection called "Media Representation" and where only seven works are described.

Another overview related to multimedia cloud computing is [30]. In this paper, the authors show an overview of the cloud storage system and its security problem. They describe several key ideas and solutions for data integrity, data confidentiality, access control, and data manipulation. The authors conclude by saying that multimedia cloud storage security is still in its infancy and they expect an important improvement shortly. For example, the authors of [31] introduce the concept of multimedia protection based on role access control. They describe the complete process of registration, role assignment, multimedia file owner's request for data encryption, and user login and access to multimedia files, which guarantees security in multimedia files.

The main purpose of [32] is to present some dominant platforms, software packages, application delivery tools, and architectures that might help a multimedia-related application to be easily deployed, maintained, and scaled up in a cloud computing environment. A specific case, based on AWS, is explained in [33]. This work analyzes how encoding or pre-transcoding tasks require high computational capacity, which are brought to the cloud to minimize costs. However, these transcoding tasks are the more computationally demanding ones within the entire streaming pipeline.

Finally, there is another work [34], where the authors explore the latest advances in latency-sensitive video computing in the cloud, which is essential for cloud-supported conversational video services such as cloud gaming, Virtual Reality (VR), Augmented Reality (AR) and telepresence. The authors take a top-down approach to cover the literature: from applications and experiences to architecture and management to optimization in and out of the cloud. They also point to major open challenges and hope to stimulate more research activities in this new and exciting direction.

Summing up, there are several review papers related to the topic of multimedia cloud computing, but none of them focus on the encoding and transcoding processes on the cloud. These processes require high computational capacity, which is brought to the cloud to minimize costs. This review paper is specifically devoted to this topic because there is no similar work and it is a research area with new improvements in the quality of experience for end users and in resource consumption to reduce the costs of these tasks.

3 Research methodology

This study has employed the systematic literature review (SLR) technique proposed by Kitchenham and Charters [35]. The goal is to define appropriate research questions and look up relevant studies that are focused on them. The search is broken down into three phases: planning the review, conducting the review, and reporting the results. These phases are used to determine the status of the research area, as well as to evaluate contributions and gaps for drawing partial conclusions for each research question and to build the general conclusion of the report. In the next subsections, each phase will be described in further detail.

Additionally, the guidelines by Petersen et al. [36] have been taken into account, in particular when classifying the study and identifying its research area. The research steps used by Kitchenham and Charters [35] and Banijamali et al. [37] are shown in Fig. 4. For example, an additional pilot study was performed to examine possible search strings (similar to Kitchenham and Charters [35]).

3.1 Identifying the need for the review

None of the reviews analyzed in subsection focus on the topic of video encoding proposals over cloud infrastructures. To perform this review a systematic process of searching for and evaluating primary sources on the topic at hand has been carried out. In this report, we provide statistical data of the selected papers and a classification of them that can be later used in other research projects to reach a sound understanding of the current state of the problem being addressed.

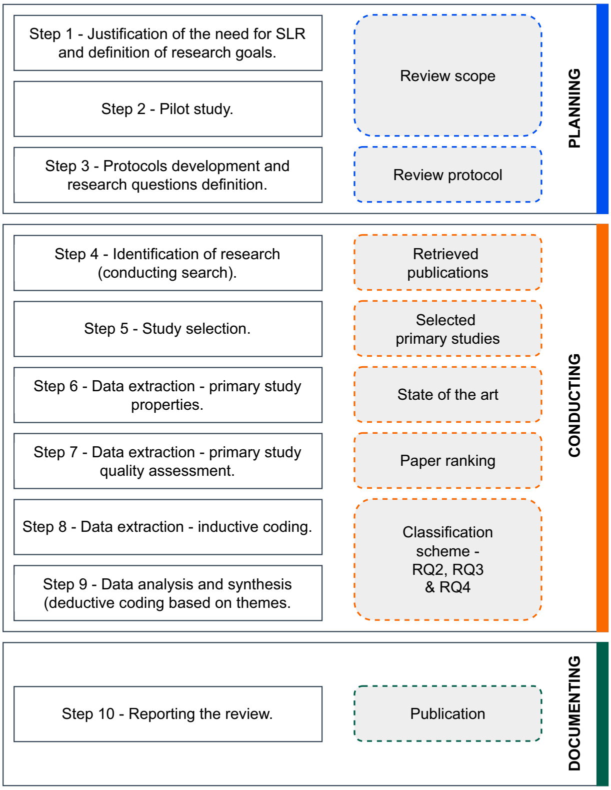

Step 1 - Justification of the need for SLR and definition of research goals.
Step 2 - Pilot study.
Step 3 - Protocols development and research questions definition.
Review scope
Review protocol
PLANNING
Step 4 - Identification of research (conducting search).
Step 5 - Study selection.
Step 6 - Data extraction - primary study properties.
Step 7 - Data extraction - primary study quality assessment.
Step 8 - Data extraction - inductive coding.
Step 9 - Data analysis and synthesis (deductive coding based on themes.
Retrieved publications
Selected primary studies
State of the art
Paper ranking
Classification scheme - RQ2, RQ3 & RQ4
CONDUCTING
Step 10 - Reporting the review.
Publication
DOCUMENTING

Fig. 4 Systematic literature review steps

3.2 Specifying the objective and research questions

To systematically identify the research purpose for this study, we used the Goal-Question-Metric (GQM) technique [38]. A GQM goal is a measuring objective that has been formalized in accordance with specific criteria [39], which are shown in Table 1.

The overall goal of this investigation is defined as follows:

Analysing the implementation of the existing cloud-based encoding designs **for the purpose** of their characterization **regarding** the type of cloud computing, the types of cloud services, and the type of streaming **from the viewpoint of** the researcher **in the context of** the convergence of video encoding/transcoding on cloud computing and MEC architectures. As shown in Table 2, this objective was divided into 4 research questions and their rationales.

3.3 Definition of inclusion and exclusion criteria

We selected studies for inclusion in the literature review if they presented a scientific contribution to the body of knowledge on video encoding in the context of cloud and edge computing. Specifically, we included scientific papers in the context of cloud/edge convergence and video content preparation that meet all the following criteria:

1. implement a video encoding or transcoding system in the cloud,
2. define a video encoding or transcoding method,
3. if the proposal has been implemented and tested in a private, public or Federated cloud environment, and
4. if the year of publication of the work is after 2010.

Accordingly, the following exclusion criteria were considered:

1. studies that concerned the preparation of video content, cloud computing, or grid computing, but not their convergence;
2. studies that dealt with topics other than video encoding or transcoding;
3. duplicate articles;
4. non-peer-reviewed studies, including introductions to special issues, calls for papers, keynote speeches, prefaces, standards, patents, etc;
5. papers that are an extension of a previous paper or papers not written in English have been excluded.

Table 1 Research objective

Analyse	the existing cloud-based encoding designs
For the purpose of	characterization
Regarding	the type of cloud computing architecture, types of cloud services, and type of streaming
From the viewpoint of	the researcher
In the context of	the convergence of video encoding/transcoding on cloud computing and MEC architectures.

Table 2 Research questions (RQs)

ID	Question	Rational
RQ1	Are there many papers on distributed video encoding in cloud environments?	Analyze whether there are many works on the topic and their relevance, due to the great demand for this type of multimedia content.
RQ2	What type of cloud architecture has been used to implement each solution?	Identifying and synthesizing the architecture or paradigm used in selected proposals.
RQ3	What videos codecs has been used in the experiments presented in each paper?	Identifying the video codecs used and the number of codecs used by each paper.
RQ4	Are the solutions intended for video on demand, live streaming, or both?	Identifying and structuring primary studies in terms of video streaming types.
RQ5	How can video encoding systems evolve over the new paradigms of cloud computing?	Identify and argue the possible new challenges related to the topic, and where new video encoding system solutions may be heading in the coming years.

3.4 Definition of the search strategy (meta-analysis)

In the systematic literature review, identifying relevant studies that answer the research questions is an important step [35]. Therefore, to develop and analyze the search strategy, researchers use different guidelines or methods [40].

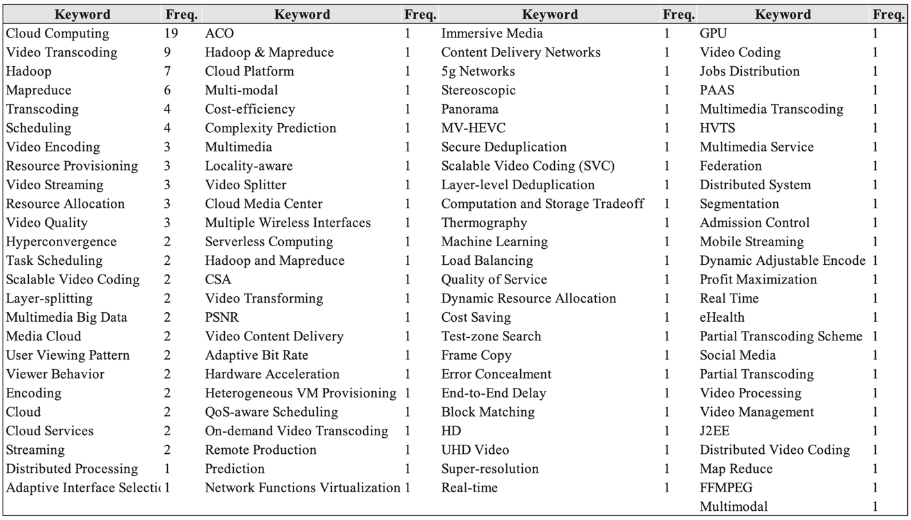
In this work, we have chosen to perform a progressive review of the information, applying sequentially the automatic and snowball search methods according to the guidelines of [41].

To start the work, a pilot study was conducted to define the optimal search strategy that would minimize noise and adequately retrieve relevant studies. The pilot study started by defining a set of keywords with the following search string: *("cloud" OR "cloud-based" OR "architecture") AND ("video" OR "media" OR "multimedia") AND ("coding" OR "encoding" OR "transcoding")*. The string selected by the authors to address the topics related to the research is grouped into three elements with a certain degree of affinity and each group is related to each other by the logical operator "AND".

To objectively evaluate the search string, a search was performed in Google Scholar where the string was adjusted to the advanced search tool by selecting all metadata as possible matches and limiting the search period predefined in the objectives of this study (2011-2022).

By consensus within the working group, the papers were listed by relevance, and patents were excluded. From a total of 443 results obtained, the first 100 articles were selected for the pilot study. The papers were analyzed separately by two authors, to reduce the bias when recording their votes concerning the relevance of the papers.

The pilot study resulted in 36 relevant papers (36%) in the scope of this study. As a next step, an analysis of the metadata collected from this group of papers was performed concluding that the most frequently used keyword in this field was "Cloud Computing". Figure 5 shows the results of the analysis of the keywords according to the frequency of occurrence in the set of relevant studies. It allows the reader to get a complete overview of the keywords associated with the topic.



Keyword	Freq.	Keyword	Freq.	Keyword	Freq.	Keyword	Freq.
Cloud Computing	19	ACO	1	Immersive Media	1	GPU	1
Video Transcoding	9	Hadoop & Mapreduce	1	Content Delivery Networks	1	Video Coding	1
Hadoop	7	Cloud Platform	1	5g Networks	1	Jobs Distribution	1
Mapreduce	6	Multi-modal	1	Stereoscopic	1	PAAS	1
Transcoding	4	Cost-efficiency	1	Panorama	1	Multimedia Transcoding	1
Scheduling	4	Complexity Prediction	1	MV-HEVC	1	HVTS	1
Video Encoding	3	Multimedia	1	Secure Deduplication	1	Multimedia Service	1
Resource Provisioning	3	Locality-aware	1	Scalable Video Coding (SVC)	1	Federation	1
Video Streaming	3	Video Splitter	1	Layer-level Deduplication	1	Distributed System	1
Resource Allocation	3	Cloud Media Center	1	Computation and Storage Tradeoff	1	Segmentation	1
Video Quality	3	Multiple Wireless Interfaces	1	Thermography	1	Admission Control	1
Hyperconvergence	2	Serverless Computing	1	Machine Learning	1	Mobile Streaming	1
Task Scheduling	2	Hadoop and Mapreduce	1	Load Balancing	1	Dynamic Adjustable Encode	1
Scalable Video Coding	2	CSA	1	Quality of Service	1	Profit Maximization	1
Layer-splitting	2	Video Transforming	1	Dynamic Resource Allocation	1	Real Time	1
Multimedia Big Data	2	PSNR	1	Cost Saving	1	eHealth	1
Media Cloud	2	Video Content Delivery	1	Test-zone Search	1	Partial Transcoding Scheme	1
User Viewing Pattern	2	Adaptive Bit Rate	1	Frame Copy	1	Social Media	1
Viewer Behavior	2	Hardware Acceleration	1	Error Concealment	1	Partial Transcoding	1
Encoding	2	Heterogeneous VM Provisioning	1	End-to-End Delay	1	Video Processing	1
Cloud	2	QoS-aware Scheduling	1	Block Matching	1	Video Management	1
Cloud Services	2	On-demand Video Transcoding	1	HD	1	J2EE	1
Streaming	2	Remote Production	1	UHD Video	1	Distributed Video Coding	1
Distributed Processing	1	Prediction	1	Super-resolution	1	Map Reduce	1
Adaptive Interface Selecti	1	Network Functions Virtualization	1	Real-time	1	FFMPEG	1
						Multimodal	1

Fig. 5 Keywords analysis

In addition, it was possible to observe the occurrence of certain words in most of the relevant paper titles such as video (69.4%), transcoding (63.9%), or cloud (41.7%).

Consequently, the pilot study conducted allowed us to modify the search string by replacing the word *"Architecture"* with *"Cloud Computing"*. The same structure and preset order as the pilot search string was maintained, as can be seen in Table 3.

As a next step, the working group defined a set of bibliographic databases to find the relevant papers for this study. This set contains many scientific publications within the field of Computer Science. Among the databases are ACM Digital Library, IEEE Xplore, and Scopus. Kitchenham et al. [42] mention that searching in IEEE, and ACM guarantees good coverage of important journals and conferences and at least two general indexing systems. However, we have also included Scopus. According to [43], Scopus contains a total of about 77.8 million core records while Web of Science covers 74.8 million scholarly data, which will allow us to have at our disposal almost all publications related to the topic.

To perform the automatic search, the search string was structured following the specifications of the advanced search tool of each of the databases. However, a segmentation of the meta-data considered for the retrieval of the papers was performed. In ACM Digital Library, only the abstract was used for the search since it considered a greater number of articles to be retrieved than using a combination of Title + Abstract + Keywords. In IEEE

Table 3 Search keywords

Search strings	
Pilot search string:	("cloud" OR "cloud-based" OR "architecture") AND ("video" OR "media" OR "multimedia") AND ("coding" OR "encoding" OR "transcoding")
Final search string:	("cloud" OR "cloud-based" OR "cloud computing") AND ("video" OR "media" OR "multimedia") AND ("coding" OR "encoding" OR "transcoding")

Xplore the Abstract was used as the search field together with a filter that limits the topics to be considered, such as cloud computing, video coding, video streaming, data compression, transcoding, resource allocation, learning (artificial intelligence), and multimedia computing. Scopus was used because it includes search platforms such as ScienceDirect and indexes external databases. Since this platform indexes scientific papers from external databases, the results obtained from the two previous databases were discarded.

