future internet

MDPI

Article

Replicas Strategy and Cache Optimization of Video Surveillance Systems Based on Cloud Storage

Rongheng Li [1], Jian Zhang [1] and Wenfeng Shen [1,2,*]

[1] School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China; lirongheng@shu.edu.cn (R.L.); zhangjian@shu.edu.cn (J.Z.)

[2] Institute for Advanced Communication and Data Science, Shanghai University, Shanghai 200444, China

* Correspondence: wfShen@mail.shu.edu.cn; Tel.: +86-152-2134-6769

Received: 17 March 2018; Accepted: 8 April 2018; Published: 10 April 2018

Abstract: With the rapid development of video surveillance technology, especially the popularity of cloud-based video surveillance applications, video data begins to grow explosively. However, in the cloud-based video surveillance system, replicas occupy an amount of storage space. Also, the slow response to video playback constrains the performance of the system. In this paper, considering the characteristics of video data comprehensively, we propose a dynamic redundant replicas mechanism based on security levels that can dynamically adjust the number of replicas. Based on the location correlation between cameras, this paper also proposes a data cache strategy to improve the response speed of data reading. Experiments illustrate that: (1) our dynamic redundant replicas mechanism can save storage space while ensuring data security; (2) the cache mechanism can predict the playback behaviors of the users in advance and improve the response speed of data reading according to the location and time correlation of the front-end cameras; and (3) in terms of cloud-based video surveillance, our proposed approaches significantly outperform existing methods.

Keywords: video surveillance system; cloud storage; replicas strategy; cache optimization

1. Introduction

With the arrival of the Big Data era, the amount of video data in modern society is growing at an unprecedented speed; especially in the field of video surveillance, there has been an explosion in the total amount of video data [1]. Data has been an essential resource, and how to better manage and use it has become crucial. The scale effect of Big Data brings extraordinary challenges to data storage and data analysis.

The issue of massive video data processing is a common problem in Big Data technology; especially with the deepening of the construction of the city, video surveillance technology has received more and more attention from the community. It is of great theoretical and practical significance to study how to realize the storage and retrieval of video data effectively. Storage technology [2–4] is a vital issue in video surveillance systems. The cloud storage of video surveillance is a research hotspot. Researchers have done a lot of work and achieved some results. However, there are still some challenges. The number of replicas of video files in the distributed system is a man-made definition [5,6]. Hadoop Distributed File System (HDFS) has a default number of copies. If we save the data according to the default number, we need massive storage space. This mechanism can effectively guarantee the security of video data. However, it cannot accurately reflect the security level of the video file in many application environments, and the redundant data always waste storage resource. The deduplication checks (i.e., the corresponding essential message exchange) create a side channel, exposing the privacy of file existence status to the attacker. The random response (RARE) approach achieves stronger privacy. Both deduplication benefit and privacy of RARE can be preserved [7].

Besides, when users play back video data to track a target's action path, how to predict the target's action path and add the video data to the cache in advance is a significant question.

For this reason, the purpose of this paper is to optimize replicas strategy and data cache strategy in cloud-based video surveillance systems. Firstly, we propose a dynamic redundant replicas mechanism based on security levels, which dynamically adjusts the number of redundant replicas. Secondly, we design a more realistic cache mechanism according to the position and time correlation of the front camera. The following sections have detailed descriptions of two improvements and validate in the actual applications.

The rest of this paper is organized as follows: Section 2 describes several studies closely related to our research. Section 3 introduces the architecture of the cloud-based video surveillance system. Sections 4 and 5 describe the dynamic redundant replicas mechanism based on security levels and data caching strategy based on location correlation in details. Experimental verification and analysis can be found in Section 6. Finally, Section 7 concludes this paper and states future research options.

2. Related Work

With the development of cloud computing technology, researchers have put forward some new system architectures based on cloud computing technology to meet the growing demand for video surveillance system. Neal et al. [8] proved cloud computing could be a new deployment solution for video surveillance system. Karimaa [9] investigated the reliability cloud-based video surveillance technology, but there is no real breakthrough in data management. In [10], Lin et al. proposed a video surveillance system under IaaS abstraction layer. The system is based on the Hadoop distributed file system to provide scalable video recording and backup capabilities. M. Anwar Hossain analyzed the suitability of cloud-based multimedia surveillance systems and proposed a cloud-based multimedia surveillance system framework [11–13]. The system can efficiently deal with system overload, meet the storage requirements of the large-scale monitoring system, and provide data access to users. Bao et al. proposed the Racki selection algorithm and DNik selection algorithm to guarantee the cluster load balance [14]. The two algorithms can fully consider cluster load balance when providing the data replicas pre-written into HDFS reasonable DataNodes. However, many unimportant data occupy a lot of space resource. Big data stream mobile computing (BDMSC) has some challenges in performing real-time energy-efficient management of the distributed resources available at both mobile devices and Internet-connected data centers. A fundamental problem for coping with the variable volume of data generated by the emerging BDMSC applications is the design of integrated computing-networking technological platforms [15]. Xie et al. [16] proposed a replicas mechanism based on computing capacity of each node in HDFS. Similarly, inactive data in the system still causes the consumption of storage resources. Mauro Conti et al. [17] proposed a distributed Fog-supported IoE-based framework. This approach saves energy consumption impressively in the Fog Data Center compared with the existing methods and could be of practical interest in the incoming Fog of Everything (FoE) realm. Xiong et al. [18] implemented a hotness-proportional replication strategy (HP) to improve the efficiency of storage space. Wei et al. [19] presented a cost-effective, dynamic replication management scheme referred to as CDRM. CDRM maintains minimal replica number for a given availability requirement. However, they did not address the replica replacement issue. Najme [20] presented adaptive data replication strategy (ADRS), including a new replica placement and replacement strategy. They implemented ADRS and evaluation results apparently show that ADRS can improve the performance of cloud storage. However, in the massive video surveillance storage system, the retention time of video file will also affect the replica factor, so we should consider this factor.

