Division-based Video Data Access Method for Hot/Cold Tiered Storage Systems

Dong Ling[1], Shuuichirou Murata[2], Ying-Feng Hsu[3], Tomoki Yoshihisa[3], Morito Matsuoka[3]

Abstract: Due to the recent prevalence of video-on-demand services, data centers for them store a large number of video files, and some data centers adopt hot/cold tiered storage systems. Hot/cold tiered storage systems achieve a low power consumption and low monetary cost because some data are stored in cold storage, of which the price per data unit is lower than that of hot storage. One of their drawbacks is the slow data access speeds of cold storage. This is a very large problem for video-on-demand services since long interruption times for video playback occur when clients do not receive each video data segment by the time to play it. However, conventional hot/cold tiered storage systems are designed for general services and lack the possibility to further reduce total playback interruption times for video-on-demand services. In this paper, we design a novel data access method for video-on-demand services using hot/cold tiered storage systems. In our method, each video file is divided into video file segments so that the clients can finish receiving subsequent segments while playing received segments. By storing the first segment into the hot storage and the others into the cold storage, our method achieves a lower power consumption and lower monetary cost while reducing total playback interruption times.

1. Introduction

With the rapid development of communication technologies and the popularization of smart phones and tablets, video-on-demand services are booming. Rough statistics show that the total duration of the video files uploaded to the YouTube service is over 1,000 hours everyday. To store such a large number of video files, data centers for video-on-demand services adopt hot/cold tiered storage systems ([1]). Hot/cold tiered storage systems consist of two types of storage devices: hot and cold. Hot storage has a comparatively higher I/O speed and lower seek time, like hard disk drives (HDD) or solid state drives (SSD), but the monetary cost per bit is higher than that of cold storage [2], such as Blu-ray discs. Specifically, one survey showed that cold storage consisting of Blu-ray discs can reduce the power consumption by 70% compared with hot storage consisting of hard disks ([3]). Hot/cold storage systems combine these two kinds of storage devices. Recently, many storage providers have employed these systems for their products, like the hybrid drive ([4]). Also, cloud storage providers like Google provide two types of cold storage ([5]).

Generally, video files are divided into video data segments to enable video-on-demand services. The clients of the users of the services receive them while playing a video. The segments are saved in hot or cold storage device separately. When a user requests playing a video from a hot/cold storage system, a system management server (HTTP server) first finds the location of the first segment from the database used for managing the two types of storage devices. After that, the server reads each segment and sends them continuously. In the case that data access speeds are low, long interruption times during video playback occur when clients do not receive each video data segment by the time to play it. Therefore, one of the drawbacks of hot/cold tiered storage systems is the slow data access speeds of cold storage. This is a very large problem for video-on-demand services since they are real-time services.

However, most hot/cold tiered storage systems are designed for general services, and they determine to which storage each video file should be stored on the basis of video playback popularity. However, as video-on-demand is a real-time service, accessing a video file that is currently saved in cold storage causes the playback interruption time to be long. In particular, interruption times for rarely accessed video files are very long ([6]). We previously proposed a method for automatically classifying video data that gives more rapid data access than manual classification in [7]. However, the data access method of the research was similar to that used for general files. There is a large possibility to reduce the interruption times caused by hot/cold tiered storage systems.

In this paper, we design a novel data access method for video-on-demand services using hot/cold tiered storage systems. In our proposed method, each video file is divided into video data segments so that the clients can finish receiving subsequent segments while playing received segments. By storing the first segment into the hot storage and the others into the cold storage, our method achieves a lower power consumption and a lower monetary cost while reducing total playback interruption times. Moreover, we develop a system with our proposed method. The system adopts the HLS (HTTP Live Streaming) protocol, which ensures that segmented video is played seamlessly. We evaluate the performance of our developed system in this research.

The rest of the paper is organized as follows. In Section 2, we will introduce related work. We will explain our proposed system in Section 3 and how to divide video files in Section 4. In Section 5, we will show the results of an evaluation of our proposed method. Finally, we conclude this paper and introduce future work in Section 6.

1 Graduate School of Information Science and Technology, Osaka University
2 Acutus Software, Inc.
3 Cyber rmedia Center, Osaka University

ⓒ2019 Information Processing Society of Japan

2. Related Work

A nanophotonics-enabled optical storage system was proposed in [8]. The authors summarized the recent developments on this kind of storage system and compared the systems with data centers that only adopt HDD. They found that their proposed optical storage systems have several advantages compared with HDDs, such as a much longer lifetime and much lower monetary cost.