3.5 Data extraction and synthesis

The results obtained using the automated search using the keywords shown in Table 3 were further processed and finally, snowballing search was applied.

1. The automated search described in the previous section serves as the first step in selecting studies. This is in accordance with the application of search strings in the above-mentioned internet databases to obtain the first set of preliminary evidence studies. When the results of all the selected data sources were added together, the initial step yielded 867 works. ACM Digital Library contributed 80 works to the total, 526 from IEEE, and 261 from Scopus.
2. The second step is the analysis of duplicate titles and articles. The group of works obtained in the first step was subjected to the inclusion/exclusion criteria previously established in Section 3.3, which also took into account the titles of each article. The duplicate articles that were obtained from the various online databases were also removed. 664 articles were obtained from this step.
3. The third step is the analysis of meta-data, considering each article's abstract and keywords to determine if they meet the selection criteria or not. Applying this step 70 articles remained.
4. The full-text analysis is the fourth step. In this task, the whole texts of the articles acquired are examined to produce a more in-depth analysis of their adherence to the selection criteria. Those who satisfy the inclusion requirements are then picked after this. The final set of papers consisted of 46 papers in total when this process was finished.
5. Using a snowballing search technique is the fifth step. Applying the selection criteria to the studies located using the first search technique is the final phase. To locate every conceivable piece of evidence, we apply the snowballing search technique approach to pick out any papers that may have been missed by the automated search. This approach entails reviewing the list of references or quotes for each article in the collection of papers, or "backward snowballing," and then "forward snowballing" the citations provided for each piece to look for further sources or original documents. The topic covered in this study has quite specific criteria, so it wasn't possible to pinpoint a specific collection of articles to use while applying the snowballing search technique. Due to the number of citations, it was decided to choose two of them as potential candidates for applying this technique: (1) "Cloud transcoder: Bridging the format and resolution gap between Internet videos and mobile devices." [44], and (2) "Prediction-Based Dynamic Resource Allocation for Video Transcoding in Cloud Computing" [45]. As a result of taking this step, we were able to add another 3 new articles that we discovered after using the snowball technique.

Once the steps of the research method had been followed, 49 articles that adhered to the selection criteria were published between January 2011 and December 2022. These articles are referred to as primary studies. The steps taken in this work and the outcomes attained at each stage are represented in Table 4.

Table 4 Primary papers selected after each step

Steps	Databases	Papers per database	Papers selected
1st Step	ACM	80	867[1]
Automatic search	IEEE	526	
on databases	SCOPUS	261	
2nd Step	ACM	68	664[2]
Analysis of titles	IEEE	416	
and duplicates	SCOPUS	180	
3rd Step	ACM	15	70
Analysis of	IEEE	36	
meta-data	SCOPUS	19	
4th Step	ACM	8	46
Full text	IEEE	27	
Analysis	SCOPUS	11	
5th Step	Snowballing search	3	49

[1] Search date: 03.12.2022
[2] Duplicate papers were not considered

3.5.1 Validity control

Following the working technique used to define the search chain, described in Section 3.4, to reduce bias and subjectivity in the article selection process, two authors analyzed a number of papers and, at their discretion, determined which papers met or not the selection criteria. The authors worked from the second to the fourth step of the search and selection process described above. Together, they processed the meta-data of the articles resulting from step 1, with the aim of identifying repeated articles. All papers were then randomly distributed (50% for each), to reduce bias among revisions. The papers accepted by each author in the different steps were incorporated into the primary set of studies.

In addition, for the last step, the two authors worked together, applying the snowballing technique to ensure that all articles meeting the inclusion/exclusion conditions were found and correctly selected.

3.6 Reporting results

A report of the results obtained (the final stage of the applied article selection methodology) is presented in Section 4 of this work to develop the analysis and classifications extracted from the chosen primary studies.

4 Media cloud encoding: review

4.1 Introduction

Based on the search and selection process described in Section 3, 49 papers were selected related to video encoding or transcoding process over cloud infrastructures. These papers have been published from 2011 until mid-2022 following the distribution presented in Fig. 6. As we can see, from 2015 until 2020, the average number of papers published was 6 papers per year. An adequate number for such a closed topic as video encoding in cloud environments.

From these works we have made a study of the titles and their abstracts to analyze the words with the highest frequency, obtaining the Fig. 7. In this figure, we observe that the five terms with the highest frequency are: video, transcoding, cloud, streaming, and system. All of them are ordered from most to least frequency of occurrence. All terms that appear in Fig. 5 are related to the topic of this review suggesting that the selection of papers has been adequate.

We have grouped the selected works into 3 big groups; according to the type of virtualization technology (Virtual Machines (VMs) vs. Containers), and delivery on the cloud (see Fig. 8). In the group based on VMs, we have distinguished between static clusters developed with MapReduce or other technologies; elastic infrastructures based on public, private, or Federated clouds; and finally, works where only virtual machines were used. Then, we have the solutions based on containers, which have been divided into the works that use serverless, which are based on the execution model where the cloud provider is responsible for executing a piece of code by dynamically allocating the resources, and other solutions implemented using only simple containers. Finally, we have the group based on the delivery, where the papers included in this group show transcoding solutions to improve the delivery of the video stream. In this case, we can distinguish between the solutions through edge architectures or content delivery networks (CDNs).

4.2 Encoding proposals based on virtual machines

4.2.1 Static clusters

In cloud computing, a static cluster refers to a fixed number of VMs that are connected and configured to work as a single unit. These resources may be hosted on a cloud provider's

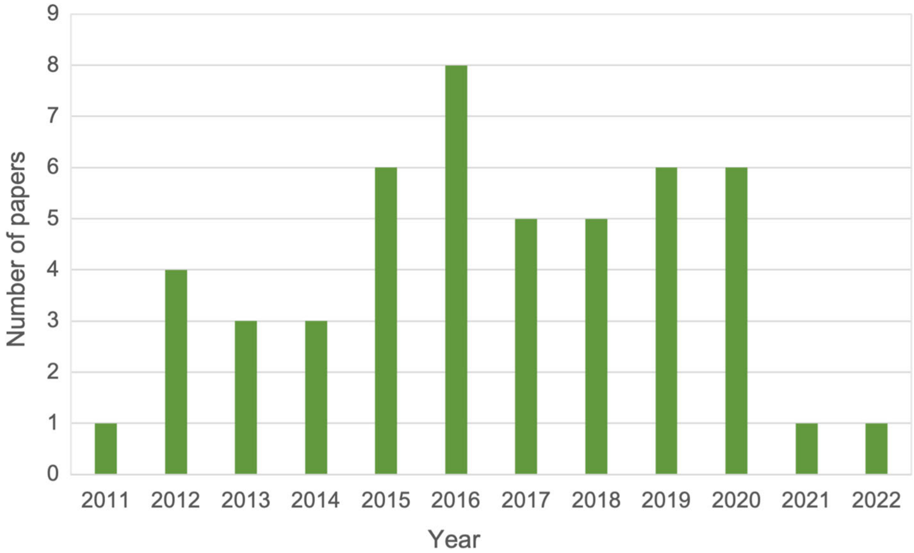


Fig. 6 Number of papers related to the topic published per year

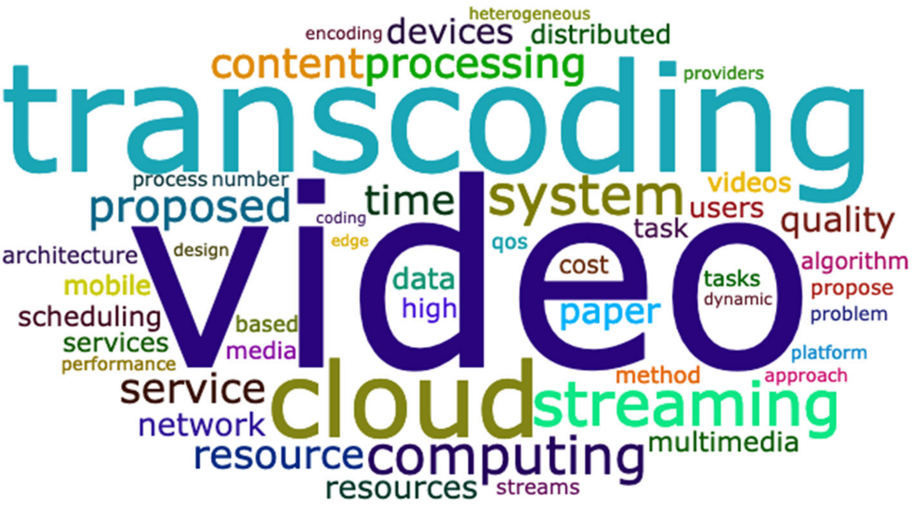


Fig. 7 Word cloud from the abstracts of the selected papers

infrastructure and accessed over the internet through a cloud computing platform. In a static cluster, the configuration of the cluster is fixed and does not change dynamically. This means that the cluster has a predetermined set of nodes or servers that are designated to perform specific tasks, and the resources of the cluster are allocated fixedly.

To use a static cluster for video encoding in the cloud, you need to provision and configure the cluster to meet the needs of your workload. This may involve selecting the types and number of computing resources to include in the cluster, as well as configuring the cluster to allocate resources fixedly.

Once the cluster is set up, you can use it to process video files by submitting them to the cluster for encoding. The cluster will then distribute the workload across the available resources and perform the encoding in parallel, potentially reducing the time required to complete the task. However, if the workload increases or decreases significantly, the static cluster may not be able to scale up or down to meet the changing demands, which could impact the performance of the task.

Two subcategories have been identified inside static clusters: MapReduce and other programming models.

MapReduce MapReduce is a programming model and an associated implementation for processing and generating large data sets with a parallel, distributed algorithm on a cluster. In a MapReduce-based video encoding system, the input video data is divided into smaller chunks and distributed across the nodes of the cluster. Each node then processes its assigned chunk of data using the map function, and the results are passed to the reduce function, which aggregates the results and produces the final output. MapReduce can be an effective way to perform video encoding tasks at scale, as it allows parallel processing and can handle large amounts of data efficiently. Even though it may not be the most suitable approach for all video encoding, due to the difficulty in scaling according to demand, there are quite a few works related to MapReduce because it does not require more complex programming and configuration compared to other approaches.

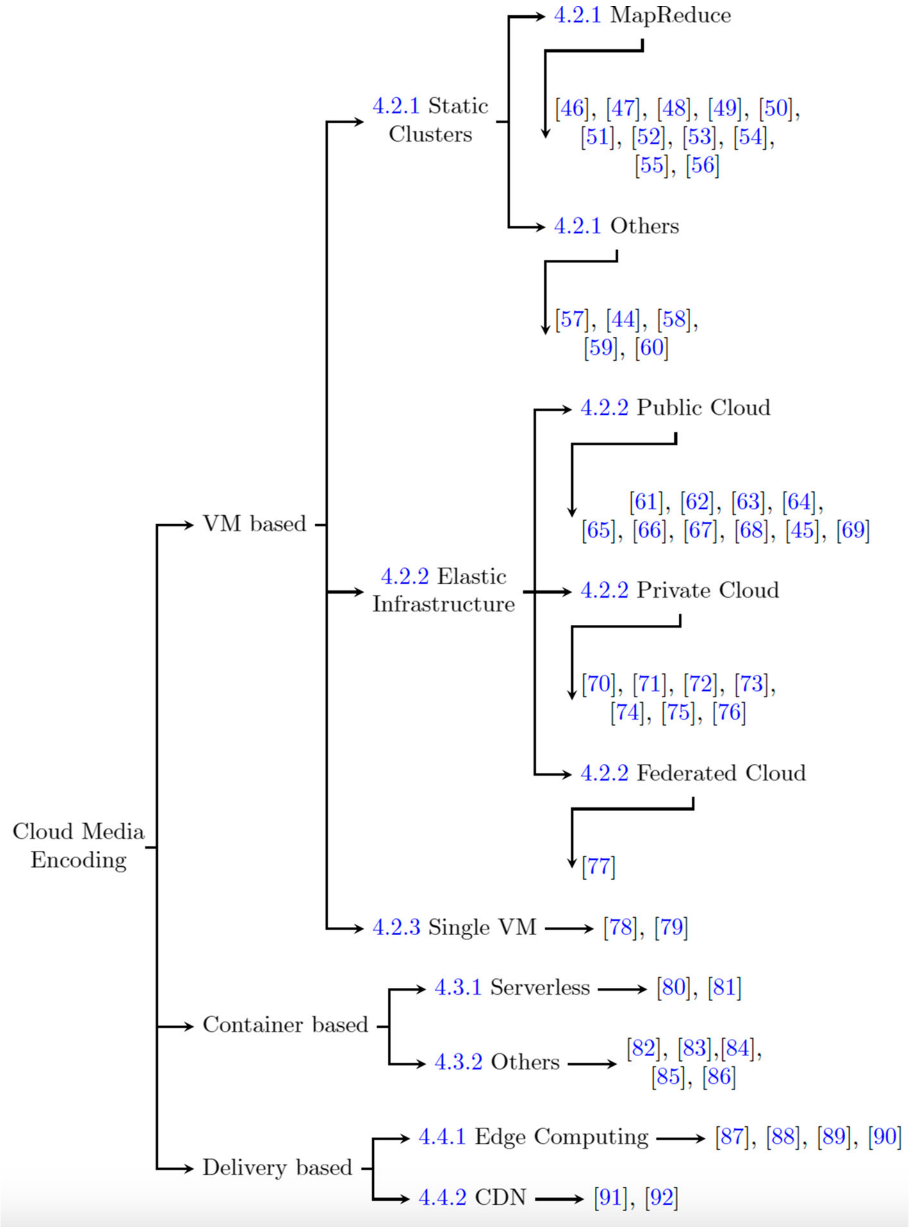


Fig. 8 Classification carried out for the selected papers

A video sequence could be divided into sub-sequences and each sub-sequence into Groups Of Pictures (GOPs). From this idea, the authors of [46] assign each GOPs to the different nodes to take advantage of parallel processing. After the encoding process, each video fragment is merged synchronously. Moreover, the variable segmentation is used and the advantages and disadvantages of obtaining heterogeneous GOPs are analyzed; however, the complexity of the GOP within the compression process is not considered. For the compression process, a speed control scheme is presented according to the type of keyframe identified in the mapping stage. The main drawback of this proposal is that it does not have an experimental phase.

The research paper [47] introduces CTrans, a cloud-based distributed video transcoding system utilizing Hadoop and MapReduce. CTrans analyzes video streams, dividing them into subtasks for individual transcoding. It organizes the output, concatenates sequence parts, and can either transmit them immediately or store them for later use, such as video-on-demand. The architecture comprises input, output modules, and the CTrans core, enabling distributed video encoding. The system strategically uses I-frames for video segmentation, known for encoding efficiency. CTrans dynamically adjusts transcoding formats during operations to cater to diverse device requests. The prototype demonstrates a substantial 70% reduction in encoding time compared to sequential encoding.

Hadoop is an open-source software framework that was born from the MapReduce programming model. In Hadoop, MapReduce is implemented as a programming interface and a runtime system that manages the execution of MapReduce jobs on a Hadoop cluster. Hadoop uses the Hadoop Distributed File System (HDFS) to distribute segments of data among nodes that can be processed massively in parallel.

