Cost Minimization of Cloud Services for On-Demand Video Streaming

Mahmoud Darwich1 · Yasser Ismail2 · Talal Darwich3 · Magdy Bayoumi4

Received: 11 June 2021 / Accepted: 9 April 2022 / Published online: 19 April 2022
© The Author(s), under exclusive licence to Springer Nature Singapore Pte Ltd 2022

Abstract
Cloud Technology is adopted to process video streams because of the great features provided to video stream providers such as the high flexibility of using virtual machines and storage servers at low rates. Video stream providers prepare several formats of the same video to satisfy all users’ devices specification. Video streams in the cloud are either transcoded or stored. However, storing all formats of videos is still costly. In this research, we develop an approach that optimizes cloud storage. Particularly, we propose a method that decides which video in which cloud storage should be stored to minimize the overall cost of cloud services. The results of the proposed approach are promising, it shows effectiveness when the number of frequently accessed video grow in a repository, and when the views of videos increases. The proposed method decreases the cost of using cloud services by up to 22%.

Keywords Cloud · Storage · Video Stream · Transcoding · Clustering

Introduction
Video streaming became widely accessible by all display devices. Almost every house around the globe possesses at least a smartphone or desktop computer, or laptop, which eases accessing social media to upload and post video streams [1, 2]. Users usually post several videos on social media. Moreover, a video stream needs to be converted to several formats to meet the specifications of different display devices (e.g., network bandwidth, frame rate, and resolution) [3]. Thinking about the huge number of users who post videos on the internet is challenging and needs to be addressed.

Generally, Video conversion requires powerful machines to perform the transcoding operations promptly to avoid any delay on users’ end. Therefore, Video stream providers adopted the cloud platform to transcode their videos. They prepare several formats and stored them to make them ready upon request by the users. However, storing all formats incurs a high cost paid by the video stream providers [4].

The pattern of accessing video streams is a long-tail distribution. This means the beginning of the curve is allocated for frequently accessed video streams and the remaining of the curve is for rarely accessed videos. The frequently accessed video streams are a small portion while the rarely accessed video streams are huge. The access pattern implies that the video stream providers’ repositories contain a small percentage of frequently accessed videos and a large number of rarely accessed videos [7].

Transcoding operation is intensive and costly, it is more expensive than storage cost because the virtual machines are charged on an hourly basis, and storing servers are charged monthly [9]. Transcoding all formats and videos could increase the cost significantly when using cloud services. Therefore, it would be less expensive if the video stream providers find a method to store the videos [8].

Accordingly, storing the frequently accessed video streams is an effective solution to avoid the costly...
transcoding operation each time a video is accessed. In this paper, we discuss an approach to how to store the frequently accessed video in cloud storage. Particularly, we develop a method that clusters the frequently accessed video streams and then stores them in their suitable cloud storage.

Our contributions in this research are summarized as follows: (1) we propose an approach that reduces the cost of could storage by applying clustering on frequently accessed videos. (2) We analyze the effectiveness of the approach when changing the percentage of frequently accessed videos in repositories and when changing the video views.

This research is an extended work of the previous research [5]. Particularly, the cost of transcoding a video stream is modeled in Sect. “Cost of Transcoding Video Streams in Cloud”. Furthermore, we develop a model (ratio) that decides either storing or deleting and transcoding of a video stream in the repository as discussed in Sect. “Decision of Transcoding and Storing a Video Stream”. Thus, the cost of transcoding and the ratio are considered in the execution of the proposed algorithm (Steps 1–10) as detailed in Sect. “Algorithm 1.6”. Then we implement experiments and analyze the effect of the proposed approach for different scenarios in terms of views and percentage of frequently accessed video variation. Accordingly, we analyze the performance of the proposed method when the views of video vary as illustrated in Fig. 9. Additionally, we analyze the effectiveness of the proposed method when we include the CDN cost in the experiments as shown in Fig. 8b.

The structure of this paper is organized as follows: Section “Introduction” introduces the research topic, Sect. “Related Work” discusses the related work, Sect. Background” provides a background, Sect. “Proposed Clustering Video Stream Method” presents the proposed method, Sect. “Experiment Setup” shows the experiment setup, Sect. “Experimental Results” reveals the results discussion, and Sect. “Conclusion” concludes the paper.

