

Video Intelligent Analysis Framework for Edge Computing

Mu Zhao

Beijing University of Technology, Beijing 100000, China

hgdmz1301@163.com

Keywords: Edge computing; Framework; Intelligent analysis; Security monitoring.

Abstract: The vigorous development of the security monitoring and the use of a large number of high-definition cameras have brought difficulties in the storage and calculation of video data. The cloud framework represented by Hadoop has the disadvantages of strong dependence on bandwidth resources and low real-time performance. This paper makes use of the advantages of edge computing in the areas of marginalization and distribution, and proposes an intelligent analysis framework based on edge computing, which transfers the analysis and computing tasks from the cloud to the edge. Using index files to upload the result data of edge nodes reduces the dependence on network resources. The primary key structure with data item semantics is designed to optimize the query speed and reduce the data response delay, thereby improving the real-time performance. Finally, the experiment proves the superiority of the framework in terms of bandwidth and delay.

1. Introduction

With the development of society and economy, security issues have been more and more concerned by departments and enterprises, and put into use numerous surveillance cameras. This situation caused a lot of problems: First, massive video storage and transmission need to consume a lot of resources. Secondly, it is extremely difficult to intelligent analysis these videos. Finally, it takes a long time to store, retrieve, and correlate the original videos and analysis results.

Hadoop can realize the distributed storage, calculation and retrieval of large batches of video data, which can save a lot of expenses for users and create effective economic benefits. However, with the long-term use, the shortcomings of Hadoop gradually emerged: (1) massive videos transmitted from monitoring terminals to cloud centers require a stable bandwidth environment, and consume a lot of time and bandwidth resources [1]. (2) Mass structured and unstructured data is stored in the cluster, which requires a long time for intelligent analysis and result retrieval [2].

For problem 1, ZOU et al. [3] proposes a method of real-time uploading multi-channel video streams with node resource debugging algorithm, and then performs distributed calculation of real-time data; For problem 2, Tian et al.[4] divides tasks into CPU-oriented and I/O-oriented services according to the different requirements of MapReduce tasks for CPU and I/O resources. CPU tasks and I/O tasks are allocated to the same node, making full use of the CPU and I/O resources of a single node, so as to improve the resource utilization of the whole cluster, and then improve the operation efficiency; Based on the combination of inverted table and R-tree, Li et al. [5] proposes a new spatial-temporal index structure, which indexes both spatial attributes and temporal dimensions.

Edge computing was proposed at the beginning of the 21st century, which refers to a distributed open platform that integrates storage, computing, networking, and application on the side close to the data producer to provide the nearer edged intelligent service. The edge computing model migrates some or all of the traditional cloud-based processing tasks to the edge end [6].

In edge computing paradigm, since the computing task is carried out at the edge of the network near the data source, there is no need to upload the data resources, which reduces the dependence on the network bandwidth state. The data exchanged is only a few important result data, thus realizing the real-time service. At the same time, the nodes distributed at the edge of the network analyze the data they own, perform their duties, which can reduce the computational pressure of the cloud computing center, and realize the distributed computing in the true sense; The data processed by edge

computing is the "small data" generated by the network edge, and the processing speed is faster. The data is uploaded to the cloud after analysis and processing, with uniform format and clear organization, which can effectively simplify the process of storage and retrieval and improve operational efficiency.

According to the advantages of real-time and distributed, it can be seen that edge computing can make up for the shortcomings of the traditional centralized data processing method. Adding edge computing to the security field can make monitoring demand response more efficient and fast.

2. Intelligent analysis framework based on edge computing

2.1 The overall framework

This paper designs an analysis framework based on edge computing. As shown in Fig. 1, the framework is mainly composed video sensor, video intelligent terminal, cloud storage server and client.