Using this type of technology, the paper [48] proposes CloudDMSS, a cloud-based distributed multimedia streaming service that adheres to Hadoop's structure while remaining adaptable to cloud computing services. It leverages Hadoop for multimedia data storage and utilizes MapReduce for processing distributed parallel tasks. CloudDMSS consists of three main components: HadoopDMT for transcoding, HadoopDMS for streaming, and CMM for multimedia management. The authors design a workflow based on dual-Hadoop clusters for each physical cluster, introducing algorithms such as content replication, system recovery for HadoopDMS, CMM management, and SRC for streaming job distribution. A web-based control panel is developed to monitor cloud resources. Performance evaluations are conducted on both a local testbed and a public cloud using Cloudit 2.0. Despite challenges in commercial cloud environments (unpredictable networks, VM I/O traffic), CloudDMSS consistently delivers stable and efficient distributed transcoding services, as confirmed by the comprehensive performance tests.

Another proposal with this technology is [49]. This paper presents a cloud video transcoding system using the Hadoop framework. Four video compression standards are proposed: low, standard, high quality, and high definition which are integrated within the framework. The proposal implements a method to split the video into shots. For the transcoding process, a modified version of FFmpeg is used to adjust to the proposed video framework and architecture. Evaluation tests are carried out using several virtual machines (4, 8, and 16) to evaluate the performance of the proposal. The quality tests performed show higher PSNR values greater than 34 dB.

The authors of [50] propose a cloud-based distributed video transcoding system that exploits the dependency between the GOPs of a video to reduce the bit rate and encoding time. They use the Scalable Video Coding (SVC) codec to obtain the dependency information about a raw video, and the encoded video is stored together with the respective GOP distances for future requests. After segmenting the video, the transcoding controller sends each segment to virtual instances in the cloud for transcoding using MapReduce. In the performance tests, they use raw video streams and private cloud architecture with 10 computing units. The overhead introduced by the deployed module is negligible (0.008% to 1.64%) for adjacent tasks. To analyze the bitrate and transcoding time, they compare the proposed method with fixed-size segments (average size of the variable segments). According to the tests, the bitrate is reduced with the proposed scheme up to 10.8% maintaining the quality analyzed with YPSNR. If the video fragment size is set to 64 GOP, the transcoding time increases 53% compared to one GOP. However, with the proposed one, it increases by 21%. Thus, the proposed also transcodes faster than the case with average fragment size.

The capacity of new users to consume high-quality video content opens the need to explore alternative video processing systems that adjust to the current consumption patterns (sources, devices, etc.) and network conditions. The work [51] proposes a dynamic self-adaptive Hadoop scheduling (SAHS) algorithm to maximize resource utilization and decrease transcoding time to improve the QoS for DASH. It uses Hadoop to perform MapReduce, a master node to manage slave nodes of different computational capacities, and a load balancer to divide workloads in similar complexity proportions among nodes. Using packets smaller than 32 MB increases the time to initiate a task and results in a longer transcoding time. Therefore, the strategy of implementing the Max-MCT (MLMCT) multi-layer splitting algorithm and using a packet size of 60 MB would result in a shorter transcoding time. The evaluations were conducted on a server with VMware as a hypervisor, where there are 1 master node and 9 slave nodes.

Other proposal, where a system distributes and parallelizes the video transcoding process and protects the content is presented on [52]. This system analyzes the video input and divides the input into several parts, where each one is individually transcoded. The output of the transcoding process is sorted and merged considering several scenarios. Some of them, where the results will be streamed live over the network, and others where the results are concatenated and stored for later use. Among the objectives of the system are: to allow static or live inputs, to store the transcoding results, to be able to change the transcoding format during an operation, to perform video protection in a distributed way to allow pay-per-quality systems, and finally to allow the system to be self-scalable. Technologies such as MapReduce and MPEG-TS storage are used to reach these objectives. The results of its prototype show that the transcoding time is reduced by 70% when it is compared to sequential encoding.

Scheduling tasks on a cloud video encoding/transcoding platform is an important aspect to be considered, as the ability to manage tasks with available resources can determine the quality of a service or the overall cost of pay-per-use services. The authors of [53] propose a task management algorithm for cloud-based video transcoding that considers the computational load of each transcoding task. Their proposal, called Dynamic Adjustment Slot and Complexity Aware Scheduler (DASCAS), includes a Slot Allocation algorithm, a task selection policy, and a speculative mechanism. The evaluation tests performed the transcoding with the H.264 codec of an HD video of more than 2 hours of duration. Their results show that this proposal decreases the processing time by 18.6% for one user and 11.4% for more than one user, compared to a Fair Scheduler (FS) algorithm.

The work [54] shows a multimodal multimedia cloud architecture that contemplates storage, networking, virtualization, resource allocation, and efficient QoS capability. The service layers used by the solution are infrastructure, platform, and service. Its platform as a service called CEMMS implements the transcoding system using Hadoop. This last one allows for storing, distributing, and processing multimedia data. The proposal includes a method for splitting the video before processing. Xuggler and ACO (Ant Colony Optimization algorithm) media processing libraries are used to determine the size of the cluster in the infrastructure layer. The tests use the FFmpeg tool and video clips in MPEG-2 format of 16 seconds each, with Full HD resolution (1920×1080 pixels) at 25 frames per second. The simulation shows that the proposed CEMMS can optimally utilize the cloud resources to achieve the minimum response time with the minimum resource cost and QoS guarantee.

Another proposal for encoding video using the Hadoop framework is shown in [55]. The proposal uses the Map Reduce algorithm to introduce an intelligent division strategy for the blocks to be encoded. Taking performance metrics, the authors consider a proper block size of 64 MB to be more efficient. The closest blocks belonging to the same coding task will be processed by adjacent heterogeneous nodes to avoid network load. A set of metrics is used to evaluate the quality and performance of the proposal. Experimental results show a 7% improvement in video distortion for a fixed 64 MB block size and a 3% improvement in frame dependency distortion. In general terms, the proposed method reports an improvement of around 6-9% in transcoding time.

The work [56] presents Stride, a distributed video transcoding system based on the Apache Spark Big Data platform. It uses FFmpeg as a transcoding tool and analyzes the impact of various parameters, such as segment size and the number of threads used by FFmpeg. They found that larger segments require more memory, so they chose a segment size of 2 seconds to provide robustness and loss recovery. They also observed that partitions containing a smaller number of segments could provide the TaskScheduler with more flexibility to minimize the idle time between cores. However, this could potentially increase Spark's scheduling overhead. Relative to local implementation, Stride uses less than 19.86% of time, using 4 times the amount of cores (32 cores), further illustrating the value of Spark scheduling. It also outperforms Scala implementations by 1.9% to 2.8%. In terms of scalability, Stride as it is given more resources gets an average speedup of 30% more for every 8 cores added to a worker node. For the testing stage, they considered one virtual machine (VM) for the master node and 4 VMs as worker nodes, 3 videos in 4K resolution, and a Constant Rate Factor (CRF) of 28.

Others In addition to MapReduce, other technologies allow the development of solutions on static clusters in cloud environments. In this subsection, we will present other existing works on video encoding over static cloud environments.

For example, [57] proposes a method for analyzing and optimizing parallel video transcoding for clusters with heterogeneous machines. They explore a taxonomy of video transcoding methods using static (or semi-dynamic) scheduling. The analysis is based on the costs incurred in the different transcoding methods on heterogeneous platforms, considering CBR (Constant-Bit-Rate) and VBR (Variable-Bit-Rate) encoding. They provide closed-form solutions for the segmentation problem in the case of CBR input content, and the appropriate heuristics for VBR content are considered. The study presents extensive experimental results on the most important encoding strategies, where they consider the execution time, and also the characteristics of the output produced. The presented performance metrics are based on actual transcoding of full-length movies, with a total of 3.7 hours and 325000 frames on average. The proposed method achieves a trade-off between execution time and distortion by combining the master node communication with N workers at the same time.

In this work, the authors use cloud computing to offer a transcoding service that covers the format and resolution limitations of mobile devices. This service uses the video stream link to store it in a cloud cache and encode it according to the user's requirements. As a result, the power consumption of mobile devices is minimized and a cache copy is available for future requests. The proposal implemented 244 servers, with 340 TB of cache in the cloud, a bandwidth of 6.9 Gbps per node capable of handling 100K requests daily. FFmpeg is used as a tool to process video content. In addition, it integrates a solution to cache the most popular videos, called "Task Predictor". Performance tests showed that 85% of the encoded videos came from P2P links.

The paper [58] presents a cloud-based transcoding system based on the computational capacity required for video transcoding taking into account the characteristics of video segments and load balancing in the cloud. A parallel transcoding framework is presented. Their strategy contemplates the adaptive segmentation of the video according to its complexity and granularity. The derived tasks are divided into subtasks, and these last ones are executed in the cluster according to the load balancer strategy (where the load on each core must be balanced) and Minimal Finish Time (MFT). This set of processes constitutes the heuristic algorithm, named the minimum longest queue finish time algorithm (MLFT). The simulation results show that the proposed algorithm outperforms the algorithms proposed by the literature in all cases (overhead and average task finish time).

Quality of experience (QoE) is a measure of how well a multimedia streaming system delivers content to the end user. It takes into account various factors such as the video or audio quality, the stability of the stream, the amount of buffering or latency, and the overall usability of the system. One key aspect of QoE is the quality of service (QoS) at each step of the streaming process. This includes the quality of the network connection, the performance of the servers and infrastructure, and the efficiency of the streaming protocols and algorithms being used. Ensuring a high QoS at each step is critical to delivering a good QoE to the end user. This requires careful planning and management of the entire streaming system, as well as regular monitoring and optimization to ensure that it is operating at peak performance. The paper [59] proposes a live video transcoding (COVT) system that optimizes resource provisioning and task scheduling on the cloud to ensure a good Quality of Service (QoS). The system focuses on two key parameters: system delay time and target video segment size. It utilizes the FFmpeg transcoding tool and offers different transcoding modes to adjust processing time and output segment size. COVT employs performance profiling for each transmission mode using historical data. Based on these profiles, it develops a prediction model to determine the optimal probability distribution that minimizes the number of CPUs required for content preparation while maintaining QoS quality constraints. Additionally, the proposed scheduling algorithm utilizes QoS measurements to prevent outages or prediction errors in bursty workloads. The system architecture comprises three main components: video consumer, video service provider, and cloud cluster. The authors implemented a prototype system for validation and evaluation, using a cluster with six virtual machines and an additional server for resource scheduling and communication with the cluster. They compared the proposal with two schemes: peak load provisioning and heuristic provisioning. Overall, COVT can save 45% of resources in terms of CPU hours compared to peak load provisioning, and with heuristic provisioning, it consumes fewer resources at the beginning and end but fails to meet QoS requirements.

Finally, some effort is devoted to mitigate the workloads in data centers involved in processing video content. Paper [60] presents a new video transcoding system that utilizes hardware acceleration to achieve significant performance improvements over traditional CPU-based solutions. The system is designed for large-scale video processing and includes a hardware acceleration block with Video Coding Units (VCU) that are deployed in a distributed warehouse-scale processing infrastructure. The system uses the Borg task scheduler to manage in parallel the loading, segmentation, concatenation, and distribution of video content in its global network. The experiments conducted on thousands of servers indicate that the system offers between 20 and 33 times better performance per cost compared to the previous system designed with last-generation CPUs. The system supports both x264 and VP9 codecs and can handle a variety of output characteristics, including on-demand, video conferences, and live broadcasts. The infrastructure is designed with VCU machines for video transcoding and non-accelerated machines to manage the distribution of work across nodes. The production acceleration system with 10 cards (20xVCU) is compared to some base cases, including a dual-socket server with x86 Intel Skylake CPU and 384 GiB of DRAM, a system with 4 Nvidia T4 GPUs with the dual-socket server as host, and the baseline of 4 GPUs with 4 additional VCU cards. Taking the CPU as a baseline, with x264 the GPU improves performance by 3.5x more, and the system with 20xVCU by 20.9x times. However, with VP9 the system with 20xVCU is 99.4x superior to the CPU. The quality analysis shows that x264 VCU has an average bitrate 11.5% higher than libx264(software), and VP9 VCU has a bitrate 18% higher than libvpx (software).

4.2.2 Elastic clusters

An elastic cluster is a group of computing resources that can be scaled up or down as needed to meet the demands of a workload. An elastic cluster in the cloud can be a useful resource for video encoding or transcoding tasks, particularly if the workload is variable or unpredictable. In a cloud environment, an elastic cluster can be configured to scale up or down as needed to meet the demands of the video encoding process. This can help to ensure that the cluster has sufficient resources to complete the task in a timely manner, while also minimizing costs by only using the resources that are needed.

To use an elastic cluster for video encoding in the cloud, you will need to provision adaptively the cluster to meet the needs of the workload. This may involve selecting the types and number of computing resources to include in the cluster, as well as setting up rules or thresholds for scaling the cluster up or down. Once the cluster is set up, you can use it to process video files by submitting them to the cluster for encoding. The cluster should then distribute the workload across the available resources and perform the encoding in parallel, potentially reducing the makespan to complete the encoding process. As the workload changes, the cluster should scale up or down.

In the following points, we are going to analyze the existing works to encode video on cloud architectures based on elastic clusters. To have a better classification of the papers, we have subdivided them according to the cloud infrastructure used, whether public, private, or Federated.

Public clouds Public cloud refers to a cloud computing model in which resources such as computing, storage, and networking are made available over the internet to anyone who wants to use them. Public clouds are owned and operated by third-party companies, which provide access to their infrastructure and resources on a pay-per-use basis.

Some papers, that present simulations or train models are classified in this category because they make assumptions or use data based on some public cloud infrastructure.

To adjust the allocation of resources to the transcoding task, [45] presents a resource provisioning (VM) algorithm for video transcoding. Using a prediction module, the master node dynamically allocates resources to the transcoding servers, both of which can simultaneously process several segments at a time. The segments have different lengths, where each GOP is composed of only one I-frame. To detect these frames, the video is spatially reduced (from 16 CIF to 1 CIF). Simulation tests were performed with the SimPy framework. The results of the simulation show that the proposed algorithm compared to a base case provides a sustainable service with the lowest number of virtual machines in the different workloads tested. As inputs, 24 and 30 fps videos are taken, with 70% of them in SD and 30% in HD, with an average sequence size between 15000 and 18000 frames and a maximum GOP size of 250 frames.

With the premise of decreasing the delay in starting video transmissions and minimizing resource provisioning costs, a cloud-based video streaming (CVSS) architecture is presented in [61]. It maps transcoding tasks to manage dynamic resource provisioning that does not impact QoS both at the start and during video streaming. The architecture is composed of six main components and covers video segmentation as well as the caching policy. The job manager is responsible for managing the tasks between the virtual machines of homogeneous resources. The scaling manager is responsible for the provisioning of resources according to QoS terms. However, they have been able to estimate that the execution time of the same GOP can vary on a machine with the same performance; they attribute this to the fact that a virtual machine can run on different hardware from the same provider. Simulation tests, with specifications from Amazon EC2 virtual machines, show that the strategy of encoding the shortest job incurs less cost and shorter startup time than other proposals considered. Its provisioning policy reduces the cost by 70% compared to static policies.