Related Work

Several related works were done to reduce the incurred cost of cloud paid by video companies. Jin et al. [10] presented an erasure frame coding for video streams in the cloud to categorize the videos based on their importance. Their experimental result shows the effectiveness of their design and the reduction of the storage cost. Balamurugan et al. [11] proposed an algorithm that aims to eliminate duplicate video without affecting the unique video data for video surveillance systems. Their proposed algorithm achieves a high efficiency of saving more storage in the system and thus it reduces the storage cost Haynes et al. [12] developed a system for video storage that improves the efficiency of storing streaming by up to 45% and reduces the storage cost by up 54%. Their scheme manages video data and eliminates video redundancies from the storage system. Yan et al. [13] proposed a method that removes the duplicate video versions from the cloud storage while maintaining the privacy of IoT metadata. Their proposed paradigm operates at the file-level and duplication levels. Hu et al. [14] addressed the problem of sharing many videos on social media which poses a high pressure on the cloud storage infrastructure. The authors tackled this problem by proposing a model that motivates the users to upload videos to local storage. Their approach reduces the traffic and service latency. Kim et al. [22] developed an approach by implementing an intra-cloud and parallel framework to transcode videos. Their design could improve precisely the tasks assignments on virtual machines and reduced the transcoding time of videos. Iturriaga et al. [23] presented a scheme that distributes the content of video streams in cloud-based Content Delivery Network CDN. their proposed approach minimized the utilization of cloud resources. Thus, the incurred cost was reduced while maintaining the quality of service QoS of the users. Darwich et al. [24] developed algorithms aiming to decrease the cost of the cloud resources paid by videos companies. They came up with an approach that calculates the access frequency to a video and thus their proposed algorithms categorize the videos that should be stored or deleted and then transcoded upon request. Gao et al. [25] developed a method that applies the transcoding operation on parts of videos. Their scheme stores the first part of a video which is more frequently accessed while deleting the rest of it and then transcoding it upon request. Their results came promising where they could efficiently use the cloud resources and lower the cost. They proved that their model could reduce the overall cost by up to 30% compared to fully storing the video. Zhao et al. [19] presented an approach that trades-off between the transcoding operations and storing videos to minimize the cost of using the cloud. Particularly, they built a weight graph that describes the transcoding relations among videos. They used the views and the weight graph to decide which videos should be stored or deleted. Their results showed a cost reduction compared to previous methods. Jokhio et al. [17] proposed a scheme that decides whether the video is to be stored or deleted and transcoded on-demand. In their design, they input the video information popularity to their algorithm. Then their method decides how long the video should be kept in the repository. Their results revealed efficiency in terms of cost.
Background

Content of Video Stream

The content of a video stream is formed by several sequences as shown in Fig. 1. Each sequence is composed by Group Of Pictures (GOPs). A sequence starts with a block called header which contains data about the sequence. Further, a GOP consists of several frames, it starts with a header, and then a sequence of different types of frames are followed. P (Predicted), B (Bi-directional), and I (intra) frames are the major types found in a GOP. Further, a frame contains small blocks called macroblock (MB) [26].

Video streams are created with different formats based on bit rate, spatial resolution, and frame rate. Generally, before streaming a video, it is modified to meet the user’s device standards (e.g., frame rate, network bandwidth, and device resolution). These modification operation are achieved on GOPs [29, 30] and is called video transcoding. we brief the transcoding types as follows:

1. Bit Rate Adjustment High-quality videos are streamed with a high bit rate and require large bandwidth. Taking into consideration the variation of the bandwidth among users’ devices, a video is modified based on the requirement of the network bandwidth to avoid any delay on the user’s side [31].

2. Spatial Resolution Reduction Considering the variety of users’ devices, a video stream does not fit all display screens specification. Therefore, the spatial resolution should be adjusted and downscaled to match low-resolution screens. In such cases, the quality of videos is impacted and reduced [20].

3. Video Compression Standard Conversion Video compression techniques are highly important for streaming and storing because some videos have large sizes and need to be decreased. The common used compression techniques are MPEG2, H.264, and HEVC [32–34].

Cloud Services Video Stream Transcoding

Cloud resources are well known for the great services offered to users because they are charged in a pay-as-you-go. Video stream providers utilize the cloud to stream videos to viewers. The major cloud services for video streaming are:

- Virtual Machines: are used for transcoding operations of videos and are charged per hour.
- Storage: is an essential service provided by cloud providers. It is a based monthly charge.
- Content Delivery Network (CDN): is a service in the cloud that offers fast access to contents like video streams, it avoids delay to stream videos. CDN replicates video streams in different geographical locations to reduce the traffic travel time in network [15, 16].

Amazon Web services (AWS)\(^1\) is well known for cloud services and the great features with high reliability presented to users. Even though, our approach is not dependent on Amazon only and could be applied to any other cloud providers. In this research, we consider AWS for the evaluation of our proposed approach.

Different types of storage are provided by AWS:

- S3 Standard: It is typically used to store frequently accessed video streams.
- S3 Standard-Infrequent Access (S3 Standard-IA): is used to store infrequently accessed video streams.
- One Zone-Infrequent Access (S3 One Zone-IA): is used to store less infrequently accessed video streams.
- S3 Glacier: stores frequently access video streams for the long term.

Proposed Clustering Video Stream Method

Table 1 represents the description of the symbols used in the equations and the proposed algorithm.