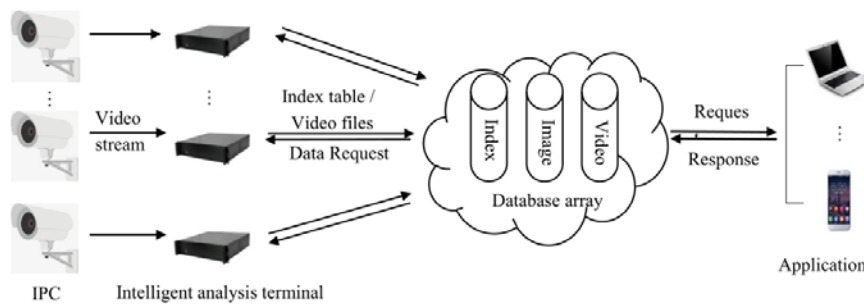


Fig. 1 Intelligent analysis of the overall framework

In addition to obtaining video data streams from the sensor for local storage, the smart terminal also uses smart module to analyze video streams to obtain analysis results and snapshots. The analysis results will form a unique index file, and the terminal uploads the package grouped by index file and image to the cloud. Since index file only contains analysis result and the related description string, the bandwidth required is far less than uploading videos. The cloud server has some databases. When the package is received, the contents of the index file are stored in the index database, and the captured image will be stored in the image database. When receiving the query data from user, the cloud server performs an association search in index database and image database. If the search is not found, cloud will send the search information to all active terminals for analysis of historical and real-time video. When this operation has a result, the terminal will actively report the result data to the cloud server.

2.2 Index file

The index file bears the role of delivering analytical results. When the terminal generates the analysis result, the framework correlates the results with a series of parameters to form an index file.

The specific form of the index file is diversified, such as XML, JSON or a user-defined format file, just match the parsing format of the cloud. As shown in Fig 2, this article takes JSON as an example to display the index file.

The following is a detailed description of what each content represents: (1) Terminal ID. The number of each device in the system that identifies its own identity; (2) Sensor ID. Each terminal has its own sensor, so it is necessary to identify the sensors they managed. (3) Analysis time. The time to analyze, and the accuracy of it depends on the usage scenario; (4) Frame ID. Sequence number of the video frame analyzed in the video, which can locate the position of the frame in the video quickly and play video on demand; (5) Analyzing result. Results from intelligent analysis are related to specific usage scenarios and can include people, cars, behaviors, etc. (6) Combination video path and file name can get the path of the video, which is convenient for quickly finding the original video.

```

{
  "TerminalID": "0001",           //(1)
  "IpcID": "0002",              //(2)
  "TimeStamp": "1553049367673", //(3)
  "FrameID": 1795,              //(4)
  "Result": {                   //(5)
    "P_Num": 5,
    "P_NO.1": {
      "Abscissa":105, "ordinate":208
    }
    "P_NO.2": {
      "Abscissa":270, "ordinate":450
    }
    .....
  },
  "FilePath": "/save/video/20190315/", //(6)
  "FileName": "P1_30.rmvb"         //(6)
}

```

Fig. 2 Index file content display

2.3 Primary key of database

There are three data tables built in the cloud: index table, picture table and video table. The index table mainly stores the contents of the index file, the picture table stores the snapshots picture, and the video table is mainly responsible for storing large-capacity video files or video file information.

After receiving the index file, the cloud will parse the data to obtain the content, and then store it to index table. In the traditional HBase database, the RowKey is independent of the data content. When retrieving, the primary keys and data content are traversed, resulting in a full table scan and reduced retrieval efficiency. Therefore, this paper implements a new RowKey format which embeds the data semantics into it, so that the results can be obtained only by one or more range searches.

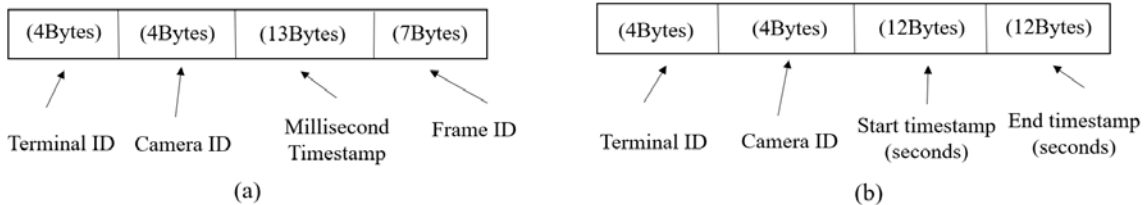


Fig. 3 Primary key structure of index, image and video table

Refer the design concept of RowKey in Reference 7 [7], this paper designs the following RowKey structure. As shown in Fig 3(a), the RowKey of the index and image table is set to 28 bytes, and as shown in Fig 3(b), the video table is set to 32 bytes. For the content of Fig 2, the RowKey is "0001000215530493676730001795", "Result", "FilePath" and "FileName" are inserted into the index table as Column. The RowKey of the image table is the same, but Column is "Image" and the image content is Value. The RowKey of the video table is "0001000215530493561553052956". If video capacity is small, the Column can be video data, but if video capacity is large, the video can be stored in HDFS, and the Column in HBase stores the address of the video in HDFS. This method allows the primary key not only to represent the order of insertion of data items, but also to have a real meaning and make logically adjacent data items are closed physically.