The features of mobile devices are increasingly higher, one of them is the resolution of their screens capable of playing high-quality videos. However, the battery is still a limited resource, so the content has to be encoded on the server side taking into account the reduction of overhead on client devices. For these reasons, [62] introduces a cloud-based real-time video transcoding framework designed to adapt high-quality, low-latency video for mobile devices. Implemented in Microsoft Azure, this framework dynamically scales by creating transcoding channels for each request, accommodating low-latency requirements, such as those in video conferences. To manage tasks efficiently, minimize costs, and address latency issues, the system incorporates a predictive-based task scheduling algorithm, considering the pay-per-use model of VMs. The architecture comprises components like resource monitor, transcoding, controller, and task manager, integrating the Azure messaging queue service. Multiple filters can be applied to the same video to generate outputs for various devices. The evaluation tests, utilizing the H.264 codec and FFmpeg tool for tasks like scaling, decoding, and transcoding, reveal that congestion between virtual machines is a significant contributor to latency. The paper suggests that employing a dual-level tree topology in Azure can decrease computational load and latency by leveraging multiple transcoding sources.

In addition, cloud computing has made it possible to redefine video content consumption models. Its constant evolution and its ability to manage jobs with a high computational load allow exploring new alternatives to processes such as video transcoding. Therefore, the authors of [63] developed a video transcoding platform using Google's cloud computing infrastructure. The system is composed of two subsystems: servers and workers. Servers handle client requests and perform an analysis of the results, which can be consulted after video encoding. The server can replicate on the global network depending on client requests. The second subsystem is made up of a group of workers that perform the transcoding processes. To store the tasks that the workers periodically consult to select a task according to the established priority policy, a database is required. A worker during transcoding will not perform additional tasks to avoid memory errors, network problems, etc. The authors conducted evaluation tests to measure the efficiency and scalability of the system. Results show that the proposed platform reduces latency compared to a local server.

As it is known, the demand for high-quality video content from mobile devices is growing; however, due to network limitations (bandwidth, fluctuations, etc.) and the limited capacity of mobile devices (processing, battery, etc.), providing this type of service remains a challenge. The proposal [64] implements an infrastructure as a service (IaaS) to address this problem. It allows them to focus on the dynamic allocation of resources in the video transcoding processes, where the processing of the different representations of the content are performed on the server. The architecture uses temporary storage of the most requested video sequences to avoid transcoding processes. Each video input is segmented into chunks and distributed as jobs to the different transcoding servers, dynamically provisioned by a load controller. The proposal contemplates the assignment of priority to videos based on their request. In addition, a module to verify the availability of the cached stream is introduced. They use the CloudSim simulation framework to evaluate the proposal. They compare with the conventional first-come-first-serve (FCFS) method and with session-based admission control [65]. The H.264 codec is employed at a fixed resolution of 720 × 1080, and frame rate change from 20 to 30 fps. Of the three test cases analyzed: (i) when the loads fluctuate insignificantly over time, (ii) the load has jumped a lot from the previous case, (iii) the load suddenly drops by a large number; only in the first case, the load predictive method is efficient. Therefore, the proposal can be considered only when the workload is constant.

Other proposal is [66]. This paper presents the Differentiated cloud-Assisted VIdeo Streaming (DAVIS) encoding framework optimizes video encoding and transcoding from a single source to multiple destinations. It dynamically selects the source and the encoding parameters and it uses Forward Error Correction (FEC) to optimize quality and maximize network utilization. The DAVIS uses end-to-end delay differences to perform quality adjustments, and its multiple destinations based on the Gilbert loss model and Markov Chains. The tests are performed in a real scenario with AWS EC2 and the Exata emulation platform, and show an improvement in objective quality, in terms of PSNR, of between 5.12 to 10.2 dB compared to literature work. It also guarantees 96.5% frame delivery under delay constraints.

One of the challenges of on-demand video streaming is how to use cloud resources to provide a proper QoE for end users during the streaming. From this idea, [67] develops the Cloud-based Video Streaming Service (CVS2) architecture for video-on-demand transcoding using cloud services. This architecture uses a QoS-aware task scheduler for mapping transcoding tasks concerning the QoE perceived by users. The proposal performs a heuristic mapping of the batch queue (main queue) to determine which type of virtual machine uses less completion time; the virtual machine will store jobs in a local queue to minimize downtime. In addition, the component responsible for managing the provisioning of heterogeneous VMs within the cluster is developed to minimize Streaming Service Provider (SSP) costs while maintaining QoE. The evaluation was carried out on a set of heterogeneous AWS machines; it was shown that the complexity of a GOP determines whether it is necessary to pay for a machine with high performance (GPU, dedicated memory, etc.). The performance tests were performed on the CloudSim simulator, using the characteristics of the virtual machines provided by AWS for the experiments. Their results indicate that the start and finish times of a task are mitigated by using heterogeneous machines. Also, it shows a saving of up to 85% in processing costs concerning static provisioning (homogeneous machines).

Some cloud service providers offer transcoding services to their users with a specific QoS guarantee, which also appears as a Service Level Agreement (SLA). This can be a challenge for cloud service providers to provide computing resources and schedule transcoding tasks while meeting the SLA requirement, more so during live streaming. Therefore, [68] presents a system that manages resource provisioning and video transcoding tasks under dynamic and uncertain live workloads over the cloud. The system uses Deep Reinforcement Learning (DRL) to train a Neural Network (NN) model to determine resource provisioning periodically. The NN model is trained using historical transcoding task data. The system also includes a method to estimate transcoding times. The estimates made are considered to schedule the transcoding tasks taking into account the real-time QoE to meet the criteria of an SLA. The system consists of two components: the environment and the agent. The environment includes a task queue, a task scheduler, and instances. The agent is responsible for training an offline NN model by transcoding task histories and using the model for online resource provisioning. The authors use a simulation system with market characteristics and recommendations for the training process. They use A3C, a state-of-the-art actor-critical learning method, as their training algorithm. Performance tests show that the DRL-based policy decreases the average task failure rate by up to 89%, while the average VM cost increases by a maximum of 4% over the other 3 base cases.

A scalable video transcoding system with a producer/consumer model over the cloud is presented in [69]. The authors of this paper survey the challenges involved in applying large-scale video processing in public clouds. The paper demonstrates a hybrid design that includes a Java application and a bash script. The Java application forms the batch processing framework, while the bash script performs the actual transcoding work (uses FFmpeg). The proposal starts from the study of an existing transcoding system that reaches linear horizontal scalability of up to 1000 vCPU cores, however, after reaching this number, performance degradation is experienced. To improve the performance of the system they integrate a message queuing layer, converting the system into a batch model. This improves the horizontal scalability of the video transcoding system. Large-scale testing on AWS EC2 indicates that the scalable video transcoding system maintains linear horizontal scalability at 10100 vCPU cores.

Cloud providers offer a catalog of instances for different workloads. Instances can be provisioned using a combination of resources such as memory, CPU, storage, GPU, etc. The configuration of these resources implies different costs, therefore finding the instances that best fit the workloads could help to reduce costs in this type of service. The work [70] shows a method for the fast estimation of the encoding time of video segments and an algorithm that combines the Pareto frontier and clustering techniques to calculate the number of instances and their resources best adjusted to a video encoding task to minimize costs. The algorithm uses x264 codec, so they perform a study of its different encoding parameters using video sequences of different complexity. The results show a cost reduction of 15.8% and up to 47.8% compared to a random selection of EC2 instances.

Private clouds Private cloud refers to a cloud computing model in which resources such as computing, storage, and networking are made available over a private network, usually to a single organization. Private clouds are typically owned and operated by the organization that uses them, and they are usually hosted on-premises data centers.

One of the first works where an elastic cluster was used on a private cloud architecture is [71]. The paper presents a transcoding system for real-time and cloud-based mobile streaming that considers bandwidth constraints and different requirements of mobile devices. The system uses map-reduce to manage transcoding jobs in a distributed environment and H.264/SVC multimedia files to improve the dynamic adjustment mechanism and avoid image quality loss. The system adjusts the most appropriate SVC codec to process the request based on the network parameters, hardware characteristics, and profile agent (device). A prototype was designed with three servers, 6 VMs per server, a mobile device, and a 3G Wi-Fi network device. Tests were performed in different transmission states and show that the system can maintain a certain level of service quality for dynamic network environments and ensure smooth and complete multimedia transmission services.

Due to the inherent complexity of video transcoding processes, multiple software solutions have been proposed to prepare this content. However, it remains a challenge to dynamically determine the optimal resource allocation to save costs and ensure the QoS. With this premise, the authors of [72] present an algorithm for dynamic resource allocation based on the large deviation principle, capable of determining the number of nodes needed based on transcoding time fluctuations (transcoding jitter). To measure the trade-off between cost savings and QoS guarantee, they designed a prototype architecture based on a cloud system. In the proposed architecture, the load analyzer is the main component and the one in charge of dynamic resource allocation that guarantees the probability of transcoding jitter below a desired threshold. For this purpose, the algorithm analyzes the bit rates in an iterative process to determine if the number of available nodes satisfies the QoS. Evaluation tests are performed on OpenStack with one controller node and 9 compute nodes, using FFmpeg for transcoding 1000 video segments of a duration between 30 sec to 10 min. Results show that as the number of additional nodes increases, workload transcoding is reduced for future stages. They also show that different QoS can be achieved by controlling the jitter threshold, keeping the cost as low as possible.

The paper [73] presents the design and implementation of an open-source video encoding and transcoding system called Morph, deployed on the cloud. Morph offers parallel video segment transcoding, task management with the ability to adjust to QoS profiles, and elasticity in resource provisioning. The system's architecture consists of three layers: the interface layer, the scheduling layer, and the provisioning layer. FFmpeg is used for transcoding tasks, with two segmentation methods: fixed and dynamic. The system was deployed in a real environment, and bottlenecks were identified, which were solved by using load balancers. Processing time analysis was performed to evaluate the system's performance given the number of available workers.

Lately, video games in the cloud are gaining popularity and by 2023 the industry is expected to generate around 4 billion dollars. However, the limitations of the network or the capacity of devices (e.g. mobile) to consume high-quality content has meant considering strategies for the delivery of this type of content. Therefore, the paper [74] presents a Content-Aware Video Encoding (CAVE) method, where the highest number of bits is assigned to the regions of each frame that the player considers important. For this approach, weights are determined in each of the blocks of a frame from the player's perspective which will then be considered in the encoding. CAVE is implemented as a software component between the game process and the encoder. It uses the HEVC codec and cloud computing infrastructure to meet the low latency requirements that require it to run in real-time. In the executed tests performance metrics such as SSIM, VMAF, BD-rate, quality fluctuations, and CPU time are measured.

Another proposal related to video transcoding in an elastic cluster over a private cloud is shown in [75]. This paper presents Diversify Scheme for Multiform Video Resources (DSMVR), a multimedia data hiding video transcoding scheme based on the framework of parallel computing and intra-cloud environment. DSMVR integrates an algorithm for scalable task allocation and a parallel computing framework for video transcoding in multiple representations. The architecture consists of a server, administrator, worker, and storage system (NAS). The administrator generates a transcoding request to the server, and the server is responsible for managing the tasks among the worker nodes. To know the relative performance of each worker, a previous codification is performed. A NAS server stores the generated data. DSMVR was implemented on Java. Performance tests were performed with 1 and 64 desktop workers. Results show that the scheme with 16 workers achieves the maximum transcoding speed. If the number of workers is increased, the performance decreases.

Some authors of this review propose a new distributed cloud-based architecture and application to accelerate video encoding in [76]. The design of this proposal contemplates horizontal elasticity, software-defined storage, computation, networking, adaptability with multiple codecs, fault tolerance, and automation of encoding operations on worker nodes (virtual machines). In the proposed solution, a video is divided into segments of fixed size, for each segment an encoding job is generated. Each job is consumed by a worker node that performs the encoding of the segment. Some technologies used for the implementation of the proposal are OpenStack for the deployment of a Virtual Infrastructure, RabbitMQ as message manager, Java (JRE) as execution environment, and FFmpeg as encoding tool. H.265 and VP9 video codecs with different segment sizes and target bit rates are used in the validation process. The proposal is evaluated in terms of scalability, workload, and work distribution, varying the number of workers from 1 to 15. They indicate that 89% less encoding time can be achieved compared to sequential encoding when using 15 workers. In terms of quality, using PSNR, the proposal achieves a similar quality to sequential encoding.

Finally, the same authors propose another distributed video encoding architecture [77] to take advantage of the elasticity, fault tolerance, and hyper-convergence that cloud computing provides. They implement a distributed application to dynamically schedule encoding tasks among several workers deployed over a virtual infrastructure. The solution design, implementation, and validation phases consider RAW content, uniform segmentation, message middleware with RabbitMQ, H.265, VP9, and AV1 video codecs, and quality metrics such as PSNR, MS-SSIM, and VIF. The designed architecture was deployed on a private cloud to achieve hyper-convergence and the highest network performance. Their experimental results show a significant decrease in execution time is achieved compared to full video encoding (1 worker), from 61% to almost 91% when using 3 to 15 workers, with an efficiency higher than 80% with 9 workers. In terms of quality, it is indicated that H.265 code does not reach the quality levels obtained with VP9 and AV1 for all the metrics considered. In addition to this, H.265 is the most affected by the change in segment size.

Federated clouds Federated cloud refers to a cloud computing model where different cloud computing infrastructures support the sharing of arbitrary resources, from arbitrary application domains with arbitrary consumer groups across multiple administrative domains via a common standard. Participants in a federation can share some of their resources and metadata making them discoverable and accessible to the rest of the participating members.

There are few jobs where a federated cloud is used to perform video encoding tasks, the only example found in this review is [78]. This paper proposes a comprehensive framework for the development of federated cloud video processing services for social network providers. The system in their proposal includes several cloud providers (CP) implemented with CLEVER middleware along with eXtensible Messaging and Presence Protocol (XMPP), this includes the federation capability and distributed processing with Hadoop technologies. They use Amazon S3 services as the distributed storage provider. To perform the resource discovery task, each CLEVER domain stores the state of all available federated domains in a NoSQL database. When a CLEVER domain requires external processing resources, pieces of information about the available federated domains are retrieved from the database. To split the transcoding work among the available federated CPs, a Cloud Broker is implemented. Experimental results performed on “Cloud Federation for Transcoding” show that horizontal federation makes the whole transcoding process more scalable and efficient. Moreover, it is claimed that the horizontal parallelization introduced by the Cloud federation is better than the vertical one provided by the single CP domain Hadoop cluster.

4.2.3 Single virtual machines

Virtual machines and cloud computing are closely related technologies that are often used together but are not the same. Cloud computing is a model of computing in which computing resources (virtualized or not) are delivered over the Internet on a pay-per-use basis. Cloud computing services can include storage, networking, computing, and other services that are provided over the Internet. The following papers are based on VMs that can be used in a cloud scenario, but they are not tested on it. They are selected for this review because they show interesting proposals related to the video encoding topic, which could be implemented in a real cloud environment.

The authors of [79] proposed a method to minimize the use of cloud computing resources for video transcoding by tuning the transcoding configurations. They used the Lyapunov optimization framework to minimize the output bit rate while maintaining queue stability. The bit rates for each configuration were estimated through a model trained offline with historical data. They tested their approach with two 1080p video streams averaging 10 minutes and three presets of the H.264 codec: superfast, faster, and medium. Each job was dispatched to a virtual machine to be processed, and the load balancer dispatched jobs among different VMs to balance the load. The simulations showed that their approach achieved a similar output bit rate with only about 5 seconds delay compared to the static method with the 'medium' preset, which suffered a longer delay of 142 seconds for a 5.47 Mbps bit rate when the system was heavily loaded.

