

UDC 004.75

Ye.I. DIATLOV*

MODELING OF CLOUD SURVEILLANCE SYSTEM

*Institute of Mathematical Machines and Systems Problems, National Academy of Sciences of Ukraine, Kyiv, Ukraine

Анотація. Стаття присвячена тематиці хмарного відеоспостереження. Автором описуються глобальна розподілена структура системи і її основні компоненти. Стаття висвітлює основні моменти щодо обробки відеопотоку і розподіленої моделі зберігання даних, описуються можливості для розподілу навантаження між рівнозначними сервісами системи. Автор підкреслює важливість реплікації бази даних і її резервного копіювання. Стаття описує основні сервісні моделі надання послуг дата-центрами і їх позиціонування щодо завдань хмарної системи відеоспостереження.

Ключові слова: хмарні системи, розподілені системи, відеоспостереження, відмовостійкість.

Аннотация. Статья посвящена тематике облачного видеонаблюдения. Автором описываются глобальная распределенная структура системы, а также ее основные компоненты. Статья освещает основные моменты касательно обработки видеопотока и распределенной модели хранения данных, описываются возможности для распределения нагрузки между равнозначными сервисами системы. Автор подчеркивает важность репликации базы данных и ее резервного копирования. Статья описывает основные сервисные модели предоставления услуг дата-центрами и их позиционирование относительно задач облачной системы видеонаблюдения.

Ключевые слова: облачные системы, распределенные системы, видеонаблюдение, отказоустойчивость.

Abstract. The article is devoted to the Cloud Surveillance. The author describes global distributed structure of surveillance system and its main components. The article highlights the main points concerning the processing of video and distributed data storage model. The paper describes main abilities to spread the load among equivalent system services. The author underlines the importance of database replications and backup. This article describes the basic service delivery models of data centers and their positioning relative to the cloud video surveillance applications.

Keywords: cloud systems, distributed systems, surveillance, failover.

1. Purpose

The purpose of the article is to introduce nowadays issues and requests in surveillance market. In response to these requests scientists and researchers works on new types of system software, which are able to process huge amount of cameras and tons of information. One of the ways of development is using cloud technologies. The author tries to find all necessary properties of ideal surveillance system and design special system software based on cloud technologies.

2. Introduction

Nowadays cloud computing become more popular and cheaper. That means all resource intensive software products will try to catch a bird with new technological abilities brought by cloud architectures.

Surveillance is one of the most important parts of human's life. Surveillance could prevent a crime, could serve as a base of evidence in court and could make our life better with a different intelligence system, based on video cameras.

On the other hand, surveillance is a resource intensive software product, which should be able to operate 24 hours per day and 7 days per week. Surveillance system should process a lot of terra bytes of video stream: to store, display, analyze. To imagine the scale of the problem we can do some easy calculations. One Full HD (1920x1080 and 25 fps) camera streams around 1 TB video data per month with a middle bitrate 2–3 Mbits/sec. The biggest HDD can store up to 8 TB of raw data (real useful size is around 7,3 TB). In this case one biggest HDD can store only 7 cameras for 1 month. For sure, we can build a huge RAID storage, but what we should do if a storage or network subsystem fails? All our data become inaccessible. We will research this issue bellow.

The next issue is about centralizing. A lot of different failures could happen even in data centers: electricity fails, network fails, maintenance, etc. All these issues force us to promote the benefits of distributed systems.

How to avoid data center failures? The only way is to build distributed and replicated system. If we are talking about video processing or even storage – there is no problem to build a distributed system. But if we are talking about database, which should be common, the only way is to build fully replicated database system. Luckily, most modern data bases support replications.

The purpose of the article is to introduce basic structure of system software which is able to process unlimited amount of cameras. Thus, let's underline key properties of ideal system: distributed, replicated, failover, redundant.

3. Basic structure

In basics, Cloud Surveillance System (CSS) consists of cameras and set of services, which serve for different tasks. The set of main services: connectors, processing nodes, replicated database, distributed file storage and API services illustrates fig. 1.