A method called "Green HDFS" (Hadoop Distributed File System) was proposed in [9]. In this research, the authors conducted an analysis on the data access patterns of the Yahoo! Hadoop cluster. They found that some of the servers in the cluster were running, although some of them could be replaced with low-cost processors. They found that adopting their approach could save about 26% of energy.

Recently, some hot/cold tired storage systems have been proposed. One of their advantages is a lower power consumption compared with general storage systems. To estimate their power consumption more accurately, about 200 kinds of power consumption models are summarized in [10].

A method for classifying data for hot/cold tiered storage systems was proposed in [7]. The authors define the popularity of files and put the files with high popularity into hot storage and those with low popularity into cold storage. Specifically, they predict whether a file will be accessed in the next period of time on the basis of the previous period of time. They set the time period as one month. They also implemented ensemble learning that combines the SVM (Support Vector Machine) and DNN (Deep Neural Network) to enhance the accuracy of the prediction. Finally, they successfully reduced the usage of hot storage by 57% and maintained the user QoE by 88% for a campus cloud. However, the method does not focus on the interruption time problem of video-on-demand services.

A real-time power cycling system for data centers for video-on-demand services was proposed in [6]. The system regards the data for the head 2 minutes of each video as an intro clip. The intro clips and the full video data are distributed to servers. A naive Bayesian algorithm is used to determine how many servers are needed for video streaming, and unneeded servers are powered off to reduce power consumption. The authors evaluated their approach by using an actual workload trace from BBC iPlayer, an extremely popular video-on-demand service in the UK. Their evaluation focused on the reduction in power consumption, and the result showed that their approach could save up to 31% of server energy in each cluster while maintaining the QoE. The approach of the system is similar to our approach in that each piece of video data is divided into data segments. However, it does not adopt hot/cold tiered storage systems.

3. Assumed System Architecture

Fig. 1 shows our assumed system architecture. The architecture consists of an HTTP server, hot storage device, and

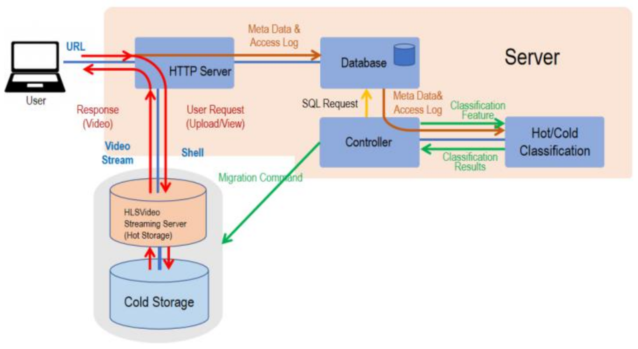

Figure 1 Our assumed system architecture for video-on-demand services using hot/cold tiered storage system

cold storage device. The HTTP server includes a streaming distribution module and a hot/cold classification module. The data access speed of the storage devices for the hot storage is high, like an HDD or SDD, and that for the cold storage is low, like Blu-ray discs.

3.1 HTTP Server

The HTTP server connects to the Internet and provides a web page for the video-on-demand service. The web page shows the video files that the system has and can provide to users. The HTTP server also manages two databases. One is for storing the meta data for the video files such as the data size, duration, and bit rate. We call this database the "file database." The other is for storing access logs. This database is used for getting the popularity of the files. We call this database the "access log database." The system can use the database services of external cloud services or internal databases for these databases.

3.2 Streaming Distribution Module

The streaming distribution module gets each video data segment from the two types of storage and continuously distributes them to the clients of the users. The details of the streaming distribution module, such as the streaming protocol and the capacity of the data buffer for streaming distribution, is up to the implementation.

3.3 Hot/Cold Classification Module

Data are stored in the hot or cold storage device on the basis of the classification of the hot/cold classification module. In conventional hot/cold tiered storage systems, video files classified as having hot popularity are stored in the hot storage since these files are frequently accessed. Files classified as having cold popularity are stored in the cold storage. In our proposed method, the first data segments of the video files are stored into the hot storage and the others to the cold storage.