A new method to accelerate the game computing process in the cloud providing additional information to the video encoder is presented on [80]. The proposed method is based on the RR-GaaS category, which offers advantages to both developers and players by performing server-side tasks such as executing game logic, updating and rendering scenes, and encoding and streaming video. The proposed method is an intelligent acceleration system that uses information from the game engine to adjust the encoding parameters. It is also generic and adaptable to game engines. For video encoding, the H.264 standard reference software is used. The visual quality of the proposed method is evaluated by performing objective tests and subjective evaluations, showing up to 24% acceleration for the whole encoding process and 39% for the motion estimation operation concerning their previous proposal. The subjective quality tests (using MOS) show that the proposed method is only 17.55% away from conventional encoding and 1.7 times more than the previously proposed method. The PSNR and MOS results confirm an improvement in the quality and performance of the proposed method.

4.3 Encoding proposals based on containers

4.3.1 Serverless

Serverless architectures can be used to perform video encoding/transcoding tasks in a cost-effective and scalable way. In a serverless architecture, tasks are typically performed by small, independent units of code (called "functions") that are executed in response to events (such as the upload of a new video file).

An initial proposal following this paradigm was developed by Fouladi et al. [81]. They propose the Mu framework intended for fine-grained parallelism, using a functional programming style deployed on AWS Lambda. Thus, instead of invoking workers in response to a single event, they invoke them in bulk, thousands at a time. In this approach, the VP8 encoder (the only one used), has to be modified to reencode consecutive chunks of 6 frames (this is a particular election made by the authors) and then perform the "rebase" phase (running serially) to compound the chunks in the final encoded video sequence. This implementation could be improved by increasing the chunks' size (e.g., chunks of 2 seconds or 48 frames) to avoid communication among workers ("rebase" phase) and increasing the reported performance significantly (more than 8 times faster).

A little later, a serverless video processing framework appeared. The authors design a solution called Sprocket [82], that uses a serverless framework for video processing in the cloud. Sprocket framework manages the access, upload, encoding, decoding, and hosting of processed video and images in a low-latency, low-cost, fully parallelized environment. In addition, it integrates image processing and computer vision applications. Developers can use all these implementations to build their processing pipelines or extend functionalities. Applications such as video filters, can take advantage of massive container-based parallelism to decode, filter, and encode each segment of a video. As for the processed files, they are stored and managed by S3. To apply a filter (e.g. color change) and encode the video, the FFmpeg tool is used. The solution deployed under the Amazon Lambda infrastructure was evaluated with 1000 instances (workers); where each second of video is processed by one instance. In the evaluation, the task that uses the most computation time is encoding with 34.7%, followed by decoding with 30.4%. The use of this serverless service to deploy applications allows minimal startup delay, high parallelism, low latency, and low cost.

4.3.2 Others

Microservices is a new model of building applications, which involves breaking down a complex application into a set of small, independent, modular components called microservices that can be developed, deployed, and scaled independently. Microservices, containers, and Kubernetes can be used together to build a scalable and reliable video encoding platform. In this scenario, the microservices that make up the video encoding platform are packaged into containers and deployed and managed by the Kubernetes platform.

Following the microservices model, and using containers and Kubernetes, there are several works, which improve the video encoding systems to give better performance. With this premise, the authors of [83] present Morph: a video transcoding system in the cloud to maximize the use of resources. The proposed method considers task scheduling and resource provisioning as a stochastic optimization problem on two-time scales. The architecture, composed of several modules, presents an interface to manage transcoding requests, a deployment of homogeneous instances for the transcoding process, a task scheduler in charge of distributing the tasks according to the Value-based task management (VBS) policy, and the resource management policy. This last one determines the number of VM instances to be shut down or activated at the beginning of each task according to the state of the system in the slow and fast time scale. To determine the selection of actions of this policy we take advantage of the Q-Learning method, which is a model-free reinforcement learning technique. The system developed in Python implements a Docker-based cloud environment and uses FFmpeg for transcoding. The tests performed show that in the transcoding time estimation, the normalized prediction error is within the range of -0.08 to 0.08. The learning-based resource provisioning policy and the value-based task scheduling policy (LRP-VBS) show that they can perform effectively for the three management policies considered.

To reduce the economic cost of the transcoding tasks, [84] presents a micro-services-based transcoding platform hosted in an ad-hoc server farm on a Rasberry Pi 2 Rack (RPi) implementing Docker. The inherent cloud transcoding tasks performed, such as segmentation, concatenation, task scheduling, transcoding, and content streaming, are handled as isolated micro-services in lightweight containers. This service model allows for scalability and continuous development. Moreover, as each microservice is encapsulated in an image managed by Docker Swarm, they can be automatically started on the 16 available servers to respond to clients when they request a movie. For transcoding, the FFmpeg tool (H.264) and the power of both the CPU and GPU of each RPi are used. The evaluation tests performed are compared with a high-performance server, where it is shown that using the full power of the server farm, the transcoding time is 2.4 times faster than the workstation compared.

Another alternative for multimedia content processing using a container-based cloud encoding system is presented on [85]. The architecture, based on the server-worker model, has the following layers: a) From Layer: which allows supplying the original video content from the clients to the system; b) Service Layer: which contains most of the internal modules of the system (transcoding, message queue, cache, etc.); and c) Controller Service: which controls and supervises the workflow among the previous layers. The deployment of the solution uses containers with Docker. Each container (worker) uses FFmpeg to transcode videos using the H.265 codec. The worker tasks are managed by a messaging queue, called Rabbit MQ. HDFS is used for content storage to ensure redundancy. Redis is used to manage the cache and MySQL to store information from other services (monitoring). Several of these services run in separate containers on a master node. In this way, it also implements two transcoding containers to take advantage of the available GPU. The deployment of the solution in a private cloud has allowed them to test the efficiency, robustness, scalability, and portability with the use of containers compared to previous work in the literature.

In [86], the authors propose a heuristic solution to reduce the delay in video streams with DASH. To address the problem, they propose a centralized system, where a master server manages the requests between data servers in the cloud. The solution is implemented in a cloud system using Docker. The master server processes metrics (free memory, CPU usage, network status, etc.) from the data servers to manage transcoding requests. The proposed scheduling algorithm depends on the network state and computational load, and it can request transcoding processes or search for the transcoding sequence in the data servers with higher availability. According to the authors, this proposal improves by 30% the video visualization process for a FIFO type request scheduling.

Finally, the proposal designed in [87] presents an architecture that predicts the optimal virtual resources for live video transcoding in the cloud. Transcoding tasks are performed using Docker containers running on a Kubernetes platform. In this research, the authors have designed several models to predict transcoding speed and CPU consumption on resources. This implementation has been performed on virtual machines managed by the Eucalyptus cloud computing environment. They use nodes with FFmpeg for video transcoding and the statistics of each process are stored in Cassandra. Then, Prometheus software is in charge of collecting statistics on the resources used by these nodes. This information allows them to periodically train a predictive model based on the stored transcoding and CPU consumption measurements. The transcoding nodes act as workers in the Kubernetes cluster. The developed prototype experimented with various learning configurations. The results indicate that Random Forest (RF) regression achieved the best overall prediction performance compared to Reinforcement Learning (RL) or Stochastic Gradient Descend (SGD) regression in this particular case.

4.4 Encoding proposals based on the delivery

4.4.1 Edge computing

In edge computing, data processing, and storage are performed at the edge of the network, closer to the user [88]. It is often used in applications that require real-time processing, such as video streaming, autonomous vehicles, and industrial IoT (Internet of Things) applications. When edge computing architectures are used in video encoding systems, they permit to processing and transmission of video data efficiently from devices (such as cameras, sensors, and drones) to the cloud or other remote servers. This allows having lower latency, less network congestion, and better scalability and security.

The authors of [89] propose an optimization solution for video transcoding, caching, and bandwidth costs in adaptive streaming. They focus on orchestrating multiple resources in the video processing cloud to reduce the cost of adaptive low-demand video streaming. Their system architecture consists of three parties: content providers, users, and Media Cloud service providers. The optimization problem considers a three-way tradeoff between caching, transcoding, and bandwidth costs at each edge server. They designed models to manage the number of video segment representations and optimize the caching of frequently used segments on edge servers. The cost model developed shows that their solution is most effective for nodes far from the origin server or when bandwidth costs are relatively expensive. Performance tests indicate that the analytical results closely match the experimental results and that there is a clear trade-off between transcoding and bandwidth costs. Increasing cache space linearly increases storage costs but decreases combined bandwidth and transcoding cost. They report up to 50% savings in operating costs of adaptive video-on-demand transmission compared to the base case and existing methods.

The contribution [90] presents a new approach to video content encoding and distribution by using pro-active caching and collaborative processing of video fragments in edge networks. In the proposed model, the network will not fetch all segments when a video is requested, but pro-actively fetch probabilistically selected segments based on their popularity. This probabilistic fragment display model is used to minimize delay and increase QoE, due to the large bounce rates that a video may have. In addition, it avoids delivering a large amount of unviewed content to CDNs. Two policies (a Proactive Caching Policy and a Cache Replacement Policy) are introduced to store only the most popular video fragments in the cache, which can later be transcoded and distributed from different MEC servers. Simulation results show that the model and policies perform more than 20% better than other perimeter caching approaches in terms of cost, average delay, and cache hit ratio for different network configurations.

In the 5G mobile network context, [91] presents a strategy to deploy virtual CDN instances dynamically in a federated multi-domain edge cloud for video content delivery. In this case, the QoE-driven scheme uses automatic scaling to adjust to video transcoding requirements on the fly by adapting bit rates to network conditions. If quality degradation is detected for a user based on significant changes in modeled MOS values, a video transcoding service is triggered to start at the edge cloud. Once the target QoE level is reached, the virtual transcoder instance can be terminated and removed from the edge node to free up resources. An Ubuntu 14.04.03 LTS desktop workstation with 8 CPU cores and 16 GB of RAM was used to evaluate the proposal. The cloud environment was created using OpenStack (DevStack Juno version) inside an Ubuntu VirtualBox server with 4 dedicated vCPUs and 8 GB of RAM. The configuration consists of an all-in-one OpenStack environment with the controller, compute, Heat, and Neutron components running on the same node. The content for the initial transmission was pre-transcoded and prepared using FFmpeg, and the video codec used was H.264/AVC. Tests showed that the simulated network conditions led to timely downloads of video segments to adjust the MOS level. This minimized the playback buffering time, positively affecting the overall QoE by decreasing the quality adjustment time in transmission.

As we see, several solutions have been implemented to process video content on cloud-edge servers. However, they have not considered how to correctly schedule tasks when they have a high number of requests. For this reason, the paper [92] presents a multi-request scheduling and collaborative service processing method (MRSCP) for DASH in the cloud edge. The method considers collaborative cache management, request scheduling, and transcoding. Moreover, neighboring nodes cooperate to process each request through seven service models; the purpose is to utilize the low latency of edge computing and improve the user's viewing experience. In order to check the proposal, several simulation tests were executed on a service with DASH, where they had 10 servers and 100 videos. Their results indicate a caching utilization of 70% compared to 43% and 60% with the SingleEC (no neighbor nodes) and ColEC (neighbor nodes) methods respectively implemented by the literature.

4.4.2 Content delivery networks

Streaming video over the Internet heavily loads the underlying content delivery networks (CDNs). Faced with bandwidth and quality of service requirements, more and more video content providers are turning to content delivery networks. A CDN can help customers to get video content at a high speed while maintaining high quality in video streams. However, because video transcoding workloads are high, CDN providers must implement solutions based on an elastic and optimized cloud.

In paper [93], a transcoding system that leverages cloud computing and content delivery network infrastructure is designed and implemented. The proposal is called C^3. It has the following components: a) Ingesting Cloud, which provides fast response and increased capacity when uploading videos to be transcoded by the client; b) Transcoding Cloud, which manages dynamic provisioning of transcoding nodes to accommodate bursty workloads using an optimal transcoding node and a client's profile. These major components are coupled to try to get the maximum benefit in performance. Requests are served by the GeoDNS component (standard CDN component), which determines the optimal ingest server to load the content. After this process, the Transcoding Cloud component is invoked to perform the transcoding. C^3 also has a load balancer to handle job bursts and minimize transcoding time. If a video exceeds the storage size allowed by an ingestion server, nearby servers will be used for uploading and processing. The evaluation tests use Microsoft Expression Encoder 4 (EE4) to encode the videos into multiple representations. The prototype includes 7 ingest servers, 4 transcoding servers, and 6 destination streaming servers located in seven different cities. They consider a VoD scenario to quantify the savings in several performance metrics. The first test measures the ingestion performance, where it is shown that by increasing from 1 to 2 servers a 35% reduction of the ingestion time is achieved. However, when using more than 5 servers, the improvement is small. This is due to the overhead of splitting the tasks and grouping them after being processed.

Finally, another work where the authors show an encoding system based on CDN is [94]. This paper presents an extensive study on the development of a large-scale transcoding platform and proposes an architecture to provide CDN Slices as a Service across multiple management domains in the cloud, intending to optimize the efficient cost incurred in terms of delivery time and latency. This architecture integrates network functions virtualization (NFV) technology guided by network management and orchestration (MANO). Specifically in this work, several virtual network functions (VNFs), such as virtual transmitters and virtual transcoders, running in multi-administrative cloud domains are used. Transcoder servers are considered virtual network functions that can manage virtual machines instantiated in different cloud domains. The tests performed using the FFmpeg tool to evaluate the proposal, the parameters analyzed are: a) Inputs: number of arriving videos and video durations, and b) Environment: processing capacity of the hosting VM. When transcoding two videos on the VM-1core, it uses 95% of the total processing resources. However, a VM-1core decreases almost to less than 50% when transcoding many videos (8 in parallel), i.e. the overall system latency and response time depend on the virtual resources of the hosting machine. Furthermore, it is observed that the average processing time increases proportionally with the duration of the videos, this would allow predicting an estimated transcoding time for the workloads using machine learning.

4.5 Analysis of the papers according to the video codecs and delivery method

4.5.1 Classification based on video codecs

As we have seen in Section 2.1, several video codecs can be used in multimedia scenarios. Table 5 shows the video codec used by each work studied in this survey. As we can see, the most used codec in the proposals is the H.264/AVC codec. The paper that has used more codecs in its tests is [77], where the authors compare the behaviour of their proposal using H.265/HEVC, VP9, and AV1, in addition to it is the only one that has used the latest generation AV1 codec. Another conclusion we can draw from this table is that there are only 3 works that have used the VP9 codec. Many of the papers analyzed do not indicate which video codec has been used in their proposals, so the table indicates N/A (Not Available).

4.5.2 Classification based on methods of online video delivery

Currently, there are primarily two methods for online video delivery through DASH. VoD allows users to watch video content like movies or TV shows at their convenience, rather than adhering to a set broadcast schedule. VoD enables users to watch pre-recorded video content on their own schedules, often through subscription services or by purchasing or renting individual titles. Instead, live streaming involves the real-time transmission of video and audio content over the Internet. Accessible to anyone with an internet connection, live streams typically broadcast events as they unfold, covering various content such as sports events, concerts, news broadcasts, and more.

Although both systems have an infrastructure to carry out the tasks of encoding, serving, streaming, monitoring, etc. live streaming systems have some more strict requirements than VoD systems, for those reasons in this subsection, we are going to make a classification of the selected works depending on whether each system is intended to work in VoD, Live, or not specified. This information is reflected in Table 6.

VoD scenarios can deal with large amounts of multimedia content with loose time constraints, therefore cloud computing and the plethora of managed services they offer is well suited to deploy this type of streaming service.