When user retrieves, the cloud first analyses the parameters to combine RowKey, and then searches in the index and image table to obtain the analysis results, photo and location of video. Based on the location obtained, it can be found whether the source video is stored in the video table.

2.4 The delay of framework

To evaluation the delay of frame, this paper assumes the relevant time in the process: Call the time from the client sends the command to the server receives the command t_u , call the time

between the server sends the feedback to the user receives the result t_d , call the time when the data center performs video analysis and obtains the results according to user commands t_r .

For traditional framework, the query process is divided into two cases: (1) The data queried in the database has records, which can be obtained in the database. The total time is called t_{a1} ; (2) The data are not recorded, so find result needs analysis in the cloud to get. The total time is called t_{a2} , so:

$$t_{a1} = t_u + t_c + t_d \quad (1)$$

$$t_{a2} = t_u + t_r + t_d \quad (2)$$

Call the time required to analyze the video in the terminal t_a , and call the total time under this framework described t_b , so:

$$t_b = t_a + t_u + t_c + t_d \quad (3)$$

In intelligent terminal, the analysis of video frames can achieve real-time, therefore, the t_a is in the level of milliseconds and can be ignored. Compared with t_{a1} , t_{a2} and t_b , the main delay occurs in t_c and t_r . By using the semantics of the primary key described in Section 2.3, this framework simplifies the search steps of database, shortens the query time and improves the response efficiency.

3. The result analysis

3.1 Experimental environment

The experimental environment of this paper is as follows:(1) System: Ubuntu 16.04. (2) Software: JDK 1.8, Hadoop 2.7, HBase 1.5. Hadoop are deployed on a cluster of six nodes, each of which is configured with a CPU Intel Core i7 870 @2.93GHz * 8, 4GB of memory and a hard drive of 200GB.

3.2 Bandwidth analysis

This experiment sets the frame rate to 24FPS, and uses three bit rates for testing. The network bandwidth is 50-80Mbps. The video coding format is H.264, the image format is Jpeg.

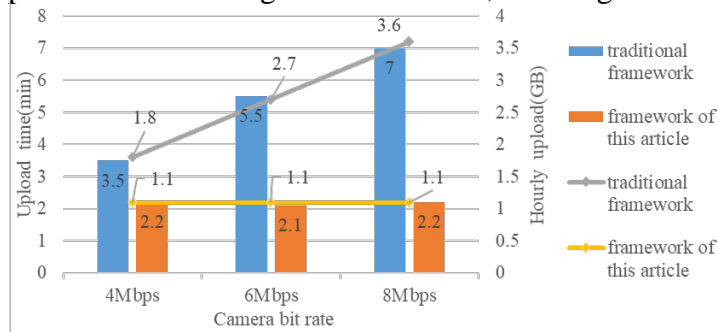


Fig. 4 Hourly uploads and upload time at different bitrate.

As shown in Fig 4, with the traditional framework, the video volume is linearly related to the bit rate. Theoretically, it takes at least 3 minutes for a single camera to transmit an hour's video. For a medium-scale security monitoring system, the number of cameras usually exceeds 1000. Even if the uploading time are staggered, there will still be 50 sensors upload at the same time, and it will take longer and the network load will be heavier.

In this framework, the video is stored and analyzed at terminal, and uploaded file is index file and the captured image, which are usually hundreds of bytes to tens of KB in size. Taking the analysis performance of 5 frames per second as an example, the data to be transmitted per second is about 300KB, which accounts for about 7% of the total bandwidth. Therefore, it can support the real-time upload of more than 100 channels. The amount of data upload per hour is about 1GB, saving nearly 40% of the bandwidth, and the network load is light.

3.3 Delay analysis

For t_{a1} , t_{a2} and t_b , when the bandwidth is the same, the value of t_u and t_d is basically the same, so the main delay gap occurs at t_c and t_r . For t_r , experience shows that computing GB-level video takes roughly minutes [8-9]. Although the value of t_r is related to nodes number, nodes configurations, nodes load, the time is generally in minute level. In theory, increasing nodes number or upgrading nodes configuration can reduce the time for computing, but due to cost and technology limitations, the number of nodes in the system will not be too many, and it is difficult to achieve second-level processing speed. By modifying the structure of the primary key, the HBase in this paper realizes that the primary key is semantically related to the data, and logically the adjacent data items are physically adjacent, reducing unnecessary traversal comparisons, thereby speeding up retrieval.