3.4 Application Scenario

A user who wants to watch a video accesses the web page provided by the HTTP server by using his/her clients such as laptops or smartphones. If the user finds an interesting video on the web page, the user pushes the play button to start playing the video, and the client sends a request for the video to be played to the HTTP server. When the HTTP server receives the request, it finds the requested video data by using the file database and

ⓒ2019 Information Processing Society of Japan

starts the streaming delivery of the video by using the streaming distribution module.

4. Proposed Method

In this section, we first explain our target problem. After that, we explain our proposed division-based video data access method to solve the problem and our implementation of our proposed method.

4.1 Problem of Conventional Method

As described in Section 1, most hot/cold tiered storage systems are designed for general services, and they determine to which storage each video file should be stored on the basis of video playback popularity. Therefore, the data access speed for videos stored in the cold storage is slow. In the case that the data access speed is slower than the data distribution speed, there is a large possibility that long interruptions will be caused in video-on-demand services.

4.2 Division-based Video Data Access Method

The main cause of the problem explained in the previous subsection is storing data to the cold storage device. We can reduce the interruption time by storing video data to fast storage (hot storage device). However, the monetary cost of hot storage device is high, so this simple solution requires much money.

Therefore, in our proposed division-based video data access method, video files are divided into some data segments. The hot/cold classification module stores the first segment of each video file to the hot storage device and the others to the cold storage device. Since the data access speed of hot storage is fast, the possibility that clients can finish receiving subsequent segments while playing received segments increases. Thus, our proposed method can reduce the interruption time. Also, our method can achieve a lower power consumption and a lower monetary cost since only a part of video files are stored in the hot storage device.

Here, one point to consider for our proposed method is how to divide video files. For this, we divide them so that clients can finish receiving the subsequent data segments while playing the already received data segments. In the next subsection, we calculate the data amount for the first segment.

4.3 Mathematical Analysis

We will show the conditions under which playback interruptions are not caused in this subsection.

4.3.1 Assumptions

To calculate the proper break points of a full video, some assumptions are established as follows.

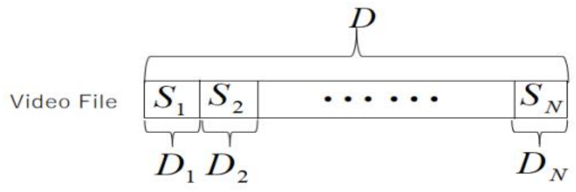

Figure 2 Image of video data division

Table 1 Symbols used

Symbols	Meaning
$T_{initial}$ [sec.]	Elapsed time for moving disc to read position
N	Number of segments of a movie file
M_i [sec.]	Margin of time for transmitting *i*th segment
D [sec.]	Total duration of movie
D_i [sec.]	Duration of *i*th segment
R_i [Mbps]	Bitrate for *i*th segment
B_i^* [Mbps]	Expected bandwidth when transmitting *i*th segment
B_i [Mbps]	Actual bandwidth when transmitting *i*th segment

- The server does not change discs once the transmission of the movie data in a disc starts.
- The server transmits segments sequentially from the cold storage to the hot storage.
- The data of the first segment is initially stored in the hot storage.
- The data of the other segments is initially stored in the cold storage.
- The bandwidth between the cold storage and the hot storage is sufficiently larger than the bandwidth between the server and the client.

4.3.2 Conditions for Playing Segments Continuously

Table 1 shows all the symbols we use in the mathematical analysis section.

The data amount for the *i*th segment is D_iR_i, and the transmission time from the cold storage to the hot storage is D_iR_i/B_i. Under the condition that the client can play segments without interruptions, the elapsed time until starting to play the *j*th segment (j= 2, $\cdots$) from the time to start playing the movie (the first segment) is $D_1+D_2+...+D_{j-1}$. To play the *j*th segment without interruptions after finishing playing the (j-1)th segment, the transmission of the *j*th segment should be completed before the time to start playing the *j*th segment. Therefore, we have the following equation

$$T_{initial} + \sum_{l=2}^{j} \frac{D_l R_l}{B_l} < \sum_{k=1}^{j-1} D_k \qquad (1)$$

4.3.3 Data Segmentation

We propose a method that takes a margin of time M_i to satisfy the above inequality.

$$T_{initial} + \sum_{l=1}^{j} \frac{D_l R_l}{B_l} = \sum_{k=1}^{j-1} D_k - M_j \qquad (2)$$

We get the following equation by transforming this as follows.