If we pay attention to live environments, the proposed solutions must offer encoded content with an end-to-end delay of less than 15 seconds, which can take advantage of the high computing capacity and degree of parallelism of the cloud environment combined with edge computing.

4.6 Relevance of selected papers

In this subsection, we are going to analyze the relevance of each selected paper in this study. To analyze the relevance of each one, we will rely on the number of citations extracted from Google Scholar[5].

As we can see in Table 7, the selected papers have been grouped by year of publication. In the year 2011, only appears one publication, which has 5 references. In the next year, 2012,

[5] Google Scholar Website: https://scholar.google.com/

Table 5 Video codec specified in the model or tests of the studied works

Work	MPEG1/2	H.264/AVC	H.265/ HEVC	VP8	VP9	AV1	EE4
Li et al. [44]	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Jokhio et al. [45]	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Zheng et al. [46]	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Diaz-Sanchez et al. [47]	✓	N/A	N/A	N/A	N/A	N/A	N/A
Kim et al. [48]	✓	N/A	N/A	N/A	N/A	N/A	N/A
Kesavaraja and Shenbagavalli [49]	N/A	✓	N/A	N/A	N/A	N/A	N/A
Zakerinasab and Wang [95]	N/A	✓ /SVC	N/A	N/A	N/A	N/A	N/A
Huang et al. [51]	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Díaz-Sánchez et al. [52]	✓	✓ /SVC	N/A	N/A	N/A	N/A	N/A
Huang et al. [53]	N/A	✓	N/A	N/A	N/A	N/A	N/A
Jayasena et al. [54]	N/A	✓	N/A	N/A	N/A	N/A	N/A
Kesavaraja and Shenbagavalli [55]	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Sameti et al. [56]	N/A	N/A	✓	N/A	N/A	N/A	N/A
Barlas [57]	N/A	✓	N/A	N/A	N/A	N/A	N/A
Lin et al. [58]	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Wei et al. [59]	N/A	✓	N/A	N/A	N/A	N/A	N/A
Pang et al. [68]	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Ranganathan et al. [60]	N/A	✓	N/A	N/A	✓	N/A	N/A
Li et al. [61]	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Cheng et al. [62]	N/A	✓	N/A	N/A	N/A	N/A	N/A
Wang et al. [63]	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Farhad et al. [64]	✓	✓	N/A	N/A	N/A	N/A	N/A
Wu et al. [66]	N/A	✓	N/A	N/A	N/A	N/A	N/A
Li et al. [67]	N/A	✓	N/A	N/A	N/A	N/A	N/A
Jiang et al. [69]	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Zabrovskiy et al. [70]	N/A	✓	N/A	N/A	N/A	N/A	N/A
Lai et al. [71]	N/A	✓ /SVC	N/A	N/A	N/A	N/A	N/A
Ran et al. [72]	N/A	✓	N/A	N/A	N/A	N/A	N/A
Gao and Wen [73]	N/A	✓	N/A	N/A	N/A	N/A	N/A
Hegazy et al. [74]	N/A	N/A	✓	N/A	N/A	N/A	N/A
Kim et al. [75]	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Gutiérrez-Aguado et al. [76]	N/A	N/A	✓	N/A	✓	N/A	N/A
Gutiérrez-Aguado et al. [77]	N/A	N/A	✓	N/A	✓	✓	N/A
Panarello et al. [78]	N/A	✓	N/A	N/A	N/A	N/A	N/A
Yang et al. [79]	N/A	✓	N/A	N/A	N/A	N/A	N/A
Semsarzadeh et al. [80]	N/A	✓	N/A	N/A	N/A	N/A	N/A
Fouladi et al. [81]	N/A	N/A	N/A	✓	N/A	N/A	N/A
Ao et al. [82]	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Table 5 continued

Work	MPEG1/2	H.264/AVC	H.265/HEVC	VP8	VP9	AV1	EE4
Gao et al. [83]	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Barais et al. [84]	N/A	✓	N/A	N/A	N/A	N/A	N/A
Dong et al. [85]	N/A	N/A	✓	N/A	N/A	N/A	N/A
Van Ma et al. [86]	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Pääkkönen et al. [87]	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Jin et al. [89]	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Baccour et al. [90]	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Taleb et al. [91]	N/A	✓	N/A	N/A	N/A	N/A	N/A
Zhao et al. [92]	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Zhuang and Guo [93]	N/A	N/A	N/A	N/A	N/A	N/A	✓
Benkacem [94]	N/A	N/A	N/A	N/A	N/A	N/A	N/A

there are four papers with 174 references. From the four papers published that year, paper [44] has 125 references, a high number due to the level of specialization of the topic of this review. In 2013 were published three papers, which have been referenced by 227 works. In 2014, the same number of papers were published as in 2013, but their impact was lower, as these papers only obtained 70 references.

In 2015 and 2016, 6 and 8 papers were published, respectively. The number of references to the papers published in 2015 is 134, while in 2016 it is 119. The publications of 2017 have obtained the highest number of references (444). [81] with 274 references, is the most referenced paper of this review, doubling its relevance concerning the second most relevant [82], published in 2018. In that year, 2018, 5 papers were published, which have 174 references.

Finally, six papers were published in 2019 and another six in 2020. The papers of 2019 have 58 references and the papers of 2020 have 36. In 2021 and 2022 only one paper has been published each year, obtaining 16 references for [60] and 5 for the paper [70]. The total number of citations of the 49 papers selected for this review exceeds 1460 references. This is indicative of the importance of the subject for the scientific world. Moreover, with the new paradigms associated with the cloud and multimedia systems that will be implemented in the coming years (see next section), these indicators will certainly improve.

Table 6 Classification of works depending on delivery methods

Type of streaming	Works
VoD (associated to bandwidth)	[44, 47–49, 56, 57, 95], [58, 61, 63, 67, 70, 73, 75], [76–78, 83, 84, 86, 89], [90, 92–94]
Live (associated to delay and jitter)	[45, 51–53, 55, 59, 68], [60, 62, 64, 66, 69, 74, 79], [80–82, 87, 91]
Not specified	[46, 54, 71, 72, 85]

Table 7 Number of citations per year

Year	Publication	Number of citations	Total cites by year
2011	[46]	5	5
2012	[47]	5	174
	[57]	26	
	[44]	125	
	[93]	18	
2013	[45]	107	227
	[58]	32	
	[71]	88	
2014	[48]	25	70
	[62]	32	
	[72]	13	
2015	[49]	2	134
	[95]	18	
	[51]	3	
	[79]	5	
	[80]	21	
	[89]	85	
2016	[52]	4	119
	[53]	4	
	[61]	51	
	[63]	11	
	[64]	6	
	[73]	19	
	[83]	12	
	[84]	12	
2017	[54]	25	444
	[59]	66	
	[66]	9	
	[67]	70	
	[81]	274	
2018	[55]	7	174
	[56]	7	
	[94]	17	
	[85]	7	
	[82]	136	
2019	[68]	1	58
	[69]	4	
	[74]	11	
	[91]	24	
	[86]	8	
	[87]	10	

Table 7 continued

Year	Publication	Number of citations	Total cites by year
2020	[75]	4	36
	[76]	0	
	[77]	3	
	[78]	5	
	[90]	23	
	[92]	1	
2021	[60]	16	16
2022	[70]	5	5
	Total citations between 2011-2022		1462

5 Challenges and open research issues

The vast majority of video codecs and cloud/edge architectures are optimized for VoD and are mainly focused on bandwidth optimization by design. It makes the state of the art with a significant gap which is directly translated to a big challenge: “to create video codecs with the possibility to optimize simultaneously for multi-dimensional factors such as latency, jitter, packet losses, aspect ratio, frame size, etc”. This multi-dimensional challenge can be materialized in a set of concrete challenges that need to be addressed.

- In the latency domain, it is required to provide ultra-low delay video codecs by design rather than creating extensions of codecs that have been optimized for frame size and bandwidth before. There are in the literature available some good attempts such as the low-latency profiles from all the video codecs analyzed. However, we are referring not to extensions over the existing video codecs but creating new ones with that target in mind to achieve the target of the sub 10 ms glass-to-glass latency. It will open the usage of video in real-time machine-to-machine communications which is currently almost impossible nowadays, at least with this sub 10ms target.
- In the complexity domain, it is required to provide ultra-simple enconder/decoder by design to truly allow video processing capabilities in tiny IoT devices (less than 5 EUR/USD/GBP/CNY). The current design of video codec is so complex that certainly are impossible to be run without the help of hardware acceleration and that hardware acceleration is currently very complex and expensive to be embedded in IoT chips. As a consequence, there is not any single true IoT microcontroller, or processor with support for them. It is important to emphasize that we are referring to true IoT devices defined as $>$ 1 Watt consumption and not micro-computers such as Raspberry Pi or similar (around 4 Watts currently). By achieving so, we will allow the scaling up of the devices and use cases provided in the “capture zone” previously described in Fig. 3. It will eventually open up the usage of video encoding/transmission/decoding in use cases where a low-cost device is deployed for 10 years of life expectancy and it can transmit video.
- In the power consumption domain, it is required new video codecs truly optimized for energy consumption, able to send devices to sleep between frames or sets of frames, and able to perform the computations of the algorithms required to predict the motion of the video codec while some parts of the microprocessor, board are switched off.

- In the bandwidth domain, there is yet room for improvement, specially with the new venue of holopresence and other 360 immersive technologies that require true real-time experiences. This implies frame resolutions never required before such as 16K or 32K and frame rates significantly higher than traditional screens such as 100 FPS, 150 FPS, or 200 FPS, that require high bitrate. Independently of the network conditions, to achieve an adequate level of QoE with this high volume content, field-of-view aware encoding must be developed.
- In the jitter domain, there is almost no investigation found in this area. It is going to be a significant aspect to consider with the venue of the haptics communications where gloves or other devices can transmit the tactile sense remotely. When it is convenient with video feeds, the synchronization between video and haptic information is in the range of sub 1ms jitter to do deliver the feeling remotely to the final user. Thus, this is imposing the need to investigate novel algorithms and architectures to deal with this aspect.
- In the scientific and industrial domain, there is also a significant challenge to be addressed. The number of channels supported by the video coded. Currently, all the existing video codecs in the market have been optimized for chroma and luminance and they are all using different channels focused on the visible spectrum. However, there is a clear need to investigate video codecs that allow encoding, transmitting, decoding, and managing multiple channels beyond the visible spectrum such as those coming from the hyperspectral and multispectral imagery where dozens or hundreds of different channels need to be delivered to create a 3D spectral cube of the sensed information. Thus, this kind of video codec needs to be dealing with the simultaneous optimization of the quality of the information delivered and the quality of experience received. This duality imposes a big challenge that is worth to be investigated and currently is an open research question.

Concerning the architectures of the cloud and edge computing used to deliver video traffic, we have seen that there are already very good examples of them but it is also true that there is significant room for improvement in different aspects such as:

- The integration of the novel Edge computing architecture with the optimization for low latency was previously indicated as a challenge to be able to truly deliver any optimization in latency achieved by video codec until the last mile of the users.
- The current trend is to provide video transmission protocols that are completely opaque for the network thus not allowing in-network optimizations. This has been a trend mainly fostered by the enhancement of privacy and security in video delivery. However, there is room to create better video transmission protocols that allow exposure to the network non-privacy or security-sensitive metadata that can be used to perform several video optimizations along the way, i.e. in the processing, storage, and delivery zones previously described. Such metadata would allow us to perform smart optimizations in video delivery by allowing the control plane of the network to be aware of such information.
- The usage of the cloud has also brought a significant challenge that is related to the fact on the decision on where the encoding of the video is carried out. In Fig. 3, we can see how the video is captured in the capture zone and sent to the processing zone. The challenge currently is that there is a complete lack of protocols to efficiently send such non-encoded raw video to the cloud and such a challenge is even more exacerbated when we are trying to optimize the use case for dimensions such as latency and jitter.
- Another relevant aspect is multicast, that can relieve packet replication when sending broadcast traffic to multiple customers. This is simply not supported in any of the current cloud and edge computing architectures, but if achieved, it could alleviate significantly the global traffic used for multimedia delivery at planetary scale.

- The current state of the art in architecture is vastly dominated by the usage of traditional cloud infrastructure and the novel edge infrastructure but there is a significant lack of research done in the area of exploiting serverless architectures where the cloud-native functions for encoding, decoding, and transmission of multimedia are only activated in an event-driven approach instead of the traditional always-on pay-as-you-go model. It would significantly produce an alleviation of resources consumed to perform cloud multimedia operations with significant benefit in terms of both operational and capital costs of infrastructures and also of development and the power saving associated with the selective activation of the functions.
- There is also significant room for improvement in the architecture for Video on Demand in the management of CDNs, which are currently optimized using geo-DNS and similar approaches to make sure we perform a sub-optimal caching scheme. Network discovery mechanisms are required in the cloud and edge infrastructures to allow the CDN to have a complete understanding of the network and thus provide such optimal allocations of resources to reduce cache missing to virtually 0 percent.
- The cloud/edge architectures are also directly associated with the usage of web browsers to receive the content. A significant step forward could be achieved by natively empowering web browsers with the streaming protocols of the future. Thus streaming protocols should be aware not only of the network status but also, of the CDN status, on the network topology, and other aspects to allow the system to predict the intention of the user concerning the video playback.
- All the video streaming protocols available in the literature are exclusively reactive, they deliver video but are not aware of the optimizations by the video codecs and thus cannot optimize such video delivery. For example, performing dynamic GOP negotiation, dynamic MTU negotiation, and other aspects to fine-tune the video delivery and prevent network package fragmentation.
- Cloud and edge architectures nowadays are starting to provide basic QoS capabilities. However, there is no attempt to make video streaming aware of such QoS capabilities. An open issue is adapting video streaming protocols to send signals directly to the cloud and edge infrastructure to dynamically take care of QoS requirements according to the requirements of the video.
- Also, concerning QoS, the cloud and edge computing are starting to investigate the introduction of novel network slices and being a logically isolated circuit network over the cloud that allows to enforcement of QoS and SLAs.

6 Discussion and conclusion

6.1 Discussion

In the last few years, video streaming has become increasingly important on the Internet and the multimedia sector is an elemental aspect in any sector of the business world. Cloud-based video encoding is a good option for businesses because it is scalable, easy to use, and has low capital and operational costs. Cloud-based video encoding also offers the flexibility to encode video using a wide range of codecs and formats. For these reasons and others that have been discussed throughout this study, this paper makes a thorough review of cloud-based video encoding and the papers related to this topic.

While we have identified 49 articles on the topic, we believe that the scientific community has made relatively few contributions in recent years (2021 and 2022). This is difficult to explain, particularly considering the numerous challenges outlined in the Section 5 that warrant improvement in cloud video encoding processes. For instance, when encoding for multi-resolution transmission of live events with 8K and 16K resolutions, a cloud-based video encoding system with exceptionally high processing capabilities is essential to attain low-latency scenarios. This becomes even more critical in real-time systems where glass-to-glass times are minimized to sub-second intervals. This difficulty is reflected in the data obtained, where more than half of the selected works are focused on VoD streaming.