Niu et al. [21] proposed a cache mechanism based on multi-agents that can automatically manage related video streams. Besides, they designed a caching framework to implement the cache interaction, cache activity control, and video streaming migration. However, the framework is too complicated and costly. Rejaie and Kangasharju [22] used a prefetching mechanism to support

higher quality cache streaming during subsequent playback of the hierarchically encoded video stream. However, when users play back the video, they pay more attention to the speed of data querying. Zhang L. et al. [23] presented a Cost-Effective Cloud Storage Caching Strategy referred to as CloudCache. They first utilized nearly free desktop machines in a local area network environment to build a local distributed file system, which is deployed as a data cache of remote cloud storage service. After that, they presented a cache replacement and file reading/writing algorithm. Performance evaluation is accomplished, using Amazon Simple Storage Service (S3) and desktop PCs in the laboratory to build the experimental environment. The evaluation results apparently show that CE-cache strategy reduces cost and dramatically improves file response speed. However, they did not address the cache size issue. Users need to define the cache size according to their needs.

3. Framework for a Cloud-Based Video Surveillance System

We proposed a cloud-based video surveillance system in this paper to meet the construction needs of the current video surveillance system and improve the scalability of video surveillance systems in the future. Figure 1 shows the hardware platform design of the video surveillance system; general cloud-based video surveillance system can divide into three parts.

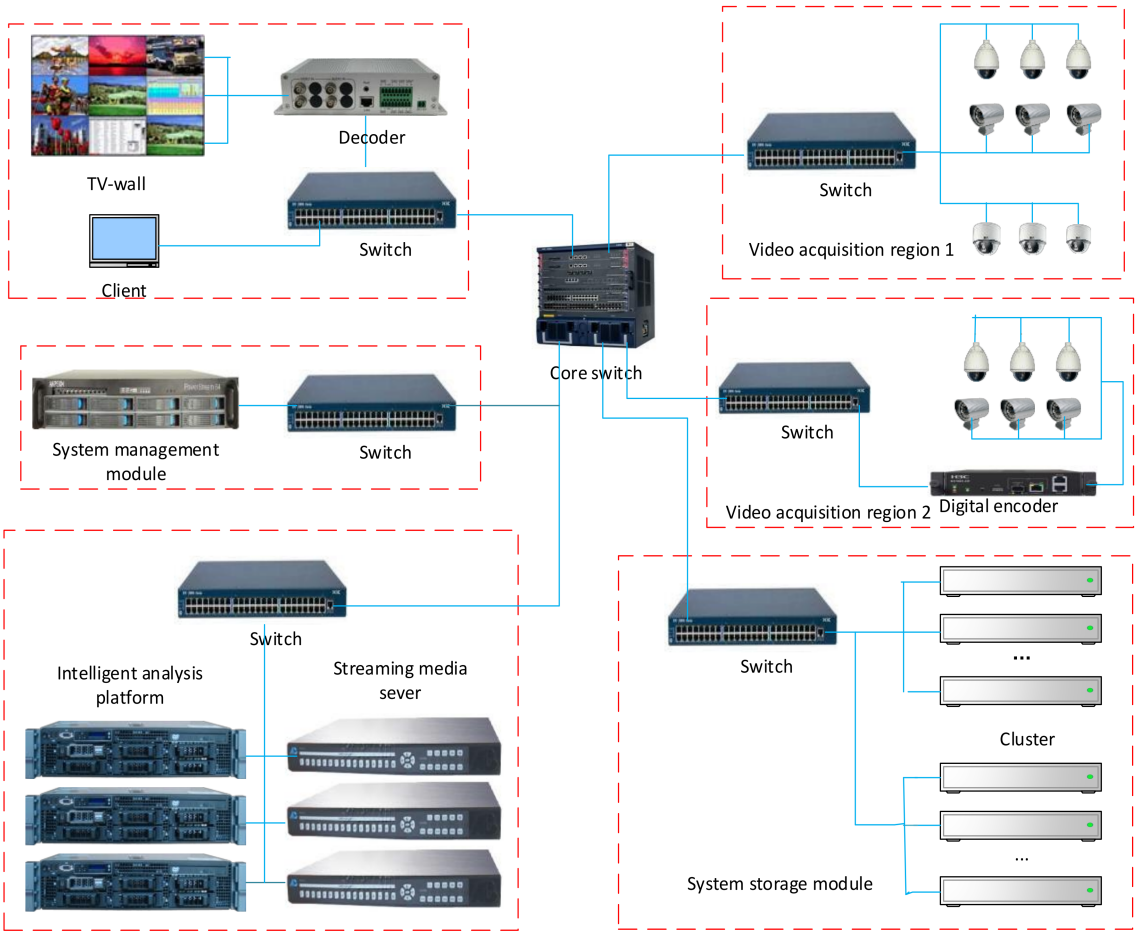

Figure 1. Cloud-Based video surveillance system platform.

3.1. Video Acquisition Area

It is the basis of the video surveillance system, which records video in the front area and sends video data through the network protocol. During the current period, the surveillance system's cameras support coding and transmission protocols are not the same. HD digital cameras have a higher

share in the market at present, and most HD digital cameras use H.264 encoding and support RTSP protocol transmission. Besides, there are more and more cameras using SVAC coding and transmission protocol [24].

3.2. Surveillance Centers

In general, video walls are the most common surveillance center for surveillance video. The surveillance client mainly facilitates person managing surveillance system.

3.3. Background Processing

The core of a video surveillance system mainly contains the following modules:

- Streaming media server module: provides mostly users a unified data access interface, includes real-time video access, historical video playback, download and data transmission protocol.
- Intelligent video analysis module: this module analyses the real-time video streams and detects an object if entering with intelligent video analysis algorithms.
- System storage module: mainly includes data storage interface, and data read interface. Data storage interface can store real-time video in a distributed storage system and establish the index of the video data, then copy multiple redundant replicas. The data read interface obtains historical video data from distributed storage system through searching the index in management module and storage module. The module integrates heterogeneous storage resources and provides a unified excuse for reading and writing. Not only can it reduce the complexity of the system, but also improve the scalability and compatibility of the system [25].
- System management module: mainly includes equipment management, resource allocation. Equipment management can monitor the state of all devices; resource management can integrate all resources in the system, and then allocate resources based on the actual needs of different applications.