$$D_j = \frac{B_j}{R_j}(D_1 + \sum_{k=2}^{j-1}(1-\frac{R_k}{B_k})D_k - T_{initial} - M_j) \quad (3)$$

Ⓒ2019 Information Processing Society of Japan

In actual situations, a server cannot measure B_j when starting to transmit the jth segment. Hence,we have the following.

$$D_j = \frac{B_j^*}{R_j}(D_1 + \sum_{k=2}^{j-1}(1 - \frac{R_k}{B_k})D_k - T_{initial} - M_j) \quad (4)$$

4.3.4 Special Case

When B_i and R_i are constant and the constant values are B and R, the above equation can be transformed to the following.

$$D_j = \frac{B}{R}(D_1 + \sum_{k=2}^{j-1}(1 - \frac{R}{B})D_k - T_{initial} - M_j) \quad (5)$$

This means the following.

$$D_{j+1} = \frac{B}{R}(D_1 + \sum_{k=2}^{j}(1 - \frac{R}{B})D_k - T_{initial} - M_j) \quad (6)$$

(6)-(5) is as follows.

$$\begin{aligned} D_{j+1} - D_j &= \frac{B}{R}(\sum_{k=2}^{j}(1 - \frac{R}{B})D_k - \sum_{k=2}^{j-1}(1 - \frac{R}{B})D_k) \\ &= \frac{B}{R}D_j(1 - \frac{R}{B}) \\ D_{j+1} &= \frac{B}{R}D_j \quad (7) \end{aligned}$$

This is a geometric sequence, and the total summation is represented by the following.

$$\sum_{k=2}^{N} D_k = \frac{(\frac{B}{R})^{N-1} - 1}{\frac{B}{R} - 1} D_2 \quad (8)$$

Note that j=2,···. Therefore,we have the following equation.

$$\begin{aligned} \sum_{i=2}^{N} D_i &= D_1 + \sum_{k=2}^{N} D_k \\ &= D_1 + \frac{(\frac{B}{R})^{N-1} - 1}{\frac{B}{R} - 1} D_2 \quad (9) \end{aligned}$$

From (3),we have the following.

$$D_2 = \frac{B}{R}(D_1 - T_{initial} - M) \quad (10)$$

As can be seen, the following equation is tenable.

$$\sum_{i=1}^{N} D_i = D \quad (11)$$

From (9), (10), and (11), we know that D_1 and M are parameters, and $T_{initial}$ is a constant.

4.4 Implementation

In this subsection, we explain our implementation of the system with our proposed method.

4.4.1 HTTP Server

As mentioned in 3.1, the HTTP server has two tables named “file” and “access log” in the database. There are nine columns in the file table. The fileid column is a unique ID assigned to each video file. The Path column is the path where video data segments are stored in the storage system. Size is the data size of the video files. SizeAfterTrans is the total data size of the segments. Date is the date on which a video file was stored. Duration is the duration of a video file. NumOfClips is the number of segments after video file division. Bitrate is the bitrate of a video file. NumOfBlu is the disc number of the Blu-ray that holds the residual part of a video file. The other table is named “access log,” which is used to save the access log of each video file. It has four columns as follows. Fileid is a foreign key that refers to a file object. Accesstime is the time at which a request comes. IpAddr is the IP address of a requested client. Readytime is the time at which a video is ready to be played.

One of the challenges is how to play segmented video seamlessly. We have tried several methods of dealing with this problem, like using web sockets to do the video streaming or using MSE (Media Source Extensions) to play two or more separate videos sequentially. We found that interruption will happen among the segments. Therefore, we finally implemented an HLS protocol to deal with this problem. HLS is a video streaming protocol based on the HTTP protocol. The HLS protocol is often used for video distribution and is supported by Microsoft Edge, Firefox, and some versions of Google Chrome.

In brief, HLS allows several video files to be played in a row with a playlist. The server does not need all of the segments when a video starts to play. This means that we just need to ensure that the next segment to be played is in the streaming server, on the basis of which we could calculate the proper break point for the video file.