Proposed Storage Approach

We propose an approach that facilitates the management of video storage in the cloud. Figure 2 presents the flow of video processing. In the cloud, a video-on-demand has two options either stored or transcoded (i.e., the video is deleted and transcoded upon request by the users). Once

\(^1\) https://aws.amazon.com/.
Table 1  Symbols used in the proposed method

| Symbols  | Description |
|----------|-------------|
| \(m_i\)  | Total number of GOPs in video \(i\) |
| \(G_{ij}\) | \(j\)th GOP in video \(i\) |
| \(G_{\text{threshold}}\) | GOP threshold in video \(i\) |
| \(j\) | index of the GOP \(G_{ij}\) in video \(i\) |
| \(r_i\) | Views received by video \(i\) |
| \(\hat{\xi}_{ij}\) | Estimated views received by GOP \(G_{ij}\) |
| \(S_{ij}\) | Size of GOP \(G_{ij}\) in megabytes |
| \(P_{\text{storage}}\) | General price of cloud storage ($/GB) |
| \(P_{\text{storage1}}\) | Price of cloud storage S3 Standard used in cluster 1 ($/GB) |
| \(P_{\text{storage2}}\) | Price of cloud storage S3 Standard-IA used in cluster 2 ($/GB) |
| \(P_{\text{storage3}}\) | Price of cloud storage S3 One Zone-IA used in cluster 3 ($/GB) |
| \(P_{\text{storage4}}\) | Price of cloud storage S3 Glacier used in cluster 4 ($/GB) |
| \(P_{\text{tran}}\) | Price of cloud virtual machine for ($/hour) |
| \(SC_{Gi}\) | Storage cost of GOP \(G_{ij}\) |
| \(TC_{Gi}\) | Transcoding cost of GOP \(G_{ij}\) |
| \(TC_{\text{video}}\) | Transcoding cost of video \(i\) |
| \(SC_{\text{video}}\) | Storage cost of video \(i\) |
| \(SC_1\) | Storage cost of GOPs in cluster 1 |
| \(SC_2\) | Storage cost of GOPs in cluster 2 |
| \(SC_3\) | Storage cost of GOPs in cluster 3 |
| \(SC_4\) | Storage cost of GOPs in cluster 4 |
| \(SC_{1:}\text{threshold}\) | Storage cost of first GOP till GOP threshold |
| \(TC_{\text{remaining}}\) | Transcoding cost of the remaining GOPs after the threshold |
| \(R_{Gi}\) | Ratio of storage cost to transcoding cost of GOP of \(G_{ij}\) |
| \(R_{\text{video}}\) | Ratio of storage cost to transcoding cost of video of \(i\) |
| \(\text{COST}_{\text{total}}\) | Total cost of using cloud for video of \(i\) |

Fig. 2  Proposed storage approach for video streams in cloud

The video is selected to be stored, it goes through the clustering method to send it to the suitable storage. The storage mechanism is based on the number of views received by a video and the accessibility rate of the cloud storage. If a video does not meet the criteria for storage, it is deleted from the repository. In case a user requests it to watch, the original video, which is stored cloud, will be transcoded and generated in the right format for the video to fit the user’s device.

**Video Streams Access Pattern**

The cost of using cloud service to store or transcode video streams is mainly dependent on accessing video streams. Therefore, it is important to explore the pattern of video accesses. Previous studies revealed the access pattern to videos is a long-tail distribution [21]. In such a pattern, the beginning of a video is highly watched and the rest parts are almost rarely accessed. Therefore, the distribution of GOPs in a video stream could be implemented by applying the Power Law model [27] to represent the long-tail pattern. Thus, the views of GOPs could be estimated. Assume \(G_{ij}\) is the \(j\)th GOP in a video \(i\) and its views \(\hat{\xi}_{ij}\) is estimated in Eq. 1 where \(\gamma_i\) represents the views of video \(i\) and \(\alpha\) is constant and \(j\) is the index of GOP in the video \(i\). We select the value of \(\alpha\) equals to 0.1 to model the long-tail distribution of GOPs in the video.

\[
\hat{\xi}_{ij} = \frac{r_i}{j^{\alpha}}.
\] (1)
Cost of Storing Video Streams in Cloud

The rates of cloud storage are charged based on the size of a video. Note that a video stream is composed of many GOPs. Knowing the size of a GOP $G_{ij}$, the storage cost of a $j$th in a video stream $i$ is computed in Eq. 2 where $SC_{G_{ij}}$ is the storage cost of $G_{ij}, S_{ij}$ is the size of $G_{ij}$ in megabytes, and $P_{storage}$ is the cloud storage rate in ($/GB$). In Eq. 2, we divide by $2^{10}$ to convert from megabytes to gigabytes.

$$SC_{G_{ij}} = \frac{S_{ij} \times P_{storage}}{2^{10}}. \tag{2}$$

Similarly, the storage cost of a whole video is calculated by summing up the storage cost of its all GOPs, the concept of Eq. 2 is extended to a video level and thus the storage cost of a video stream $i$ is $SC_{video}$ and illustrated in Eq. 3

$$SC_{video} = \sum_{ij} SC_{G_{ij}} = \sum_{ij} \frac{S_{ij} \times P_{storage}}{2^{10}}. \tag{3}$$