4. Dynamic Redundant Replicas Mechanism Based on Security Levels

To ensure the security and storage time of the video data segments, it is necessary to store several data replicas in different nodes when storing the video data segment. However, in various situations, the requirements of the security for video files are different. Even in the same scene, security requirements may differ in different monitoring locations. For example, a checkout counter needs a high security level. Users [5,6] define the replicas mechanism in the distributed surveillance system, and users set the number of redundant replicas in advance. This mechanism can effectively guarantee the security of the video data. However, the coarse-grained redundant replicas mechanism does not accurately reflect the security level of video files [26]. Keeping meaningless video data will cause the storage resources to be wasted. Besides, man-made definition replicas mechanism may lose critical data.

From the above analysis, we put forward a dynamic redundant replicas mechanism based on the video file security levels (SL-DRM). It keeps different redundant replicas based on different security levels. The security level can be dynamically adjusted according to some influencing factors. In the current surveillance field, the factors affecting the number of replicas of the video data only are the retention time, man-made definition, access hit of video files, and intelligent video analysis results. Intelligent video analysis directly indicates whether the video file has saved value; when the video data is saved for the first time, the user-defined security level shows the importance of the data to the user. With the increase of the retention time, the security level of video data needs to consider the retention time, and access hit. Access hit can measure the importance of video files to users; storing redundant copies for a long time may result in the consumption of a significant amount of storage space. Hence, the retention time also has a considerable influence on the video security level. There is

no direct correlation between the factors, but their changes affect the security level of the video file. Table 1 shows the notations that we will use.

Table 1. Notations Definition.

Notations	Definition
x	video file x
n	number of redundant replicas
S_{x-init}	initial security level of video file x
S_x	security level of video file x
I_x	security level defined by users of video file x
A_x	intelligent video analysis result of video file x
m_i	number of accesses to the i-th replica
H_x	access hit of video file x
T_x	effects of retention days on the security level of video file x
t	retention days of video file x
∂_i	Constant parameter

The replica management strategy can divide into static and the dynamic replica management strategy based on the number of copies and the fixed position of the replica placement. In the current business system, the static replica strategy is adopted; for example, HDFS creates three copies of each file by default in the number of backups. The cost of data management would increase with the number of replicas increasing. Too many copies may significantly enhance the availability, but it also brings unnecessary consumption [20]. Hence, we take the default copy number of HDFS as the maximum number of copies and define four security levels for video files. As shown in Table 2, Level 0 is the lowest level of security, and it does not retain redundant replica of the video file; level 1 only saves one redundant replica of the video file; level 2 and level 3 increase the number of redundant replicas in turn. Then, according to the various factors that affect the security level of video files, the security levels of each file will downgrade or upgrade through some formulas.

Table 2. Security level definition.

Number	Level	Demotion	Upgrade	Redundant Replicas
1	0	N	Y	0
2	1	Y	Y	1
3	2	Y	Y	2
4	3	Y	N	3

When calculating the security levels of video files, we mainly consider the following four factors:

1. I_x represents User-specify security levels. It can be divided into three levels: lower, normal, and important. As shown in Table 3, 0 represents lower, 1 represents normal, and 2 represents important.
2. T_x denotes effects of retention days on video file x security levels. As the number of days of video data storage increases, the security levels of the data will gradually decrease, but the video file does not need to be considered for the first time.
3. H_x denotes the access hit of the video file x.The more frequent access to video files, the more critical the video files, and the higher the security levels are.
4. A_x denotes the analyzing result for the video file x by using the method introduced in [27].

The intelligent video analysis algorithm used in this paper is the Edge Frame Difference and Gaussian Mixture Model algorithm (EFD-GMM) [27]. EFD-GMM is an improved algorithm based on the Edge Frame Difference and Gaussian Mixture Model. It can model the moving objects and detect the moving objects. The algorithm can help to solve the problem of noise, illumination, and error

target. This algorithm can analyze the moving target in video data. For the results of the intelligent video analysis, if the target is detected in video file x, $A_x = 1$; Otherwise the $A_x = 0$.

Table 3. User-specify security levels definition.

Number	Level	Security Definition
1	0	lower
2	1	normal
3	2	important

There is no retention time, access hit and other factors when the video saved for the first time. Hence, users determine security level of video file and specify security levels and intelligent video analysis results. We propose Formula (1) to calculate the initial security level of the video file. Here, A_x has two kinds of values, 0 means no value and 1 means valuable.

$$\begin{cases} S_{x-init}=I_x+A_x \\ A_x \in \{0,1\} \end{cases} t = 1 \tag{1}$$

With the video files' retention time growing and the number of accesses increasing, the security level will change relatively. Firstly, we give Formula (2) to calculate the access hit of the video file. As shown in Formula (2), H_x will increase as the number of visits increases, also $H_x < 1$.

$$H_x(m) = \frac{\sum_{i=1}^{n+1} m_i}{1 + \sum_{i=1}^{n+1} m_i} \quad m_i \geq 0 \tag{2}$$

Keeping multiple redundant replicas for a long time can cause a lot of consumption of storage system. As the retention time increases, the effects on the security level will higher, at the same time the security level of the video file will downgrade. Hence, t and T_x are positively related. The effects of retention time on security level can be defined in Formula (3). The reason we take the logarithmic relationship is that logarithm does not change the nature of the data and the correlation, but the scale of the variable is compressed, such as $800/200 = 4$, but $\log 800/\log 200 = 1.2616$, the data is more stable, and it also weakens the collinearity and heteroscedasticity of the model. The logarithmic relationship dramatically reflects the impact of factors on variables.

$$T_x(t) = \log_2\left(\frac{t}{2}\right) \; t > 1 \tag{3}$$

In summary, we have considered all the factors that affect the security level of the video file. Finally, we get Formula (4) to calculate the video file security level. The size of the ∂_i needs to consider the application scenario. Users can define weights according to their own needs such as in the case of more cold data, ∂_3 occupies a more significant value; if it is a high-access system, ∂_2 holds significant weight. They can set a higher value on their more concerned factor which makes the proposed strategy adaptable. As T_x plays a negative role in the retention time increasing, so $\partial_3 < 0$.