4.4.2 Hot/Cold Storage

We used a computer (CPU: 4 cores at 3.60 Hz, Memory: 24 [GB], OS: Ubuntu 18.04) as the hot storage and another computer (CPU: 4 cores at 3.60 Hz, Memory: 24 [GB], OS: Ubuntu 18.04) as the cold storage. The capacity of the hot storage was 4 [TB] of hard disk, and that of the cold storage was 50 [GB] per Blu-ray disk. They were connected to a HTTP server via 10-Gbps Ethernet. For the data transmission between the hot storage and the cold storage, we used the SCP protocol, which is a widely used data transfer protocol. Also, the data transmission between these two types of storage and the HTTP server was the SCP protocol. The data stored in them were managed by the HTTP server.

4.4.3 Streaming Distribution Module

For video streaming, we applied the Nginx server on the hot storage, instead of establishing a video streaming server separately. This was to eliminate the communication time between the storage and the HTTP server. To use the HLS protocol, we implemented Nginx-rtmp-module on our server. It is also an open-source module that supports video streaming. It supports a wide range of protocols. By using FFmpeg, a HLS

Table 2 Structure of DB

Table Name	Column	Data Type
File	Fileid (Primary key)	AutoField
	Path	CharField
	Size	IntegerField
	SizeAfterTrans	IntegerField
	Date	DateField
	Duration	IntegerField
	NumOfClips	IntegerField
	Bitrate	IntegerField
	NumOfBlu	IntegerField
Access log	Fileid	ForeignKey
	Accesstime	DateField
	IpAddr	IPAddressField
	Readytime	DateField

video stream will be generated in the Nginx source folder. Then, the HTTP server will display this video by using Video.js, an open-source video player.

4.4.4 Video File Classification Method

In previous research, we applied a machine learning method to classify video files into hot (popular) or cold (unpopular). The input features for the learning are the meta information, such as the uploaded date, quality, and access logs. The output is the probability of file access. This is a feasible approach, as is proved in [10].

5. Performance Evaluation

In this section, we show evaluation results for our proposed method.

5.1 Evaluation Experiments

To evaluate the performance of our proposed method, we created three videos by using FFmpeg with different bit rates from the same raw video file. Table 3 shows the specifications of the video files. For the evaluation, we measured the wait time, the used storage size, and the total interruption time. The wait time is the time from when the HTTP server receives the request to play a video file to the time when the video is ready to play on the client side. We extracted the time when the request arrived from the access log, and we used Ajax to collect the time when the movie was ready to be played from the client. The playback interruption time is the total pause time that occurs while the client plays the video.

Table 3 Detailed Specifications of test files

Filename	Size	Bit rate	Duration	FPS
Rawfile.mp4	4.6 [GB]	21 [Mbps]	1722.75 [sec.]	59 [Hz]
test0.mp4	295.1 [MB]	1 [Mbps]	1722.75 [sec.]	59 [Hz]
test1.mp4	2.2 [GB]	10 [Mbps]	1722.75 [sec.]	59 [Hz]
test2.mp4	4.4 [GB]	20 [Mbps]	1722.75 [sec.]	59 [Hz]

Table 4 Situation details

Situation	Description
1	Using both our division method and saving segments separately into hot/cold storage
2	Using only our division method and putting all segments into hot storage
3	Using only our division method and putting all segments into cold storage
4	Not using division method and instead putting full video into hot storage
5	Not using division method and instead putting full video into cold storage

We used a simulator we developed for the evaluation. The parameters for the simulator were based on our implemented system explained in Section 4. The network bandwidth was 1 [Gbps]. The length of *M* was set as a constant value, 10 [sec.]. D_l was set to 50 [sec.]. $T_{initial}$ was 30 [sec.] according to a document on the cold storage.

5.2 Evaluation Results

5.2.1 Wait Time

We measured the wait time for the five situations shown in Table 4. For each situation, we ran the simulator 100 times and regarded the average value as the result. The results are shown in Fig. 2.

From Fig. 2, Situation 4, in which video files were stored in the hot storage, gives the shortest wait time. However, the system in this situation occupied 2.2 [GB] of the hot storage device as we explain in the next subsection. Situations 1 and 2, in which our proposed method was applied, had a slightly longer wait time than Situation 4. In Situations 4 and 5, it was necessary to finish fetching the files, therefore it took much more time than the other situations.

The used storage sizes are shown in Table 5. The column "hot storage" shows the size of files saved in the hot storage, and the column "cold storage" shows the file sizes for the cold storage. test0.mp4 was divided into two segments and test1.mp4 and test2.mp4a into three segments.