If we focus on the solutions proposed by the scientific community, we observe that the majority of the works employ the H.264 codec. However, there are very few instances where the VP9 and AV1 codecs are utilized. Concerning VP9, it is already widely employed in the industry; for instance, YouTube extensively uses it on its platform. Despite this, the analyzed works show that few researchers choose to use it. Admittedly, VP9 is a more computationally complex codec than H.264, but it offers higher compression capabilities. Similar considerations apply to AV1, where the limited adoption is attributed to its implementation occurring toward the end of 2018. The low variability in codec usage might be considered a weakness in many of the works reviewed in this analysis.

Another aspect highlighted in the study is the utilization of cloud technologies across different time intervals. Initially, the focus was heavily on the use of Hadoop and MapReduce for encoding tasks distributed among various workers. Subsequently, the scientific community shifted towards employing cloud systems, including public clouds such as Amazon AWS or Microsoft Azure, as well as private clouds-many of which were implemented on OpenStack. The latest work is focused on architectures where Docker and Kubernetes are used to perform coding tasks on serverless environments.

Another aspect that has not been addressed in the selected articles is the utilization of the cloud for coding tasks employing AI techniques. For instance, facilitating real-time generation of metadata from videos, identification of objects, places, and actions, and development of custom models for classifying and tracking objects in videos. Furthermore, AI can contribute to video transcoding, detection of inappropriate content, creation of content recommendation engines, and automatic tagging of videos for ad insertion, among other applications. All these tasks are computationally intensive, and migrating them to the cloud can enhance their performance.

In summary, there are various review papers on the subject of multimedia cloud computing; however, none of them specifically delve into the encoding and transcoding processes in the cloud. These processes demand substantial computational capacity, a requirement met by migrating to the cloud to mitigate costs. This review paper uniquely focuses on this specific topic, as there is no comparable work. It addresses a research area marked by novel enhancements in end-user quality of experience and the optimization of resource consumption to curtail the expenses associated with these tasks.

Conclusively, upon completing this review, we can effectively address the inquiries outlined in Table 2. The findings indicate a considerable number of works focusing on video coding within cloud environments. These works have been categorized based on the employed architecture or paradigm. Additionally, we have discerned the codecs utilized in each study and determined whether the proposed solutions were tailored for VoD or live environments. Lastly, we have pinpointed emerging challenges aimed at enhancing encoding systems in both the cloud and MEC.

6.2 Conclusion

In this review paper, we have presented the evolution of streaming with a special focus on the latest streaming methods and the importance of encoding processes. We have analyzed the importance of cloud systems in multimedia environments and a cloud infrastructure for media scenarios has been detailed. Moreover, it entails an assessment of previous reviews and surveys about the subject under consideration. The findings indicate an absence of comparable scholarly contributions to the one expounded in this paper.

This study has employed the systematic literature review (SLR) technique developed by Kitchenham and Charters [35]. Through this methodology, we have identified the need for this review

After that, some information about the selected papers has been introduced. We have also grouped the works into 3 big groups; according to the type of virtualization technology (VMs vs. Containers), and delivery on the cloud (see Fig. 8).

Moreover, each paper has been analyzed to detect if its proposal would be appropriate for live or VoD streaming, and what video codecs were used in the proposals. Lastly, we have analyzed the relevance of the selected papers according to the received cites. In this study, the most relevant papers are: [44, 81, 82], and [45] with more than 100 cites each. Finally, this review concludes presenting challenges and open research issues related to video encoding on the cloud.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature. Grant by PID2021-126209OB-I00 funded by MCIN/AEI/10.13039/501100011033 and by "ERDF A way of making Europe", by the "European Union" The research stay abroad of the author Miguel Garcia-Pineda at the University West of Scotland was funded by the grant BEST/2021/178 - Generalitat Valenciana.

Declarations

Competing interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Ahrendt D, Cabrita J, Clerici E, Hurley J, Leončikas T, Mascherini M, Riso S, Sándor E (2020) Living, working and covid-19. Technical report, Eurofound
2. Sandvine (2022) 2022 global internet phenomena report. Technical report, Sandvine. https://www.sandvine.com/global-internet-phenomena-report-2022
3. Tang H, Liu J, Yan S, Yan R, Li Z, Tang J (2023) M3net: multi-view encoding, matching, and fusion for few-shot fine-grained action recognition. In: Proceedings of the 31st ACM International Conference on Multimedia. MM '23, pp 1719–1728. Association for Computing Machinery, New York, USA. https://doi.org/10.1145/3581783.3612221
4. Zhu W, Luo C, Wang J, Li S (2011) Multimedia cloud computing. IEEE Sig Process Mag 28(3):59–69. https://doi.org/10.1109/MSP.2011.940269

5. He J, Wen Y, Huang J, Wu D (2014) On the cost-qoe tradeoff for cloud-based video streaming under amazon ec2's pricing models. IEEE Trans Circuits Syst Video Technol 24(4):669–680. https://doi.org/10.1109/TCSVT.2013.2283430
6. Attaran M, Woods J (2019) Cloud computing technology: improving small business performance using the internet. J Small Bus Entrep 31(6):495–519. https://doi.org/10.1080/08276331.2018.1466850
7. Schulzrinne H, Casner S, Frederick R, Jacobson V (2003) RFC 3550: RTP: A transport protocol for real-time applications. IETF
8. Schulzrinne H, Rao A, Lanphier R, Westerlund M, Stiemerling M (2016) RFC 7826: real-time streaming protocol version 2.0. IETF
9. Lei X, Jiang X, Wang C: Design and implementation of streaming media processing software based on rtmp. In: 2012 5th international congress on image and signal processing. pp 192–196 (2012). https://doi.org/10.1109/CISP.2012.6469981
10. Ruether T (2021) What is webrtc? Technical report
11. Blum N, Lachapelle S, Alvestrand H (2021) Webrtc: real-time communication for the open web platform. Commun ACM 64(8):50–54
12. Petrangeli S, Pauwels D, Van Der Hooft J, Žiak M, Slowack J, Wauters T, De Turck F (2019) A scalable webrtc-based framework for remote video collaboration applications. Multimed Tools Appl 78(6):7419–7452
13. Kua J, Armitage G, Branch P (2017) A survey of rate adaptation techniques for dynamic adaptive streaming over http. IEEE Commun Surv Tutor 19(3):1842–1866. https://doi.org/10.1109/COMST.2017.2685630
14. Thang TC, Ho Q-D, Kang JW, Pham AT (2012) Adaptive streaming of audiovisual content using mpeg dash. IEEE Trans Consum Elect 58(1):78–85. https://doi.org/10.1109/TCE.2012.6170058
15. Seufert M, Egger S, Slanina M, Zinner T, Hoßfeld T, Tran-Gia P (2015) A survey on quality of experience of http adaptive streaming. IEEE Commun Surv Tutor 17(1):469–492. https://doi.org/10.1109/COMST.2014.2360940
16. Kalva H (2006) The h.264 video coding standard. IEEE MultiMedia 13(4): 86–90 https://doi.org/10.1109/MMUL.2006.93
17. Li Z.-N, Drew M.S, Liu J (2014) New Video Coding Standards: H.264 and H.265. Springer, Cham. pp 395–434. https://doi.org/10.1007/978-3-319-05290-8_12
18. Mukherjee D, Bankoski J, Grange A, Han J, Koleszar J, Wilkins P, Xu Y, Bultje R (2013) The latest open-source video codec vp9 - an overview and preliminary results. In: 2013 Picture Coding Symposium (PCS). pp 390–393 https://doi.org/10.1109/PCS.2013.6737765
19. Han J, Li B, Mukherjee D, Chiang C-H, Grange A, Chen C, Su H, Parker S, Deng S, Joshi U, Chen Y, Wang Y, Wilkins P, Xu Y, Bankoski J (2021) A technical overview of av1. Proc IEEE 109(9):1435–1462. https://doi.org/10.1109/JPROC.2021.3058584
20. Chen Y, Murherjee D, Han J, Grange A, Xu Y, Liu Z, Parker S, Chen C, Su H, Joshi U, Chiang C.-H, Wang Y, Wilkins P, Bankoski J, Trudeau L, Egge N, Valin J.-M, Davies T, Midtskogen S, Norkin A, Rivaz P (2018) An overview of core coding tools in the av1 video codec. In: 2018 Picture Coding Symposium (PCS). pp 41– 45 https://doi.org/10.1109/PCS.2018.8456249
21. Bross B, Wang Y-K, Ye Y, Liu S, Chen J, Sullivan GJ, Ohm J-R (2021) Overview of the versatile video coding (vvc) standard and its applications. IEEE Trans Circuits Syst Video Technol 31(10):3736–3764. https://doi.org/10.1109/TCSVT.2021.3101953
22. Battista S, Meardi G, Ferrara S, Ciccarelli L, Maurer F, Conti M, Orcioni S (2022) Overview of the low complexity enhancement video coding (lcevc) standard. IEEE Trans Circuits Syst Video Technol 1. https://doi.org/10.1109/TCSVT.2022.3182793
23. Grois D, Nguyen T, Marpe D (2016) Coding efficiency comparison of av1/vp9, h.265/mpeg-hevc, and h.264/mpeg-avc encoders. In: 2016 Picture Coding Symposium (PCS). pp 1–5 https://doi.org/10.1109/PCS.2016.7906321
24. García-Lucas D, Cebrián-Márquez G, Cuenca P (2020) Rate-distortion/complexity analysis of hevc, vvc and av1 video codecs. Multimed Tools Appl 79(39):29621–29638. https://doi.org/10.1007/s11042-020-09453-w
25. Al-hammouri M, Madani B, Aloqaily M, Ridhawi I.A, Jararweh Y (2018) Scalable video streaming for real-time multimedia applications over dds middleware for future internet architecture. In: 2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA). pp 1–6 https://doi.org/10.1109/AICCSA.2018.8612848
26. Escolar AM, Alcaraz-Calero JM, Salva-Garcia P, Bernabe JB, Wang Q (2021) Adaptive network slicing in multi-tenant 5g iot networks. IEEE Access 9:14048–14069
27. Anton M, Avinash D, Devang S, Murthy P (2021) Production media management: transforming media workflows by leveraging the cloud. Technical report, Netflix

28. Xu Y, Mao S (2013) A survey of mobile cloud computing for rich media applications. IEEE Wirel Commun 20(3):46–53. https://doi.org/10.1109/MWC.2013.6549282
29. Wen Y, Zhu X, Rodrigues JJPC, Chen CW (2014) Cloud mobile media: reflections and outlook. IEEE Trans Multimed 16(4):885–902. https://doi.org/10.1109/TMM.2014.2315596
30. Huang C-T, Qin Z, Kuo C-CJ (2011) Multimedia storage security in cloud computing: an overview. In: 2011 IEEE 13th international workshop on multimedia signal processing. pp 1–6 https://doi.org/10.1109/MMSP.2011.6093775
31. Yang J, He S, Lin Y, Lv Z (2017) Multimedia cloud transmission and storage system based on internet of things. Multimed Tools Appl 76(17):17735–17750. https://doi.org/10.1007/s11042-015-2967-9
32. Tselios C, Tsolis G (2016) A survey on software tools and architectures for deploying multimedia-aware cloud applications. In: Karydis I, Sioutas S, Triantafillou P, Tsoumakos D (eds.) Algorithmic Aspects of Cloud Computing. Springer, Cham pp 168–180. https://doi.org/10.1007/s11042-015-2967-9
33. Toshniwal A, Rathore K.S, Dubey A, Dhasal P, Maheshwari R (2020) Media streaming in cloud with special reference to amazon web services: a comprehensive review. In: 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS). pp 368–372 https://doi.org/10.1109/ICICCS48265.2020.9121097
34. Abdallah M, Griwodz C, Chen K-T, Simon G, Wang P-C, Hsu C-H (2018) Delay-sensitive video computing in the cloud: a survey. ACM Trans Multimed Comput Commun Appl 14(3s). https://doi.org/10.1145/3212804
35. Kitchenham B, Charters S (2007) Guidelines for performing systematic literature reviews in software engineering. Technical report. Keele University and Durham University
36. Petersen K, Feldt R, Mujtaba S, Mattsson M: Systematic mapping studies in software engineering. In: 12th International Conference on Evaluation and Assessment in Software Engineering (EASE) 12. pp 1–10 (2008)
37. Banijamali A, Pakanen O-P, Kuvaja P, Oivo M (2020) Software architectures of the convergence of cloud computing and the internet of things: a systematic literature review. Inf Softw Technol 122:106271. https://doi.org/10.1016/j.infsof.2020.106271
38. Van Solingen R, Berghout EW (1999) The goal/question/metric method: a practical guide for quality improvement of software development. McGraw-Hill
39. Van Latum F, Van Solingen R, Oivo M, Hoisl B, Rombach D, Ruhe G (1998) Adopting gqm based measurement in an industrial environment. IEEE Softw 15(1):78–86. https://doi.org/10.1109/52.646887
40. Petersen K, Feldt R, Mujtaba S, Mattsson M (2008) Systematic Mapping Studies in Software Engineering. In: Proceedings of the 12th international conference on evaluation and assessment in software engineering. EASE'08. pp 68–77. BCS Learning & Development Ltd., Swindon, GBR
41. Wohlin C (2014) Guidelines for snowballing in systematic literature studies and a replication in software engineering. In: Proceedings of the 18th international conference on evaluation and assessment in software engineering. EASE '14. Association for Computing Machinery, New York, USA https://doi.org/10.1145/2601248.2601268
42. Kitchenham B, Brereton P (2013) A systematic review of systematic review process research in software engineering. Inf Softw Technol 55(12):2049–2075. https://doi.org/10.1016/j.infsof.2013.07.010
43. Singh VK (2021) Singh P, Karmakar M, Leta J, Mayr P: The journal coverage of web of science, scopus and dimensions: a comparative analysis. Scientometrics 126(6):5113–5142. https://doi.org/10.1007/s11192-021-03948-5
44. Li Z, Huang Y, Liu G, Wang F, Zhang Z.L, Dai Y (2012) Cloud transcoder: bridging the format and resolution gap between Internet videos and mobile devices. In: Proceedings of the international workshop on network and operating system support for digital audio and video. pp 33–38. ACM Press, New York, USA https://doi.org/10.1145/2229087.2229097, http://dl.acm.org/citation.cfm?doid=2229087.2229097
45. Jokhio F, Ashraf A, Lafond S, Porres I, Lilius J: Prediction-based dynamic resource allocation for video transcoding in cloud computing. In: 2013 21st Euromicro international conference on parallel, distributed, and network-based processing. IEEE, pp 254–261. (2013) https://doi.org/10.1109/PDP.2013.44, http://ieeexplore.ieee.org/document/6498561/
46. Zheng L, Tian L, Wu Y (2011) A rate control scheme for distributed high performance video encoding in cloud. In: 2011 International conference on cloud and service computing. IEEE, pp 131–133 https://doi.org/10.1109/CSC.2011.6138510, http://ieeexplore.ieee.org/document/6138510/
47. Diaz-Sanchez D, Marin-Lopez A, Almenarez F, Sanchez-Guerrero R, Arias P (2012) A distributed transcoding system for mobile video delivery. In: 2012 5th Joint IFIP Wireless and Mobile Networking Conference (WMNC). IEEE, pp 10–16. https://doi.org/10.1109/WMNC.2012.6416151, http://ieeexplore.ieee.org/document/6416151/