Using the linear relationship can directly reflect the impact weights of the influencing factors under different scenarios.

$$\begin{cases} S_x = \lceil \partial_1 S_{x-init} + \partial_2 H_x + \partial_3 T_x \rceil \\ \partial_1 > 0, \; \partial_2 > 0, \; \partial_3 < 0 \end{cases} \tag{4}$$

Assume there are w video streams writing data into the storage system. Each stream will write block files, file collection is set as $rm - x = \{x_1, x_2, \ldots, x_w\}$, I_x for each stream is $rm - I_x = \{I_1, I_2, \ldots, I_w\}$. Though the intelligent video analysis results, we can get $rm - S_{x-init} = \{S_{1-init}, S_{2-init}, \ldots, S_{w-init}\}$ and determine the number of initial redundant replicas. Then with the changes in retention time and access hit, redefine the number of redundant replicas, set $NRR = \{n_1, n_2, \ldots, n_w\}$. The security levels algorithm is proposed as Algorithm 1.

Algorithm 1 Security levels algorithm.

Information: $rm - x = \{x_1, x_2, \ldots, x_w\}$
Input: $rm - S_{x-init} = \{S_{1-init}, S_{2-init}, \ldots, S_{w-init}\}$
Output: number of redundant replicas
for each x_i in $rm - x$ do
    get retention time from storage module
    calculate T_x by Formula (3)
    get number of accesses from storage module
    calculate H_x by Formula (2)
    get S_{x-init} from $rm - S_{x-init} = \{S_{1-init}, S_{2-init}, \ldots, S_{w-init}\}$
    calculate S_x by Formula (4)
    compare S_x with Table 2
    put the number of redundant replicas into $NRR = \{n_1, n_2, \ldots, n_w\}$.
end for
return $NRR = \{n_1, n_2, \ldots, n_w\}$.

5. Data Cache Strategy Based on Location Correlation

Distributed cache technology is mainly used to improve the response speed of data reading of storage system, which is a significant way to improve the performance of the distributed system. In the distributed storage system, a suitable data cache mechanism can predict the user's data reading behavior according to the running time of the system, and the data is read into the buffer in advance. It can increase the speed of data reading.

When a surveillance client plays back particular time of historical video and tracks related targets, it usually begins in a specific location then tracks the target's path. It needs to visit the video data in multiple regions, and there is a certain location correlation among these areas. Hence, according to the location correlation of the front camera, when a video file of a camera is accessed, location-correlated cameras' video data segments for the same period will be added to the cache.

To implement the cache mechanism based on location correlation of front camera, we add a cache pool with eight caches in streaming media server. The eight caches have used the method of circulation allocation. When cache pool is full, the highest cache is the priority to be replaced. Also, the size of each cache is 2 MByte. When the video rate is 2 Mb/s, the frame rate is 25 fps, and I frame interval is 12, at least two Group of Pictures (GOP) data segments can be stored.

When a user accesses a video file, the first two GOP data segments are in the cache pool. Then the data can be obtained from the cache directly while data is also read from the storage system according to the index information. Otherwise, search the video file in the storage system according to the indexing. Then the first two GOP data segments and location-correlated cameras' data segments in the same period are added to the cache pool.

To describe the data cache strategy based on location correlation (LC-cache), we set a surveillance area as an example. As shown in Figure 2, there are four exits and twelve cameras. The camera parameters are in Table 4.

The cache strategy must be established based on the location correlation between cameras. Due many cameras are installed in crowded places like bank counters or shops, these cameras may have overlapping coverage areas or the front and back position correlation. Therefore, the correlation among cameras is considered in two ways:

- There is overlap between the cameras;
- There are front and back position correlation between the cameras.

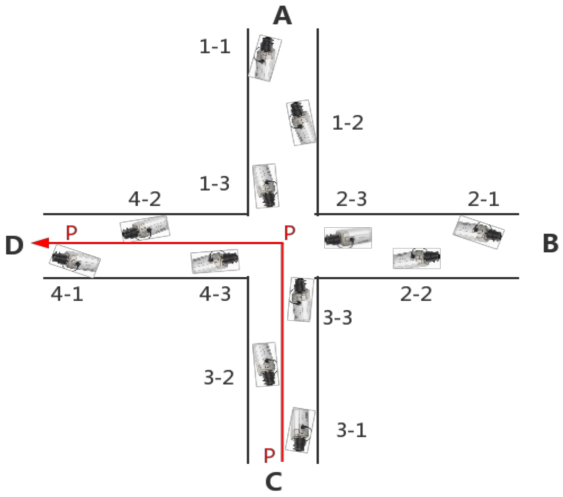

Figure 2. Monitoring area.

Table 4. Camera parameters.

Number	Name	Value
1	camera model	VC-SDI-B3500WE
2	effective pixels	1920 (H) × 1080 (V)
3	aspect ratio	HD:16:9 SD:4:3
4	signal to noise ratio	More than 50 dB
5	electronic shutter	1/30~1/10,000 s

Table 5 records the correlation between cameras in Figure 2. There are overlaps between 1-1 and 1-2, so they are associated. There is no overlap and the correlation between the front and back positions between 1-1 and 1-3, so they are not associated. If someone appears in a camera, he may appear in associated cameras. We simulate a path: pedestrian P enters from exit C, goes through the crossroads and leaves from exit D. Hence, the pedestrian P will appear in the camera 3-1, 3-2, 1-3, 3-3, 4-3, 2-3, 4-1 and 4-2.

Table 5. Camera correlation tables.

Camera Number	Location Correlation
1-1	1-2
1-2	1-1,1-3
1-3	1-2,2-2,2-3,3-2,3-3,4-2,4-3
2-1	2-2
2-2	2-1,2-3
2-3	1-2,1-3,2-2,3-2,3-3,4-2,4-3
3-1	3-2
3-2	3-1,3-3
3-3	1-2,1-3,2-2,2-3,3-2,4-2,4-3
4-1	4-2
4-2	4-1,4-3
4-3	1-2,1-3,2-2,2-3,3-2,3-3,4-2

As shown in Figure 3, the cache pool is empty at first. When the user plays back camera 3-1's video data for the first time, not only the data needs to send to the user, but also the first two GOP data segments need to add to the cache. Besides, camera 3-2's first two GOP data segments for the same period will be added to the cache. The state of the cache pool is shown Figure 3(2).