Cost of Transcoding Video Streams in Cloud

The cost of transcoding a video depends on the time of utilization of the virtual machine in the cloud and its type, the VM rate is in an hourly basis. Assume $TC_{G_{ij}}$ is the transcoding cost of GOP $G_{ij}$ in a video $i$, $\tau_{ij}$ is the time to complete the transcoding operation in seconds, and $P_{trans}$ is the price of a virtual machine per hour. The equation of the transcoding cost is given in Eq. 4. We divide by 3600 to convert seconds to an hour.

$$TC_{G_{ij}} = \frac{\tau_{ij} \times P_{trans}}{3600}. \tag{4}$$

It is worth to mention that Eq. 4 estimates the transcoding cost of GOP one time. Therefore, if the GOP $G_{ij}$ receives $\xi_{ij}$ views (i.e., which was estimated in Eq. 1), then the transcoding cost of $G_{ij}$ is computed in Eq. 5

$$TC_{video} = \frac{\xi_{ij} \times \tau_{ij} \times P_{trans}}{3600}. \tag{5}$$

It is noteworthy that the transcoding time $\tau_{ij}$ of GOP $G_{ij}$ is based on the historical calculated transcoding time of the same GOP.

Accordingly, Eq. 4 is implemented at the video level to calculate the transcoding cost $TC_{video}$ for video stream $i$ by summing up the transcoding of its GOPs and is illustrated in

$$R_{G_{ij}} = \frac{SC_{G_{ij}}}{TC_{G_{ij}}} = \frac{\frac{S_{ij} \times P_{storage}}{2^{10}}}{\frac{\xi_{ij} \times \tau_{ij} \times P_{trans}}{3600}}. \tag{7}$$

When the storage cost $SC_{G_{ij}}$ of a GOP $G_{ij}$ is greater than its transcoding cost $TC_{G_{ij}}$ (i.e. $SC_{G_{ij}} > TC_{G_{ij}}$) it is worth to delete the GOP and transcode it upon request. In this case the ratio $R_{G_{ij}} > 1$. That means the GOP $G_{ij}$ receives less views.

When the storage cost $SC_{G_{ij}}$ of a GOP $G_{ij}$ is less than its transcoding cost $TC_{G_{ij}}$ (i.e. $SC_{G_{ij}} < TC_{G_{ij}}$) it is worth to store the GOP because it is easier to provide the user a stored GOP than transcode it on fly. In this case the ratio $R_{G_{ij}} < 1$. That means the GOP $G_{ij}$ receives more views and becomes frequently accessed GOP.

When the storage cost $SC_{G_{ij}}$ of a GOP $G_{ij}$ is equal to its transcoding cost $TC_{G_{ij}}$ (i.e. $SC_{G_{ij}} = TC_{G_{ij}}$) it is worth to store the GOP because it is easier to provide the user a stored GOP than transcode it on fly. In this case the ratio $R_{G_{ij}} = 1$.

Similarly, the ratio concept discussed for GOPs could be achieved on video level. Thus, the ratio of storage cost to transcoding cost of video stream $i$ is illustrated in Eq. 8

$$R_{video} = \frac{SC_{video}}{TC_{video}} = \frac{\sum SC_{G_{ij}}}{\sum TC_{G_{ij}}} = \left(\frac{\sum S_{ij} \times P_{storage}}{2^{10}}\right) \cdot \left(\frac{\sum \xi_{ij} \times \tau_{ij} \times P_{trans}}{3600}\right). \tag{8}$$

When the storage cost $SC_{video}$ of a video $i$ is greater than its transcoding cost $TC_{video}$ (i.e. $SC_{video} > TC_{video}$) it is worth to delete the video and transcode it upon request. In this case the ratio $R_{video} > 1$. That means the video $i$ receives less views.

When the storage cost $SC_{video}$ of a video $i$ is less than its transcoding cost $TC_{video}$ (i.e. $SC_{video} < TC_{video}$) it is worth

\[
\text{SN Computer Science} \\
\text{A SPRINGER NATURE journal}
\]
to store the video. In this case the ratio $R_{video} < 1$. That means the video $i$ receives more views and becomes frequently accessed video.

When the storage cost $SC_{video}$ of a video $i$ is equal to its transcoding cost $TC_{video}$ (i.e. $SC_{video} = TC_{video}$) it is worth to store the video because it is easier to provide the user a stored version than transcoding it on fly. In this case the ratio $R_{video} = 1$.

**Algorithm 1**

The algorithm is an extended version of the previous research [5]. The purpose of the algorithm is to cluster the frequently accessed video stream and store them in different cloud storage to minimize the incurred cost and delete the non frequently accessed videos and transcoding upon request using an original version. The algorithm is carried out on all GOPs in any video in the repository of video streams providers and its pseudo-code is shown in Algorithm 1.

The inputs of the algorithm are GOPs, their sizes, their transcoding times, price of cloud storage, and price of transcoding machines. The algorithm output decides if videos should be deleted and transcoding upon request or stored in different cloud storage and then it computes the total cost of using cloud services for each video.