5.2.2 Playback Interruption Time

In this subsection, we show the total playback interruption times for our proposed method. We implemented a software named *Dummy net* to control the bandwidth between the hot and cold storage. We show the interruption times as bandwidth changed from 100 [Mbps] to 1 [Gbps]. In our experimental environment, the maximum bandwidth between the hot and cold storage devices was 1 [Gbps]..

For test0.mp4, even when the network bandwidth was 100 [Mbps], interruptions did not occur. For test1.mp4 and test2.mp4, for which the bit rate was bigger than that of test0.mp4,

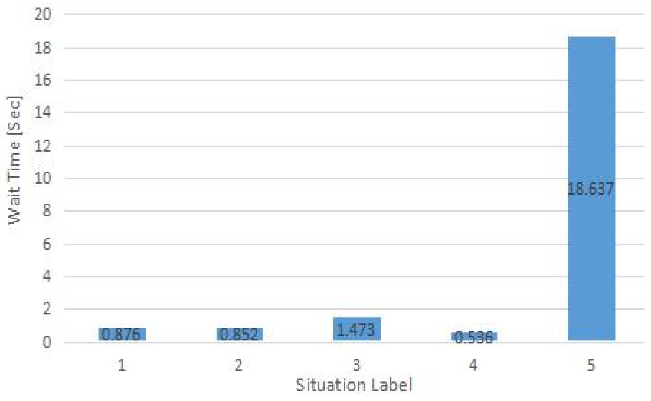

Figure 2 Results for wait time

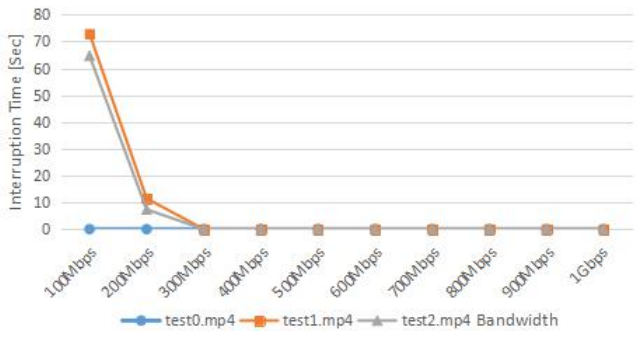

Figure 3 Interruption time of each test file

interruptions occurred when the bandwidth was under 200 [Mbps]. To avoid interruption when network conditions are bad, we could adjust the value of *M*, but this would cause the amount of storage capacity used to increase.

Playback interruptions can also occur in the situation when the hot storage doesn't have enough capacity to store the segments needed by video streaming. For example, if there are three test video files stored in the hot/cold tiered storage system and the video streaming server is requested to stream them at the same time, the capacity of the hot storage should have at least the capacity of 7.3[GB]. Otherwise, playback interruptions occur.

Next, we show the evaluation result of the playback interruption time when the hot storage space is fixed First, we have several assumptions:

1. while multiple files are transmitted between hot storage and cold storage, the network bandwidth will be distributed to them in equal.
2. When the space of hot storage devices is not insufficient, the transmission will be suspended until the hot storage has space to stored the file.
3. The redundant video file in the hot storage will be deleted, when the client has loaded all the video.
4. The network bandwidth between the client and the server is 100Mbps.
5. The output bandwidth of video streaming server is infinity.

We measured the average playback interruption time on the situation:

Table 5 Used storage size for test files

Filename	SizeAfterTrans		Hot storage	Cold storage
test0.mp4	280 [MB]	8.00 [MB]	8.00 [MB]	272 [MB]
		272 [MB]		
test1.mp4	2.40 [GB]	69.1[MB]	69.1 [MB]	2.30 [GB]
		1.50 [GB]		
		858 [MB]		
test2.mp4	4.70 [GB]	137 [MB]	137 [MB]	4.60 [GB]
		1.40 [GB]		
		3.20 [GB]		

1. We have 100 video files to save in the Hot/Cold storage system.
2. All the video file are the same as test0.mp4 and they are all divided as mention in Table 5.
3. The number of video streaming at the same time is 20.
4. All the request has arrived at the same time.

As shown in Fig.5, we measured the average interruption time on the situations explained in the Table 4. When the storage space of the hot storage device is not enough, our method can also keep a low interruption time.