If users find pedestrian P moving to the camera 3-2 is monitoring area, it will replay camera 3-2's video, at this point cache pool hit. Hence, there is no need to locate the video file according to the indexing system. At the same time, camera 3-1, camera 3-3 are associated with camera 3-2. Since the camera 3-1's GOP data segments have been in the cache pool, it only needs to store camera 3-3, the state of the cache pool is shown Figure 3(3). Then pedestrian P is moving towards camera 3-3's monitoring area and repeats the above process, so camera 1-2, 1-3, 2-2, 2-3, 4-2, 4-3's first two GOP data segments will be added to the cache pool, the state of the cache pool is shown Figure 3(4). Then pedestrian P is moving towards camera 4-2's monitoring area, so the camera 4-1's first two GOP data segments will be added to the cache pool, the state of the cache pool is shown Figure 3(5). At last, P leaves from the exit 4, the cache pool also hits.

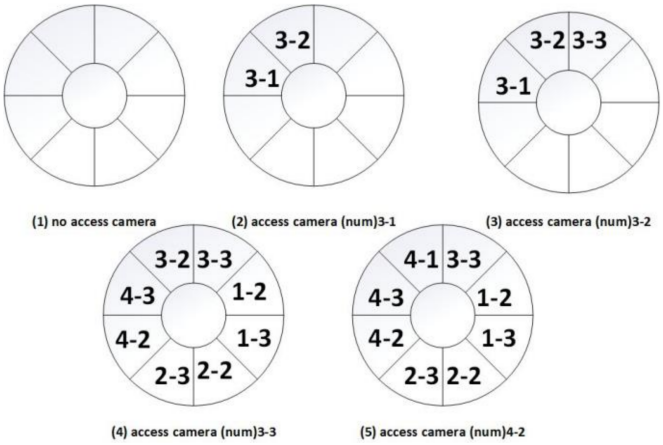

Figure 3. Cache pool status.

6. Experiments and Analysis

In this section, basic environment and discussion of the simulation results are presented.

6.1. Basic Environment

Our experimental platform includes a client, a management server, a streaming media server, 5 cluster hosts and 12 cameras. The client has an I5-4590 CPU, 8 GB RAM, 1 TB disk and a 1 Gbps port. The management server has two E5-2620 CPU, 32 GB RAM, 2 TB disk and two 1 Gbps ports. The streaming media server configures with an I7-4790 CPU, 8 GB RAM, 1 T disk and two 1 Gbps ports. The cluster in the experiment consists of five physical servers, each with an E3-1225 CPU, 4 GB RAM, 2 TB disk and two 1 Gbps ports. The cameras used in our experiments is VC-SDI-B3500WE. The network connection uses a 24-ports switch, with 52 Gbps bandwidth.

6.2. Dynamic Redundant Replicas Mechanism Based on Security Levels

In this experiment, we evaluate dynamic redundant replicas mechanism based on security levels (SL-DRM). The SL-DRM strategy is compared with static replicas mechanism (SRM) and two dynamic replicas strategies (CDRM and ADRS).

We use three cameras, camera-1's I_x is 0, which video file will not be accessed; camera-2's I_x is 1 and camera-3's I_x is 2. Unlike camera-1, we accessed the video file of camera-2 on day 2, 3 and camera-3 on day 2, 3, 8, 9. The cameras are used to test the number of redundant replicas of video files. Then, we select a video file for the same period from different cameras; the size of the video file is 2 G. To ensure the safety of the video file, we defined each file initially hold two redundant replicas in SRM and one replica in CDRM and ADRS respectively. The experiment is repeated five times, which result is showed in Figure 4.

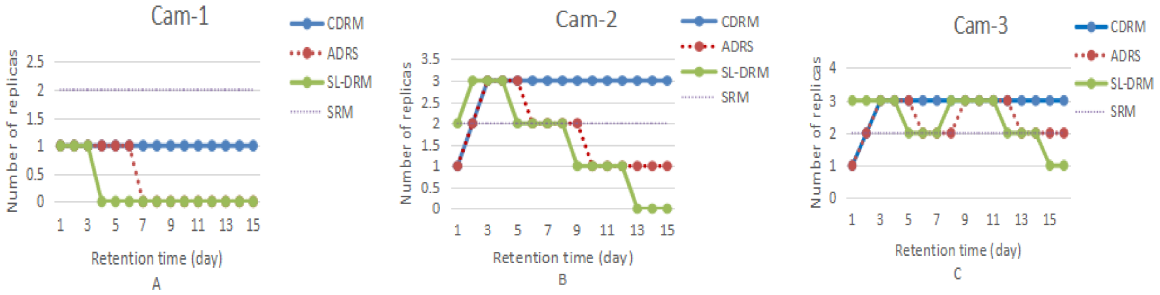

Figure 4. The first set of experiment data records.

For SL-DRM strategy, in the beginning, surveillance system detected that something was passing through the monitoring area. Hence, in SL-DRM strategy, the A_x was 1, security level increased. After that, due video file of camera-1 had not been accessed, S_x also reduced, the camera-1 had not kept redundant replicas. We accessed the video file of camera-2 and camera-3 on day 2, 3. Therefore, the security level went higher, the number of redundant replicas also increased. After that, we had not accessed video file of camera-2, the security level was finally reduced to 0 too. Due to camera-3 being accessed on day 8, 9, the number of redundant replicas is changed from two to three for SL-DRM strategy. As the retention time grew, camera-3 had one redundant replica on the day 15. When the security level rises, the number of replicas will increase again. Otherwise, the redundant replicas will be deleted.

CDRM maintains minimal replica number for a given availability requirement. Availability requirement is defined by users [19]. CDRM strategy can increase the number of replicas but cannot address replica replace issue. ADRS strategy can change the number of replicas, but is not as flexible as SL-DRM. As a result, SL-DRM can adapt to the change of scenario. It guarantees important data security as well as reduces unnecessary replication.