For each GOP $G_{ij}$ in each video $i$ in the repository (Step 1), the algorithm estimates the views $\xi_{ij}$ received by each GOP by implementing Eq. 1 (Step 2). The cost of storage and transcoding are computed by applying the Eqs. 2 and 4 (steps 3 and 4). Then in the next step (Step 5), the ratio of storage cost to transcoding cost is calculated. If the ratio is less than 1, it means that the storage cost is less than the transcoding cost and hence the algorithm decides to store the GOP (Steps 6 and 7). The algorithm keeps searching for a GOP that the next GOP has a ratio greater than 1. This GOP is called threshold GOP $G_{threshold}$. All the remaining GOPs after the threshold have a ratio greater than 1 which means the storage cost is higher than the transcoding cost.

The algorithm requests to delete all GOPs after the threshold and transcoding them upon request (Steps 8 and 9). Figure 3 illustrates the access pattern of GOPs in a video and which is a long-tail distribution. It is noticed that the beginning GOPs are more accessed than the rest GOPs which are located after the threshold GOP.

**Algorithm 1**: Clustering Method

**Input**
- All $n$ GOP of video $i$
- Size of GOP $G_{ij}$ in video $i$: $S_{ij}$
- Needed time to transcoding GOP $G_{ij}$: $\tau_{ij}$
- Price of transcoding machine: $P_{trans}$
- Price of cloud storage: $P_{storage}$
- Count views of GOP $G_{ij}$: $\xi_{ij}$

**Output**: Total cost of using cloud services for video stream $i$

1. For each GOP $G_{ij}$ in each video stream $i$
2. Estimated views for GOP $G_{ij}$: $\xi_{ij} \leftarrow \frac{S_{ij}}{P_{storage}}$
3. Estimate cost of transcoding of GOP $G_{ij}$ that receives $\xi_{ij}$ views: $TC_{G_{ij}} \leftarrow \frac{\xi_{ij} \cdot \tau_{ij} \cdot P_{trans}}{3600}$
4. Compute cost of storage of GOP $G_{ij}$:
   - $SC_{G_{ij}} \leftarrow \frac{\xi_{ij} \cdot P_{storage}}{2^{10}}$
5. Compute the ratio: $R_{G_{ij}} \leftarrow \frac{SC_{G_{ij}}}{TC_{G_{ij}}}$
6. If $R_{G_{ij}} \leq 1$
   - Store $G_{threshold} \leftarrow G_{ij}$
7. Else
   - Delete and Transcode the remaining GOPs upon request;
8. Apply K-Means clustering on the stored GOPs $G_{ij}$ to $G_{threshold}$ with $K = 4$
9. Compute storage cost of Cluster 1:
   - $SC_1 \leftarrow \sum S_{ij} \cdot P_{storage} \cdot 2^{10}$
10. Compute storage cost of Cluster 2:
    - $SC_2 \leftarrow \sum S_{ij} \cdot P_{storage} \cdot 2^{10}$
11. Compute storage cost of Cluster 3:
    - $SC_3 \leftarrow \sum S_{ij} \cdot P_{storage} \cdot 2^{10}$
12. Compute storage cost of Cluster 4:
    - $SC_4 \leftarrow \sum S_{ij} \cdot P_{storage} \cdot 2^{10}$
13. Compute cost of transcoding of the remaining GOPs in video $i$:
    - $TC_{remaining} \leftarrow \sum \frac{\xi_{ij} \cdot \tau_{ij} \cdot P_{trans}}{3600}$
14. Total cost of using cloud services for a video stream $i$:
    - $COST_{total} \leftarrow SC_{1-threshold} + TC_{remaining}$

Fig. 3 Access pattern to a video follows long-tail. All GOPs before GOP$_{threshold}$ are clustered and stored while the rest GOPs are deleted and transcoded upon request.

Then the K-means algorithm is implemented with $K = 4$ (steps 11–15) to cluster the stored GOPs. We select the value of $k = 4$ to match the storage types in the cloud. The reason
we rely on K-means is because of its simplicity and we do not consider many factors in the clustering process. Cluster 1 contains the most frequently accessed GOPs. Sequentially, clusters 2, 3, and 4 are dedicated for less frequently accessed GOPs. Note that the storage price of cluster 1 is higher than the rest of the cloud storage because it provides the fastest accessibility. While clusters 2, 3, and 4 have different prices and accessibility rates. In this research, we dedicate S3 Standard for Cluster 1, S3 Standard-IA for Cluster 2, S3 One Zone-IA for Cluster3, and S3 Glacier for cluster 4. In Step 16, the algorithm computes the storage cost of the stored GOPs. While in Step 17, it computes the transcoding cost of the remaining GOPs. The last Step 18, the total cost of using the cloud service, which is the summation of storage and transcoding costs, is calculated for the video \(i\).

### Experiment Setup

#### Videos Synthesis

Large repositories of video streams are necessarily required for the experiments. However, providing access to video stream providers is almost impossible. Furthermore, downloading video from the internet is a lengthy and costly procedure. Therefore, we come up with a way to synthesize video streams. The content of the synthesized videos is the characteristics that fit the experimental purposes. Particularly, size, transcoding time, and count of GOPs in a video are the necessary variables for the experiments.