48. Kim M, Han S, Cui Y, Lee H, Cho H, Hwang S (2014) CloudDMSS: robust Hadoop-based multimedia streaming service architecture for a cloud computing environment. Clust Comput 17(3):605–628. https://doi.org/10.1007/s10586-014-0381-0
49. Kesavaraja D, Shenbagavalli A (2015) Hadoop scalable video transcoding technique in cloud environment. In: 2015 IEEE 9th International Conference on Intelligent Systems and Control (ISCO). IEEE, pp 1–6. https://doi.org/10.1109/ISCO.2015.7282276, http://ieeexplore.ieee.org/document/7282276/
50. Zakerinasab MR, Wang M (2015) Does chunk size matter in distributed video transcoding? In: 2015 IEEE 23rd International Symposium on Quality of Service (IWQoS). pp 69–70 https://doi.org/10.1109/IWQoS.2015.7404710
51. Huang J-C, Wu C-Y, Chen J-J (2015) On high efficient cloud video transcoding. In: 2015 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS). IEEE. pp 170–173 https://doi.org/10.1109/ISPACS.2015.7432759, http://ieeexplore.ieee.org/document/7432759/
52. Díaz-Sánchez D, Sánchez-Guerrero R, Arias P, Almenarez F, Marín A (2016) A distributed transcoding and content protection system. Telecommun Syst 61(1):59–76. https://doi.org/10.1007/s11235-014-9952-x
53. Huang C-C , Chen J-J, Tsai Y-H (2016) A dynamic and complexity aware cloud scheduling algorithm for video transcoding. In: 2016 IEEE International Conference on Multimedia and Expo Workshops (ICMEW). IEEE, pp 1–6 https://doi.org/10.1109/ICMEW.2016.7574743, http://ieeexplore.ieee.org/document/7574743/
54. Jayasena KPN, Li L, Xie Q (2017) Multi-modal multimedia big data analyzing architecture and resource allocation on cloud platform. Neurocomputing 253:135–143. https://doi.org/10.1016/j.neucom.2016.11.077
55. Kesavaraja D, Shenbagavalli A (2018) Framework for fast and efficient cloud video transcoding system using intelligent splitter and hadoop MapReduce. Wireless Pers Commun 102(3):2117–2132. https://doi.org/10.1007/s11277-018-5501-3
56. Sameti S, Wang M, Krishnamurthy D (2018) Stride: distributed video transcoding in spark. In: 2018 IEEE 37th International Performance Computing and Communications Conference (IPCCC). IEEE, pp 1–8 https://doi.org/10.1109/PCCC.2018.8711214 . https://ieeexplore.ieee.org/document/8711214/
57. Barlas G (2012) Cluster-based optimized parallel video transcoding. Parallel Comput 38(4–5):226–244. https://doi.org/10.1016/j.parco.2012.02.001
58. Lin S, Zhang X, Yu Q, Qi H, Ma S (2013) Parallelizing video transcoding with load balancing on cloud computing. In: 2013 IEEE International Symposium on Circuits and Systems (ISCAS2013). IEEE, pp 2864–2867 https://doi.org/10.1109/ISCAS.2013.6572476, http://ieeexplore.ieee.org/document/6572476/
59. Wei L, Cai J, Foh CH, He B (2017) QoS-aware resource allocation for video transcoding in clouds. IEEE Trans Circuits Syst Video Technol 27(1):49–61. https://doi.org/10.1109/TCSVT.2016.2589621
60. Ranganathan P, Stodolsky D, Calow J, Dorfman J, Guevara M, Smullen IV CW, Kuusela A, Balasubramanian R, Bhatia S, Chauhan P, Cheung A, Chong IS, Dasharathi N, Feng J, Fosco B, Foss S, Gelb B, Gwin SJ, Hase Y, He D-k, Ho CR, Huffman Jr RW, Indupalli E, Jayaram I, Kongetira P, Kyaw CM, Laursen A, Li Y, Lou F, Lucke KA, Maaninen JP, Macias R, Mahony M, Munday DA, Muroor S, Penukonda N, Perkins-Argueta E, Persaud D, Ramirez A, Rautio V-M, Ripley Y, Salek A, Sekar S, Sokolov SN, Springer R, Stark D, Tan M, Wachsler MS, Walton AC, Wickeraad DA, Wijaya A, Wu HK (2021) Warehouse-Scale Video Acceleration: Co-Design and Deployment in the Wild. In: Proceedings of the 26th ACM international conference on architectural support for programming languages and operating systems. ASPLOS 2021. Association for Computing Machinery, New York, USA. pp 600–615 https://doi.org/10.1145/3445814.3446723, https://doi.org/10.1145/3445814.3446723
61. Li X, Salehi M.A, Bayoumi M, Buyya R (2016) CVSS: a cost-efficient and QoS-aware video streaming using cloud services. In: 2016 16th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid). CCGRID '16, IEEE, pp 106–115 https://doi.org/10.1109/CCGrid.2016.49, http://ieeexplore.ieee.org/document/7515677/
62. Cheng R, Wu W, Lou Y, Chen Y (2014) A cloud-based transcoding framework for real-time mobile video conferencing system. In: 2014 2nd IEEE international conference on mobile cloud computing, services, and engineering. IEEE, pp 236–245. https://doi.org/10.1109/MobileCloud.2014.31, https://ieeexplore.ieee.org/document/6834967
63. Wang Y, Chen W-T, Wu H, Kokaram A, Schaeffer J (2016) A cloud-based large-scale distributed video analysis system. In: 2016 IEEE International Conference on Image Processing (ICIP). IEEE, pp 1499–1503. https://doi.org/10.1109/ICIP.2016.7532608 . http://ieeexplore.ieee.org/document/7532608/
64. Farhad SM, Bappi MSI, Ghosh A (2016) Dynamic resource provisioning for video transcoding in IaaS cloud. In: 2016 IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on

Data Science and Systems (HPCC/SmartCity/DSS). IEEE, pp 380–384. https://doi.org/10.1109/HPCC-SmartCity-DSS.2016.0061, http://ieeexplore.ieee.org/document/7828402/

65. Cherkasova L, Phaal P (2002) Session-based admission control: a mechanism for peak load management of commercial web sites. IEEE Trans Comput 51(6):669–685
66. Wu J, Cheng B, Yang Y, Wang M, Chen J (2017) Delay-aware quality optimization in cloud-assisted video streaming system. ACM Trans. Multimedia Comput. Commun Appl 14(1). https://doi.org/10.1145/3152116
67. Li X, Salehi MA, Bayoumi M, Tzeng N-F, Buyya R (2018) Cost-efficient and robust on-demand video transcoding using heterogeneous cloud services. IEEE Trans Parallel Distrib Syst 29(3):556–571. https://doi.org/10.1109/TPDS.2017.2766069
68. Pang Z, Sun L, Huang T, Wang Z, Yang S (2019) Towards QoS-aware cloud live transcoding: a deep reinforcement learning approach. In: 2019 IEEE International Conference on Multimedia and Expo (ICME). IEEE, pp 670–675. https://doi.org/10.1109/ICME.2019.00121, https://ieeexplore.ieee.org/document/8785022/
69. Jiang Q, Lee YC, Zomaya AY (2019) Scalable video transcoding in public clouds. In: 2019 19th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID). IEEE. pp 70–75 https://doi.org/10.1109/CCGRID.2019.00017, https://ieeexplore.ieee.org/document/8752872/
70. Zabrovskiy A, Agrawal P, Kashansky V, Kersche R, Timmerer C, Prodan R (2022) Fspot: fast and efficient video encoding workloads over amazon spot instances. Comput Mater Continua 71(3):5677–5697 https://doi.org/10.32604/cmc.2022.023630
71. Lai C-F, Wang H, Chao H-C, Nan G (2013) A network and device aware QoS approach for cloud-based mobile streaming. IEEE Trans Multimedia 15(4):747–757. https://doi.org/10.1109/TMM.2013.2240270
72. Ran Y, Shi Y, Yang E, Chen S, Yang J (2014) Dynamic resource allocation for video transcoding with QoS guaranteeing in cloud-based DASH system. In: 2014 IEEE Globecom Workshops (GC Wkshps). IEEE, pp 144–149 https://doi.org/10.1109/GLOCOMW.2014.7063421, http://ieeexplore.ieee.org/document/7063421/
73. Gao G, Wen Y (2016) Morph: a fast and scalable cloud transcoding system. In: Proceedings of the 24th ACM international conference on multimedia. MM '16, Association for Computing Machinery, New York, USA. pp 1160–1163 https://doi.org/10.1145/2964284.2973792
74. Hegazy M, Diab K, Saeedi M, Ivanovic B, Amer I, Liu Y, Sines G, Hefeeda M (2019) Content-aware video encoding for cloud gaming. In: Proceedings of the 10th ACM multimedia systems conference. MMSys '19, Association for Computing Machinery, New York, USA. pp 60–73 https://doi.org/10.1145/3304109.3306222
75. Kim H-W, Mu H, Park JH, Sangaiah AK, Jeong Y-S (2020) Video transcoding scheme of multimedia data-hiding for multiform resources based on intra-cloud. J Ambient Intell Humaniz Comput 11(5):1809–1819. https://doi.org/10.1007/s12652-019-01279-1
76. Gutiérrez-Aguado J, Peña-Ortiz R, Garcia-Pineda M, Claver JM (2020) A cloud-based distributed architecture to accelerate video encoders. Appl Sci 10(15):5070. https://doi.org/10.3390/app10155070
77. Gutiérrez-Aguado J, Peña-Ortiz R, García-Pineda M, Claver JM (2020) Cloud-based elastic architecture for distributed video encoding: Evaluating H.265, VP9, and AV1. J Netw Comput Appl 171. https://doi.org/10.1016/j.jnca.2020.102782
78. Panarello A, Celesti A, Fazio M, Puliafito A, Villari M (2020) A big video data transcoding service for social media over federated clouds. Multimed Tools Appl 79(13–14):9037–9061. https://doi.org/10.1007/s11042-019-07786-9
79. Yang M, Cai J, Zhang W, Wen Y, Foh CH (2015) Adaptive configuration of cloud video transcoding. In: 2015 IEEE International Symposium on Circuits and Systems (ISCAS), vol 2015-July. IEEE, pp 1658–1661 https://doi.org/10.1109/ISCAS.2015.7168969, https://ieeexplore.ieee.org/document/7168969
80. Semsarzadeh M, Yassine A, Shirmohammadi S (2015) Video encoding acceleration in cloud gaming. IEEE Trans Circuits Syst Video Technol 25(12):1975–1987. https://doi.org/10.1109/TCSVT.2015.2452778
81. Fouladi S, Wahby RS, Shacklett B, Balasubramaniam KV, Zeng W, Bhalerao R, Sivaraman A, Porter G, Winstein K (2017) Encoding, fast and slow: low-latency video processing using thousands of tiny threads. In: 14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17). pp 363–376
82. Ao L, Izhikevich L, Voelker GM, Porter G (2018) Sprocket: a serverless video processing framework. In: Proceedings of the ACM Symposium on Cloud Computing. SoCC '18, Association for Computing Machinery, New York, USA. pp 263–274. https://doi.org/10.1145/3267809.3267815
83. G Gao, Wen Y, Westphal C (2016) Resource provisioning and profit maximization for transcoding in Information Centric Networking. In: 2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). IEEE, pp 97–102. https://doi.org/10.1109/INFCOMW.2016.7562053, http://ieeexplore.ieee.org/document/7562053/

84. Barais O, Bourcier J, Bromberg Y.-D, Dion C (2016) Towards microservices architecture to transcode videos in the large at low costs. In: 2016 International Conference on Telecommunications and Multimedia (TEMU). IEEE, pp 1–6 https://doi.org/10.1109/TEMU.2016.7551918, http://ieeexplore.ieee.org/document/7551918/
85. Dong Y, Zhang X, Zhao Y, Song L (2018) A containerized media cloud for video transcoding service. In: 2018 IEEE International Conference on Consumer Electronics (ICCE). IEEE. pp 1–4 https://doi.org/10.1109/ICCE.2018.8326347, http://ieeexplore.ieee.org/document/8326347/
86. Van Ma L, Park J, Nam J, Jang J, Kim J (2019) An efficient scheduling multimedia transcoding method for DASH streaming in cloud environment. Clust Comput 22(S1):1043–1053. https://doi.org/10.1007/s10586-017-1259-8
87. Pääkkönen P, Heikkinen A, Aihkisalo T (2019) Online architecture for predicting live video transcoding resources. J Cloud Comput 8(1):1–24. https://doi.org/10.1186/s13677-019-0132-0
88. Khan WZ, Ahmed E, Hakak S, Yaqoob I, Ahmed A (2019) Edge computing: a survey. Futur Gener Comput Syst 97:219–235. https://doi.org/10.1016/j.future.2019.02.050
89. Jin Y, Wen Y, Westphal C (2015) Optimal transcoding and caching for adaptive streaming in media cloud: an analytical approach. IEEE Trans Circuits Syst Video Technol 25(12):1914–1925. https://doi.org/10.1109/TCSVT.2015.2402892
90. Baccour E, Erbad A, Bilal K, Mohamed A, Guizani M (2020) PCCP: Proactive Video Chunks Caching and Processing in edge networks. Futur Gener Comput Syst 105:44–60. https://doi.org/10.1016/j.future.2019.11.006
91. Taleb T, Frangoudis PA, Benkacem I, Ksentini A (2020) CDN slicing over a multi-domain edge cloud. IEEE Trans Mob Comput 19(9):2010–2027. https://doi.org/10.1109/TMC.2019.2921712
92. Zhao X, Zhang S, Dou W (2020) Multi-request scheduling and collaborative service processing for DASH-video optimization in cloud-edge network. In: 2020 IEEE 13th International Conference on Cloud Computing (CLOUD). IEEE, pp 582–589. https://doi.org/10.1109/CLOUD49709.2020.00087, https://ieeexplore.ieee.org/document/9284322/
93. Zhuang Z, Guo C (2012) Building cloud-ready video transcoding system for Content Delivery Networks (CDNs). In: 2012 IEEE Global Communications Conference (GLOBECOM). IEEE, pp 2048–2053 https://doi.org/10.1109/GLOCOM.2012.6503417, http://ieeexplore.ieee.org/document/6503417/
94. Benkacem I, Taleb T, Bagaa M, Flinck H (2018) Performance benchmark of transcoding as a virtual network function in CDN as a service slicing. In: 2018 IEEE Wireless Communications and Networking Conference (WCNC), vol. 2018-April. IEEE, pp 1–6 https://doi.org/10.1109/WCNC.2018.8377402, https://ieeexplore.ieee.org/document/8377402/
95. Zakerinasab MR, Wang M (2015) Dependency-aware distributed video transcoding in the cloud. In: 2015 IEEE 40th conference on Local Computer Networks (LCN), vol 26-29-Octo. IEEE. pp 245–252 https://doi.org/10.1109/LCN.2015.7366317, http://ieeexplore.ieee.org/document/7366317/

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Wilmer Moina-Rivera[1] · Miguel Garcia-Pineda[1] · Juan Gutiérrez-Aguado[1] · Jose M. Alcaraz-Calero[2]

Wilmer Moina-Rivera
wilmoiri@alumni.uv.es

Juan Gutiérrez-Aguado
juan.gutierrez@uv.es

Jose M. Alcaraz-Calero
jose.alcaraz-calero@uws.ac.uk

[1] Department of Computer Science, Universitat de València, Avda. de la Universitat, s/n, Burjassot, 46100 Valencia, Spain

[2] School of Computing, Engineering and Physical Sciences, University of the West of Scotland, High St., Paisley PA1 2BE, Scotland, UK