In the next experiment, we investigate the replication rate, which is determined as the ratio of replicas to the total files (original files and replicas). The lower value indicates that strategies are better at saving storage. 50 video files (total file size is around 15 GB) are all from the same camera, and the size of the file is the same. In 15 days, we randomly select seven days to visit all the files. On the 15th day, we count the number of replicas. We repeat this experiment 5 times.

The results of the replication rate are shown in Figure 5. The replication rate of CDRM and ADRS is more than 0.5. That is, under the CDRM strategy [19] and ADRS strategy [20], each file at least have one replica. The SL-DRM strategy considered all the factors affecting the number of replicas of the video file, so the replication rate is less than 0.2. The SL-DRM strategy shows better storage usage performance, compared with other two strategies (CDRM, ADRS).

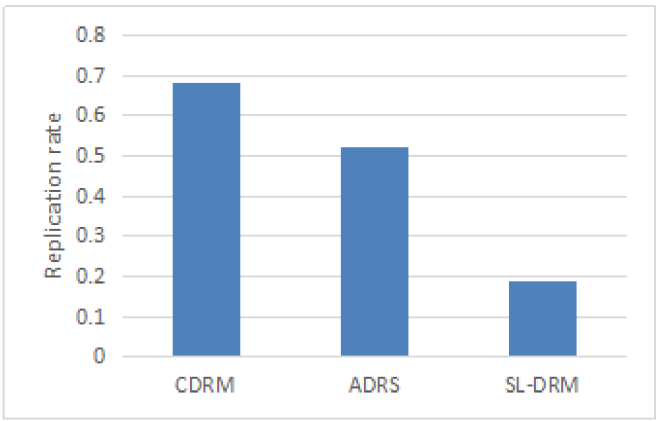

Figure 5. Replication rate of the dynamic replication algorithms.

6.3. Data Cache Strategy Based on Location Correlation

We also evaluate the data cache strategy based on location correlation (LC-cache) and set up two sets of experiments. The first group of experiments is to verify LC-cache strategy can reduce the response time of reading data effectively; the second group of experiments is to verify the hit rate of LC-cache. LC-cache is compared with other two strategies (CloudCache and LRU [28]). The design and results of the two sets of experiments are presented separately.

In CloudCache strategy, as the increase of cache size, the average reading speed is significantly increased, and the hit rate is gradually increased [23]. We set LC-cache strategy's cache pool as mentioned in Section 5. To compare response time and hit rate, the two strategies have the same cache size (16 MByte).

In the first group of experiments, we have implemented the surveillance environment in Figure 2 by using twelve cameras; the cameras are named in accordance with Figure 2. Firstly, according to Table 5, we set the location correlation between the cameras. Then, we design three paths, Path-1: exit C→exit D, Path-2: exit C→exit A, Path-3: exit C→exit B, record and compare the response time of data read when using different cache strategy. Figure 6 shows the first playback response time under different cache strategies.

According to the response time recorded in Figure 6, in the beginning, each cache strategy has no pre-cached data, so there has almost the same response time when access 3-1 camera. After that, we find that LC-cache strategy can improve the response time of data read significantly. According to the location correlation between the cameras, the first two GOP data segments of the video file may be added to the cache pool in advance, so the speed of reading data get significantly improved. The reduced response time is the time to query the location of the video file in the storage system and opens the file to read the data. The LRU cache does not have any data at the beginning of the experiment, nor does the data reading frequently in the experiment. The LRU cache mechanism has almost no effect in this case. Hence, there is nearly no difference between the LRU cache mechanism and no cache mechanism.

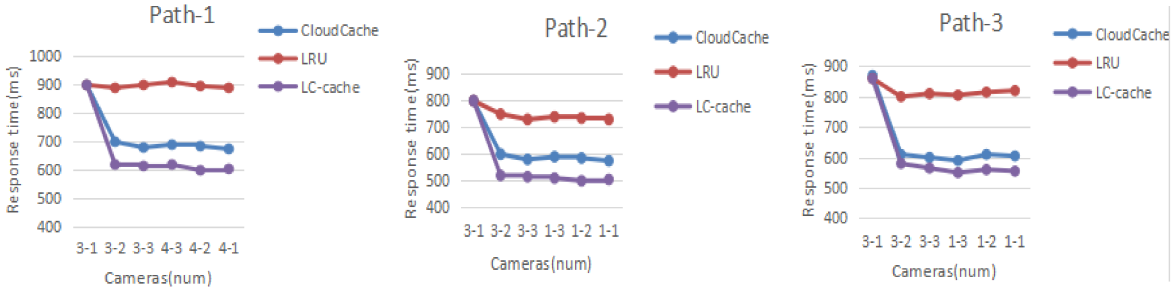

Figure 6. The First Playback Response Time.

After the first experiment, each strategy will cache some data. We repeat the experiment 100 times with path 2, record and calculate the average response time of the camera under different cache strategies, as shown in Figure 7. We find that the LC-cache strategy shows a better efficiency of the data access, compared with the other two strategies (CloudCache and LRU).

In the second group of experiments, we are primarily evaluating the hit rate of the cache pool. We designed four sets of video-on-demand sequences, as shown in Table 6. The first group is the way to track pedestrians, the second group is in random order, and the above two groups are in the same period. Besides, the video-on-demand methods that are used by the other two groups are the same with the first two groups respectively, but the time of playback is also random. We repeat this experiment 100 times. Figure 6 shows the hit rate in each case.

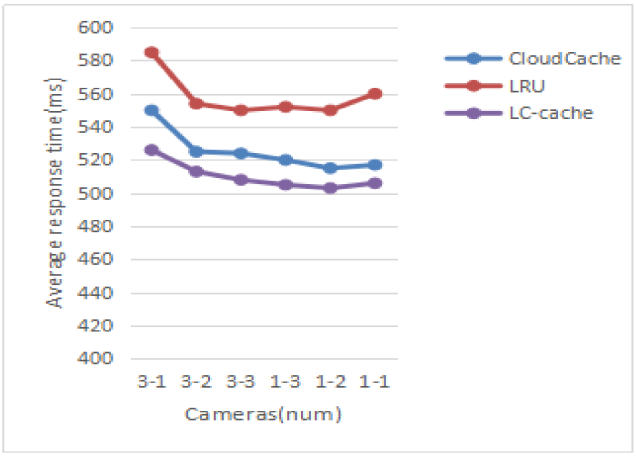

Figure 7. Average Playback Response Time.