Accordingly, an analysis of a bunch of videos downloaded from YouTube is performed to reveal the relations among size, transcoding time, and numbers of GOPs in videos. The downloaded videos contain different categories including music, movies, animation, documentary, news, and sports. Transcoding operations are applied to the videos to generate information about the size, transcoding time, and GOPs number in each video. Furthermore, the extracted information is analyzed, we conclude that the GOPs sizes and GOPs number fits in a Gaussian curve as shown in Figs. 4 and 5. The Gaussian values are detailed in Table 2. More importantly, we implement the linear regression analysis on GOPs size and transcoding time used in [28], the analysis results in a linear equation between GOPs size and their corresponding transcoding time. The regression analysis is displayed Fig. 6.

By utilizing the generated data and the relationship between the variables, we create repositories of videos, each one contains 50,000 videos.

#### Videos Access Pattern

We create the access pattern to synthesized video in the repository by implementing a Weibull distribution equation. Accordingly, the plotted curve follows a long tail shape. The access pattern shows the frequently accessed videos at the beginning of the curve while the videos having fewer views are located in the curve’s long tail. The Weibull distribution has two coefficients shape \(\alpha\) and scale \(\beta\) which play a critical role to define the percentage of frequently accessed videos in the repository. Decreasing the coefficient \(\alpha\) value results in increasing the number of frequently accessed videos, we vary \(\alpha\) between 0.4 and 2.4 and set \(\beta = 1\) in the experiments for the sake of having a long tail shape.

#### Storage and Computation Prices

The prices of cloud storage and transcoding virtual machines utilized in the experiments are the actual prices of Amazon Web Service (AWS) as detailed in Tables 3 and 4.

### Proposed Method and other Methods for Comparison

The proposed approach is compared to previous works and two baselines methods:

- **Fully Storing Videos Method** which stores all videos regardless of their frequency of access.
- **Deleting and Transcoding all Videos Method** them upon request.
- **Partial-Storing videos method** [6] which stores the most-watched part of a video in one of the cloud storage.
- **VideoTranscode or Delete Method**: This method reduces the cost of cloud services for video streams and was proposed by Jokhio et al. [17]. They proposed an approach to store the whole video or delete and transcode it upon request. They considered the popularity of videos in their method to decide which video should be stored or transcoded.
- **Trade-off Between Storing and Transcoding Method**: Zhao et al. proposed a method to tare-off between storing and transcoding costs. Thus, their scheme reduced the cost of cloud services. In their approach, they used the transcoding relationships among the versions of a video with the views of each video. Their method decides which video versions should be stored or deleted and transcoded on demand.
Experimental Results

Effect of Varying the Percentage of Frequently Accessed Videos in Repository

We build several repositories made of synthesized videos. Each repository contains 50,000 videos with a different percentage of frequently accessed videos. We adjust the shape coefficient in the Weibull distribution equation to change the percentage of the frequently accessed video in each repository. Table 5 displays the α values with the corresponding percentage of frequently accessed videos.

The simulation results are presented in Fig. 8a. The horizontal axis represents the varying percentage of frequently accessed videos and the vertical axis represents the total cost of cloud services. We note that the cost of applying the fully storing method is around $1600 and constant along with a different percentage of frequently accessed videos because it aims to store all videos regardless of the frequency access to the videos. The method of deleting videos and transcoding upon request work fairly for a small percentage of frequently accessed videos. However, it incurs a high cost when the percentage gets increased because we assume each time the user views the video, we apply the transcoding operation on the original video.
The store or transcode method [17] significantly minimizes the cost of using cloud service, it decides efficiently which videos streams should kept in the repository and which videos should be deleted and transcoded upon demand. Moreover, it could reduce the cost by up 57% compared to fully storing method. We notice that the method steadily keeps the cost almost unchangeable and low when the percentage of the frequently accessed increases in the repository.

The trade-off storage and transcoding method [19] further decreases the cost of cloud services. It highly reduces the cost up by 60% compared to the fully storing method. This method has slightly more impact on reducing the cost when the percentage the frequently accessed video streams gets high in the repository.

It is worth to mention that methods of [17, 19] are essentially based on transcoding or storing the whole video. While the partially storing method [5], which partially stores or transcode the GOPs of a video, outperforms the two aforementioned methods, it reduces the cost significantly as it stores only the frequently accessed parts of the videos in the S3 standard storage and deletes the rest. It could decrease the incurred cost by up to 75% compared to the fully storing method. It is worth mentioning that the method works efficiently when the percentage of frequently accessed increases because it decides more parts of videos to be stored, which is less expensive than transcoding them. The proposed method outperforms all methods because it stores frequently accessed GOPs of each video in different cloud storage. It differs from the partially storing method, which stores all frequently accessed video in one storage (i.e. S3 standard), and which has the highest price, while the proposed method aims to distribute the videos in different cloud storage which have lower rates as shown in Fig. 7. It could reduce the cost by 20% compared to the partially storing method.