5.3 Discussion

When D_1 was 50 [sec.] and the margin time *M* was 10 [sec.], the ratio of hot/cold storage was approximately 3/100 as shown in Table 4, though the size of some video files will expand after division. As mentioned in Section 4, there are two parameters for our proposed method; a longer D_1 gives a larger storage space, but the number of segments decreases, and a larger *M* gives a larger possibility of avoiding interruptions.

6. Conclusion

In this research, we proposed a division-based video access method for hot/cold tiered storage systems. The main research objective was reducing the playback interruption time for video-on-demand services while reducing the monetary cost and power consumption of data centers. In our proposed method, video files are divided into data segments, and the first segment is stored in hot storage. Since the data access speed of hot storage is fast, our proposed method can reduce the interruption time.Our evaluation results revealed that our proposed method

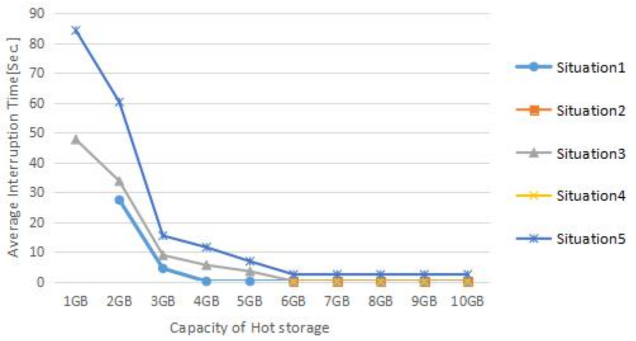

Figure 5 Results for average interruption

can reduce the interruption times and storage space used. A smaller storage space used leads to a lower power consumption for hot/cold tiered storage systems.

In the future, we will integrate a hot/cold classification module that classifies video files by using machine learning techniques and investigate its performance in practical situations.

Acknowledgements

This research is an achievement of Fujitsu Next Generation Cloud Research Alliance Laboratories (FCRAL) established at Cybermedia Center of Osaka University by Fujitsu Ltd. And Fujitsu Laboratories Ltd., and in part supported by a Grants-in-Aid for Scientific Research(C) numbered JP18K11316.

References

[1] R. Irie, S. Murata, Y.-F. Hsu, and M. Matsuoka, "A Novel Au tomated Tiered Storage Architecture for Achieving Both Cost S aving and QoE," in Proc. of the IEEE 8th International Sympo sium on Cloud and Service Computing (SC2), pp. 32–40, 201 8.

[2] "Introducing the View of SNIA Japan Cold Storage Technical Working Group on 'Cold Storage'," available at https://www.sn ia.org/sites/default/files/SDC/2016/presentations/cold_storage_data/ Kazuhiko%20Kawamura_Intro_SNIA_Japan_ColdStorage_rev1-11. pdf.

[3] "Redefining Archival Cloud Storage," available at https://cloud. google.com/storage/archival/.

[4] "Cold Storage: The Road to Enterprise," available at https://w ww.pcworld.com/article/138102/article.html.

[5] R. T. Kaushik and M. Bhandarkar, "GreenHDFS: Towards An Energy-conserving, Storage-efficient, Hybrid Hadoop Compute Cluster," in Proc. of the USENIX Annual Technical Conferenc e, vol. 109, p. 34, 2010.

[6] V. S. Marco, Z. Wang, and B. Porter, "Real-time Power Cycli ng in Video on Demand Data Centres Using Online Bayesian Prediction," in Proc. of the IEEE 37th International Conference on Distributed Computing Systems (ICDCS), pp. 2125–2130, 2017.

[7] Y. Hsu, R. Irie, S. Murata, and M. Matsuoka, "A Novel Auto mated Cloud Storage Tiering System through Hot-cold Data Cl assification," in Proc. of the IEEE 11th International Conferenc e on Cloud Computing, pp. 492–499, 2018.

[8] M. Gu, X. Li, and Y. Cao, "Optical Storage Arrays: A Perspe ctive for Future Big Data Storage," Light: Science & Applicati ons, vol. 3, no. 5, p. e177, 2014.

[9] "Tested: New Hybrid Hard Drives from Samsung and Seagat e," available at https://www.pcworld.com/article/138102/article.ht ml.

[10] M. Dayarathna, Y. Wen, and R. Fan, "Data Center Energy C onsumption Modeling: A Survey," IEEE Communications Surve ys & Tutorials, vol. 18, pp. 732–794, 2016.

ⓒ2019 Information Processing Society of Japan