Table 6. Video-On-Demand Sequences.

Round	Video-On-Demand Sequences
Round 1	3-1 (9:00),3-2 (9:00),3-3 (9:00),4-3 (9:00),4-2 (9:00),4-1 (9:00),3-2 (9:00),3-3 (9:00)
Round 2	3-1 (9:00),4-3 (9:00),2-2 (9:00),1-1 (9:00),3-1 (9:00),2-2 (9:00),1-1 (9:00),4-3 (9:00)
Round 3	3-1 (9:00),3-2 (9:10),3-3 (9:20),4-3 (9:30),4-2 (9:40),4-1 (9:50),3-2 (9:30),3-3 (9:20)
Round 4	3-1 (9:00),4-3 (9:10),2-2 (9:20),1-1 (9:30),3-1 (9:40),2-1 (9:50),1-1 (9:30),4-3 (9:10)

According to the experimental data in Figure 8, we can see that the LC-cache strategy only has an obvious advantage in the first group while in the other three groups, the cache hit rate does not have any advantage. The reason is that the data cache strategy is mainly based on cameras' locations and time correlation of video file, and predicts the user's playback behaviors in advance. Hence, it is not surprising that the hit rate is lower in the other three groups of experiments. However, in the real surveillance system, the user's playback behavior often has location and time relevance, so the proposed location correlation data cache strategy not only can reduce the response time of data read but also have a higher cache hit rate in the real distributed video surveillance system. Due to the CloudCache strategy not being able to predict the user's playback behaviors in advance, under the fixed cache size, the hit rate is almost unchanged.

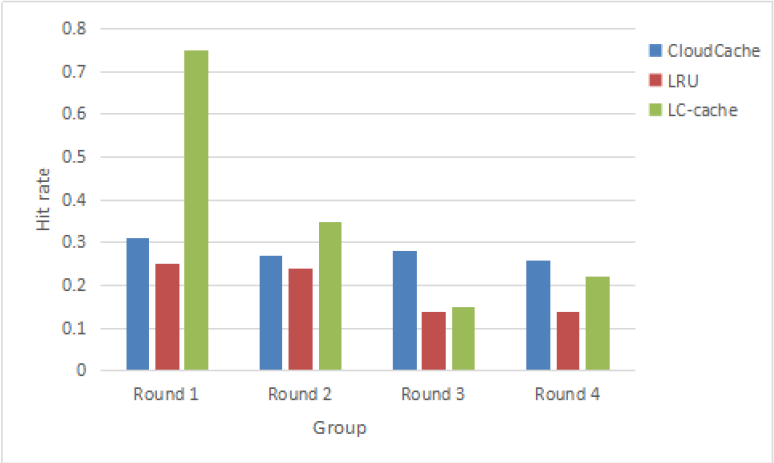

Figure 8. Cache Hit Rate Table.

7. Conclusions

In this paper, the problem of storage and retrieval of cloud-based video surveillance system are discussed. To save storage space while ensuring data security, we present a dynamic redundant replicas mechanism based on security levels. It has been proved that the dynamic redundant replicas mechanism can dynamically adjust the number of redundant replicas according to retention time and access hit. The method ensures the security of video files and appropriates to save storage resources. Performance comparisons show that our approach is effective. In video retrieval, it also has been proved that the data cache strategy based on location correlation can effectively reduce the response time of video playback effectively. The above two strategies are conducive to optimizing storage and retrieval of the massive video surveillance. Their combination can improve the access speed of video data in the case of saving resources. In terms of cloud-based video surveillance, our proposed approaches significantly outperform existing methods. Therefore, the two optimization strategies proposed in this paper provide a solution to the efficient management of massive video data and can meet the needs of the current and future development in distributed video surveillance systems. In future, we would incorporate replica placement strategy with SL-DRM strategy, speeding up the search for system indexes. We believe that our proposed approach can adapt such settings.

Acknowledgments: This work is partially supported by The National Key Research and Development Program of China (No. 2017YFB0701600), Shanghai Innovation Action Plan Project under the grant No. 16511101200, Shanghai University Material Genetic Engineering Institute (No. 14DZ2261200), and the Japan Society for the Promotion of Science under Grants-In-Aid for Scientific Research 15H04678.

Author Contributions: Rongheng Li and Jian Zhang conceived and designed the experiments; Rongheng Li performed the experiments; Rongheng Li and Wenfeng Shen analyzed the data; Wenfeng Shen contributed analysis tools; Rongheng Li wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Devasena, C.L.; Revathi, R.; Hemalatha, M. Video Surveillance Systems—A Survey. *IJCSI* **2011**, *8*, 635–642.
2. Han, J.; Jeong, D.; Lee, S. Analysis of the HIKVISION DVR File System. In Proceedings of the 7th International Conference on Digital Forensics and Cyber Crime (ICDF2C), Seoul, Korea, 6–8 October 2015; pp. 189–199.
3. Lim, K.S.; Park, S.; Han, J. EVM: A New Methodology for Evidential Video Management in Digital CCTV Systems. In *LNEE Future Information Technology, Application, and Service*; Park, J., Leung, V., Wang, C.L., Shon, T., Eds.; Springer: Dordrecht, The Netherlands, 2012; Volume 2, pp. 225–230.
4. Hossain, M.A.; Song, B. Efficient Resource Management for Cloud-enabled Video Surveillance over Next Generation Network. *Mob. Netw. Appl.* **2016**, *21*, 806–821. [CrossRef]
5. Ghemawat, S.; Gobioff, H.; Leung, S.T. The Google File System. In Proceedings of the 19th ACM Symposium on Operating Systems Principles, Bolton Landing, NY, USA, 19–22 October 2003; pp. 29–43.
6. Honnutagi, P.S. The Hadoop distributed file system. *IJCSIT* **2014**, *5*, 6238–6243.
7. Zahra, P.; Kang, C.C.; Chia, M.Y.; Mauro, C. RARE: Defeating Side Channels based on Data-Deduplication in Cloud Storage. In Proceedings of the IEEE INFOCOM Workshops CCSNA, Honolulu, HI, USA, 15–19 April 2018.
8. Neal, D.; Rahman, S. Video Surveillance in the Cloud? *Int. J. Cryptogr. Inf. Secur.* **2015**, 58–61. [CrossRef]
9. Karimaa, A. Video surveillance in the cloud: Dependability analysis. In Proceedings of the 4th International Conference on Dependability (DEPEND'11), Nice, France, 21–27 August 2011; pp. 92–95.
10. Lin, C.F.; Yuan, S.M.; Leu, M.C.; Tsai, C.T. A framework for scalable cloud video recorder system in surveillance environment. In Proceedings of the 2012 9th International Conference on Ubiquitous Intelligence Computing and 9th International Conference on Autonomic Trusted Computing (UIC/ATC), Fukuoka, Japan, 4–7 September 2012; pp. 655–660.
11. Hossain, M.S.; Hassan, M.M.; Qurishi, A.I.M.; Alghamdi, A. Resource Allocation for Service Composition in Cloud-based Video Surveillance Platform. In Proceedings of the 2012 IEEE International Conference on Multimedia and Expo Workshops, Melbourne, Australia, 9–13 July 2012; pp. 408–412.