Similarly, we run our experiments using the Content Delivery Network shown in Fig. 8b which incurs a high significant cost for fully storing method around $6000 because all videos are replicated and stored in different geographical locations. As the percentage of frequently accessed videos increases, the cost of using CDN increases, since more replicated videos are stored in different locations. Our proposed method outperforms the other methods and reduces the cost by up to 15% compared to the partially storing method.

**Effect of Varying the Number of Views for Frequently Accessed Videos**

The frequently accessed videos have variant views in repositories. It is worthwhile to study the effect of view variation when applying the proposed method. In this experiment, we increase the average views of frequently accessed videos by up to 5%. The fully storing method keeps the cloud cost constant along with changing views because it stores all videos in the repository regardless of frequency of access and their views variations, this method imposes significant cost paid by the videos providers. The fully deleting and transcoding method increases the cloud cost gradually when the views rate increase because for each single view request, the video is transcoded and the cloud charges video stream providers. When the views of a video are increased by 5% the cost is doubled as shown in Fig. 9.

The two methods [17, 19] efficiently reduce the cost of cloud services. particularly, both methods significantly decreases the cost by up 69% compared to the fully storing method. As the views of video streams increase, both methods work efficiently.

The partially storing method reduces the incurred cost significantly when videos views increase because more parts (i.e., GOPs) are stored in S3 standard storage which incurs less cost than transcoding. Moreover, it overcomes the performance of the aforementioned methods and decreases the cost further by up 7% compared to the methods [17, 19].

---

![Clustered frequently accessed GOPs based on the number of views](image)

**Table 5** Percentage of frequently accessed videos is changed by adjusting $\alpha$ in the Weibull distribution equation

| $\beta$ | $\alpha$ | Frequently accessed videos (%) |
|--------|----------|-------------------------------|
| 1      | 0.4      | 30                            |
| 0.6    | 25       |                               |
| 1      | 20       |                               |
| 1.4    | 15       |                               |
| 1.8    | 10       |                               |
| 2.4    | 5        |                               |
The proposed clustering method reduces further the cost and outperforms the other methods because its concept is based on storing the frequently accessed video in S3 Standard, S3 Standard-IA, One Zone-IA, and S3 Glacier which charge less than S3 Standard. The overall cost reduction by implementing the proposed method is 20% compared to the partially storing method.

**Conclusion**

In this paper, we propose a method to minimize the cost of cloud resources. This research is beneficial for video stream providers, it aims to minimize the cost of cloud resources. Particularly, we develop a method that clusters similar frequently accessed videos based on the views and then distributes them to suitable cloud storage. Our proposed method shows efficiency in cost reduction when the repository contains a high percentage of frequently accessed video, it reduces the cost by up to 20% compared to the partially storing method. Moreover, when increasing the views of frequently accessed videos the incurred cost is reduced by up to 15%. In addition, when considering CDN in the experiments, our proposed method outperforms other methods and decreases the cloud cost by up to 22%.

---

Fig. 8 Incurred costs of fully storing, fully deleting and re-transcoding, store or transcode [17], trade off storage and transcoding [19], partially storing [5], and proposed clustering methods methods: a percentage of frequently accessed videos in the repository; b Content delivery network cost is considered to the storage cost

Fig. 9 Incurred costs of fully storing, fully deleting and re-transcoding, store or transcode [17], trade off storage and transcoding [19], partially storing [5], and proposed clustering methods when the count views of videos is raised
Funding  No funding resources.

Declarations

Conflict of interest  The authors declare that they have no conflict of interest.