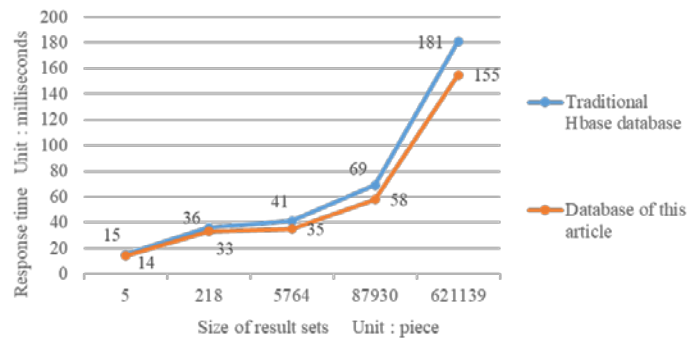


Fig. 5 Retrieval response delay graph of database

By establishing databases and conducting experiments, taking number of result sets as the abscissa and the search response time as the ordinate, a database search response delay graph can be drawn.

It can be seen from the Fig 5 that HBase has a great advantage than traditional databases in querying a large number of result, and can perform hundreds of thousands of pieces of data query in seconds. The primary key optimization scheme used in this paper has a greater advantage in the retrieval response time than the unoptimized HBase, which can achieve 10% efficiency improvement.

4. Summary

This paper summarizes the shortcomings of the cloud computing by discussing the current needs of security monitoring. By introducing the advantages of edge computing, an intelligent analysis framework based on edge computing is proposed. Then, it analyzes the advantages of this framework compared with Hadoop in terms of bandwidth resource and response delay. Experiments show that the framework described in this paper can reduce the dependence on network, lessen the network load and reduce the amount of data upload by about 40%. The semantic-based primary key structure can accelerate the query speed of database, shorten the query response time by about 10%, and realize real-time processing of tasks. In summary, this framework can realize real-time intelligent analysis of video and real-time query response to the analysis results, and has low dependence on network resources and low load, which provides new technical support for security surveillance field.

References

- [1] Zhao Z M, Liu F, Cai Z, et al. Edge Computing: Platforms, Applications and Challenges. *Journal of Computer Research and Development*. Vol. 55 (2018) No. 2, p. 327-337.
- [2] Wu W T, Lin W W, Hsu H, et al. Energy-efficient hadoop for big data analytics and computing: A systematic review and research insights. *Future Generation Computer Systems*. Vol. 86 (2018) No. 1, p. 1351-1367.

- [3] Zou Y B, Sun L Q, Li Y, et al. Video Monitoring and Analysis System for Pig Breeding Based on Distributed Flow Computing. Transactions of the Chinese Society for Agricultural Machinery. Vol. 48 (2017) No. 1, p. 365-373.
- [4] Tian C, Zhou H J, He Y Q, et al. A Dynamic MapReduce Scheduler for Heterogeneous Workloads. 8th International Conference on Grid and Cooperative Computing. Lanzhou, China, 2009, p. 218-224.
- [5] Li Z S, Lee K, Zheng B H, et al. IR-Tree: An Efficient Index for Geographic Document Search. IEEE Transactions on Knowledge and Data Engineering. Vol. 23 (2011) No. 4, p. 585-599.
- [6] Mach P, Becvar Z. Mobile Edge Computing: A Survey on Architecture and Computation Offloading. IEEE Communications Surveys and Tutorials. Vol. 19 (2017) No. 3, p. 1628-1656.
- [7] Wang L, Cheng C Q, Wu S Z, et al. Massive remote sensing image data management based on HBase and GeoSOT. International Geoscience and Remote Sensing Symposium. Milan, Italy, 2015, p. 4558-4561.
- [8] Magrini M, Moroni D, Pieri G, et al. Real Time Image Analysis for Infomobility. International Workshop on Multimedia Understanding Through Semantics, Computation, and Learning. Berlin, Heidelberg, 2011, p. 207-218.
- [9] St-Aubin P, Saunier N, Miranda-Moreno L. Large-scale automated proactive road safety analysis using video data. Transportation Research Part C: Emerging Technologies. Vol. 58 (2018) No. 1, p. 363-379.