12. Hossain, M.A. Framework for a Cloud-Based Multimedia Surveillance System. *Int. J. Distrib. Sens. Netw.* **2014**, *10*, 1–11. [CrossRef]
13. Hossain, M.A. Analyzing the Suitability of Cloud-Based Multimedia Surveillance Systems. In Proceedings of the Companion Publication of the 2013 IEEE International Conference on High Performance Computing and Communications (HPCC) & 2013 IEEE International Conference on Embedded and Ubiquitous Computing (EUC), Porto, Portugal, 21–23 October 2013; pp. 644–650.
14. Bao, G.; Yu, C.; Zhao, H.; Luan, Y. Researching on the Placement of Data Replicas in the System of HDFS Cloud Storage Cluster. In *Proceedings of 2013 Chinese Intelligent Automation Conference;* Springer: Berlin/Heidelberg, Germany, 2013; pp. 259–269.
15. Baccarelli, E.; Cordeschi, N.; Mei, A.; Panella, M. Energy-efficient dynamic traffic offloading and reconfiguration of networked data centers for big data stream mobile computing: Review, challenges, and a case study. *IEEE Netw.* **2016**, *30*, 54–61. [CrossRef]
16. Xie, J.; Yin, S.; Ruan, X.; Ding, Z. Improving MapReduce performance through data placement in heterogeneous Hadoop clusters. In Proceedings of the 2010 IEEE International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum, Atlanta, GA, USA, 19–23 April 2010; Volume 66, pp. 1322–1337.
17. Pooranian, Z.; Shojafar, M.; Conti, M.; Chiaraviglio, L.; Naranjo, P.G.V. A Novel Distributed Fog-Based Networked Architecture to Preserve Energy in Fog Data Centers. In Proceedings of the 2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), Orlando, FL, USA, 22–25 October 2017; pp. 604–609.
18. Xiong, R.; Luo, J.; Dong, F. Optimizing data placement in heterogeneous Hadoop clusters. *Clust. Comput.* **2015**, *18*, 1465–1480. [CrossRef]
19. Wei, Q.; Veeravalli, B.; Gong, B.; Zeng, L.; Feng, D. CDRM: A Cost-effective Dynamic Replication Management Scheme for Cloud Storage Cluster. In Proceedings of the 2010 IEEE International Conference on Cluster Computing, Heraklion, Greece, 20–24 September 2010; pp. 188–196.
20. Najme, M. Adaptive data replication strategy in cloud computing for performance improvement. *Front. Comput. Sci.* **2016**, *10*, 925–935.
21. Niu, W.; Li, G.; Tong, E.; Yang, X.; Chang, L.; Shi, Z.; Ci, S. Interaction relationships of caches in agent-based HD video surveillance: Discovery and utilization. *JNCA* **2014**, *37*, 155–169. [CrossRef]
22. Rejaie, R.; Kangasharju, J. Mocha: A quality adaptive multimedia proxy cache for internet streaming. In Proceedings of the 11th International Workshop on Network and Operating Systems Support for Digital Audio and Video, Port Jefferson, NY, USA, 25–26 June 2001; pp. 3–10.
23. Zhang, L.; Tang, B. A Cost-Effective Cloud Storage Caching Strategy Utilizing Local Desktop-Based Storage. In Proceedings of the SpaCCS 2016 International Workshops, Zhangjiajie, China, 16–18 November 2016; pp. 382–390.
24. Wu, J.; Cheng, B.; Wang, M.; Chen, J. Priority-Aware FEC Coding for High-Definition Mobile Video Delivery Using TCP. *IEEE Trans. Mob. Comput.* **2017**, *16*, 1090–1106. [CrossRef]
25. Lianfu, Y. The analysis of critical technology on cloud storage security. In Proceedings of the 2013 International Conference on Computer Sciences and Applications & IEEE Computer Society, Washington, DC, USA, 14–15 December 2013; pp. 26–28.
26. Bin, L.; Jiong, Y.; Hua, S.; Mei, N. A QoS-aware Dynamic Data Replica Deletion Strategy for Distributed Storage Systems under Cloud Computing Environments. In Proceedings of the Second International Conference on Cloud and Green Computing, Xiangtan, China, 1–3 November 2012; pp. 219–225.
27. Li, C.; Su, J.; Zhang, B. Cloud-Based Video Surveillance System Using EFD-GMM for Object Detection. In *Lecture Notes in Computer Science, Proceeding of the ICCCS 2016, Nanjing, China, 29–31 July 2016*; Sun, X., Liu, A., Chao, H.C., Bertino, E., Eds.; Springer: Cham, Switzerland, 2016; Volume 10039, pp. 261–272.
28. Podlipnig, S.; Böszörmenyi, L. A survey of web cache replacement strategies. *ACM Comput. Surv.* **2003**, *35*, 374–398. [CrossRef]