References

1. Li X, Darwich M, Salehi MA, Bayoumi M. A survey on cloud-based video streaming services. Amsterdam: Elsevier; 2021.
2. Report GIP. https://www.sandvine.com/trends/global-internetphenomena/. Accessed Jan 2019.
3. Ahmad I, et al. Video transcoding: an overview of various techniques and research issues. IEEE Trans Multimedia. 2005;7(5):793–804.
4. Li X, Salehi MA, Bayoumi M. VLSC: video live streaming using cloud services. In: 6th IEEE international conferences on big data and cloud computing (BDCloud), 2016.
5. Darwich M, Ismail Y, Darwich T, Bayoumi M. Cost-efficient storage for on-demand video streaming on cloud. In: 2020 IEEE 6th world forum on internet of things (WF-IoT). New Orleans, LA, USA, 2020.
6. Darwich M, Beyazit E, Salehi MA, Bayoumi M. Cost efficient repository management for cloud-based on-demand video streaming. In: 2017 5th IEEE international conference on mobile computing, services, and engineering (MobileCloud), pp. 39–44. IEEE, 2017.
7. Sharma N, Krishnappa DK, Irwin D, Zink M, Shenoy P. GreenCache: augmenting off-the-grid cellular towers with multimedia caches. In: Proceedings of the 4th ACM multimedia systems conference, 2013.
8. Li X, Salehi MA, Bayoumi M, Buyya R. CVSS: a cost-efficient and QoS-aware video streaming using cloud services. In: Proceedings of the 16th ACM/IEEE international conference on cluster cloud and grid computing, ser. CCGrid ’16, May 2016.
9. https://aws.amazon.com/ec2/pricing/on-demand/.
10. Jin H, Wu C, Xie X, Li J, Guo M, Lin H, Zhang J. Approximate code: a cost-effective erasure coding framework for tiered video storage in cloud systems. In: Proceedings of the 48th international conference on parallel processing, pp. 1–10. August, 2019.
11. Balamurugan NM, Adimoolam M, John A. A novel efficient algorithm for duplicate video comparison in surveillance video storage systems. J Ambient Intell Human Comput. 2021. https://doi.org/10.1007/s12652-021-03119-7.
12. Haynes B, Daum M, He D, Mazumdar A, Balazinska M, Cheung A, Ceze L. VSS: a storage system for video analytics [technical report]. arXiv preprint arXiv:2103.16604. 2021.
13. Yan H, Li X, Wang Y, Jia C. Centralized duplicate removal video storage system with privacy preservation in IOT. Sensors. 2018;18(6):1814.
14. Hu H, Wen Y, Niyato D. Public cloud storage-assisted mobile social video sharing: a supermodular game approach. IEEE J Select Areas Commun. 2017;35(3):545–56.
15. Saroiu S, et al. An analysis of internet content delivery systems. ACM SIGOPS Oper Syst Rev. 2002;36:315–27.
16. Vakali A, Pallis G. Content delivery networks: status and trends. IEEE Internet Comput. 2003;7(6):68–74.
17. Jokhio F, Ashraf A, Lafond J, Lilius S. A computation and storage trade-off strategy for cost-efficient video transcoding in the cloud. In: 39th Euromicro conference on software engineering and advanced applications, 2013.
18. Zhao H, Zheng Q, Zhang W, Du B, Chen Y. A version-aware computation and storage trade-off strategy for multi-version VoD systems in the cloud. In: 2015 IEEE symposium on computers and communication, pp. 943–948, 2015.
19. Zhao H, et al. A version-aware computation and storage trade-off strategy for multi-version VOD systems in the cloud. In: 20th IEEE symposium on computers and communication (ISCC), 2015.
20. Bjork N, Christopoulos C. Transcoder architectures for video coding. IEEE Trans Consumer Electron. 1998;44(1):88–98.
21. Miranda, LCO, Santos RLT, Laender AHF. Characterizing video access patterns in mainstream media portals. In: Proceedings of the 22nd international conference on world wide web, 2013.
22. Kim H-W, Mu H, Park JH, Sangaih AK, Jeong Y-S. Video transcoding scheme of multimedia data-hiding for multiform resources based on intra-cloud. J Ambient Intell Human Comput. 2019. https://doi.org/10.1007/s12652-019-01279-1.
23. Iturriaga S, Goñi G, Nesmachnow S, Dorronsoro B, Tchernykh A. Cost and QoS optimization of cloud-based content distribution networks using evolutionary algorithms. In: Latin American high performance computing conference, pp. 293–306. Springer, Cham, 2018.
24. Mahmoud D, Salehi MA, Beyazit E, Bayoumi M. Cost-efficient cloud-based video streaming through measuring hotness. Comput J. 2019;62(5):641–56.
25. Guo G, Zhang W, Wen Y, Wang Z, Zhu W. Towards cost-efficient video transcoding in media cloud: insights learned from user viewing patterns. IEEE Trans Multimedia. 2015;17(8):1286–96.
26. Jokhio F, et al. Analysis of video segmentation for spatial resolution reduction video transcoding. In: International symposium on intelligent signal processing and communications systems (ISPACS), 2011.
27. Newman MEJ. Power laws, Pareto distributions and Zipf’s law. Contemporary physics. 2005;46(5):323–51.
28. Li X, Yamini J, Darwich MK, Landreneau B, Salehi MA, Bayoumi M. Performance analysis and modeling of video transcoding using heterogeneous cloud services. University of Louisiana at Lafayette, LA, Technical Report, 2016.
29. Ishfaq A, Wei X, Sun Y, Zhang Y-Q. Video transcoding: an overview of various techniques and research issues. IEEE Trans Multimedia. 2005;7(5):793–804.
30. Vetro A, Christopoulos C. Sun H. Video transcoding architectures and techniques: an overview. IEEE Signal Process Mag. 2003;20(2):18–29.
31. Werner O. Requantization for transcoding of MPEG-2 intra-frames. IEEE Trans Image Process. 1999;8(2):179–91.
32. Haskell BG, Puri A, Netravali AN. Digital video an introduction to MPEG-2 standards. New York: McGraw-Hill; 2001.
33. Wiegand T, Sullivan GJ, Bjontegaard G, Luthra A. Overview of the H. 264/AVC video coding standard. IEEE Trans Circuits Syst Video Technol. 2003;13(7):560–76.
34. Sullivan GJ, Ohm J-R, Han W-J, Wiegand T. Overview of the high efficiency video coding (HEVC) standard. IEEE Trans Circuits Syst Video Technol. 2012;22(12):1649–68.

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