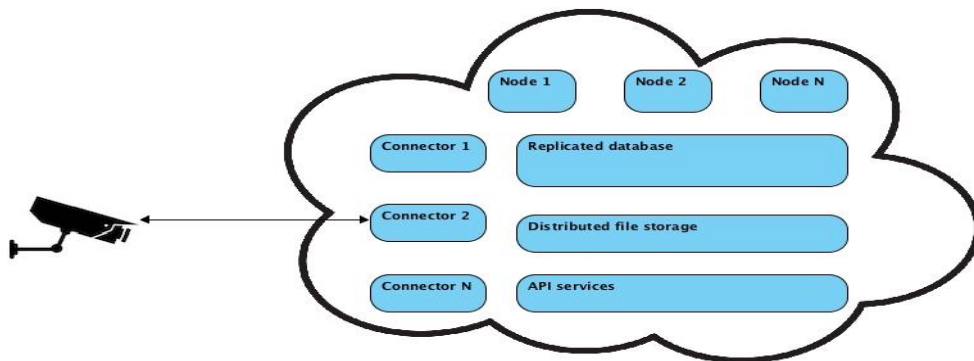


Fig. 1. Main cloud services

First, when camera starts up, it should request Connector the list of servers which will serve it. Until the camera connected to CSS has no configuration except network settings: IP address, mask, gateway and DNS addresses. Here we got the first ability to duplicate Connector service – use multiple DNS “A” records. The method described here is called Round Robin DNS [1]. In its simplest implementation, Round Robin DNS works by responding to DNS requests not only with a single potential IP address, but with one out of a list of potential IP addresses corresponding to several servers that host identical services. The order in which IP addresses from the list are returned is the basis for the term Round Robin. With each DNS response, the IP address sequence in the list is permuted. Usually, basic IP clients attempt connections with the first address returned from a DNS query, so that on different connection attempts, clients would receive service from different providers, thus distributing the overall load among servers. This feature allows spreading all requests from cameras within the system.

The mission of Connector is to transmit camera configuration. This configuration in a special way tells camera the list of servers which can serve it. This list depends on different parameters such as camera geolocation and predefined tasks.

After camera received a list of servers (with role “Node”) it will establish connection to them one by one in a random way. If connection with Node server is broken, camera will connect to another one from the list. This principle of operation is described in the article [2].

4. Database failover

Database is the mission critical part of any system. Best DBA practices tell us [3] to use: backups and replications to build failover system. The main idea of replication is to keep few database servers, which should synchronize all data and keep it up to date. Nevertheless, we should keep in mind to setup database backup system and backup validation system.

5. Distributed file storage

Failover file storage is the next mission critical component. There are few distributed file systems which could serve us best to author’s mind: Ceph, OrangeFS, Parallel Virtual File System (PVFS), RozoFS and GlusterFS. The main benefit of listed file systems is using TCP transport to connect servers.

Distributed File Storage (DFS) usually consist of several server's roles: Master, Backup, Chunks. Master and Backup servers contains database of stored files (meta data). Chunk servers contain files (raw data). Our DFS allows replicating and storing each file several times on different Chunk servers.

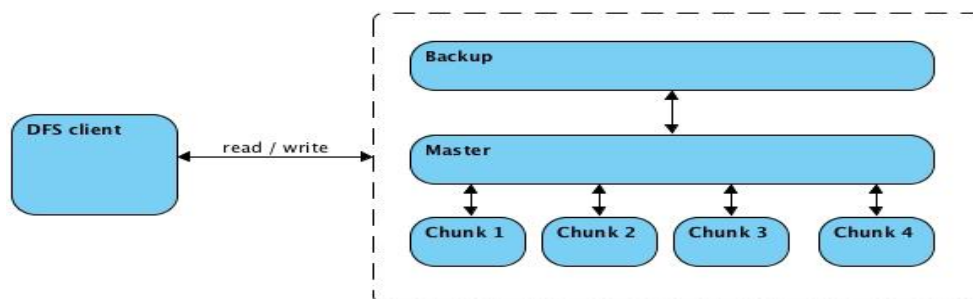


Fig. 2. Scheme of Distributed File Storage

Principles of storing data among DFS illustrates fig. 2. The main benefit of using Chunk servers is fault tolerance, which means that we do not lose any file if some Chunk server fails.

Depending on tasks it is recommended to store each file at least 2 times in different Chunk servers. In case of Chunk server fault (network fault, HDD fault), we can retrieve needed file from another Chunk server. All files, which were lost on faulty server, will be replicated from 2nd file copy to available Chunk servers.

Described system does not exclude of using RAID technology. To author’s mind it is more cheap to replicate file rather than redundant each Chunk storage with RAID.

The other advantage of our DFS is scalability. We can add or remove Chunk servers on the fly.

6. Video processing

Let’s go deeper and review more detailed system architecture. Fig. 3 shows us new system roles: message queue cluster, cleaner. Also, we can study detailed Node structure.

Message queue cluster is a failover messaging environment based on replicated database and replicated interchangeable servers. This service allows all services within the system keep in touch.

Back to the Node we should remind main Cloud Surveillance System tasks. The first one is recoding of huge amount of cameras. We should prepare our system to process as more as possible cameras to release system in projects like Smart City [2]. The second task is broadcasting. Broadcasting different types of public events becomes more popular nowadays. And the last but most important thing is video analyze.

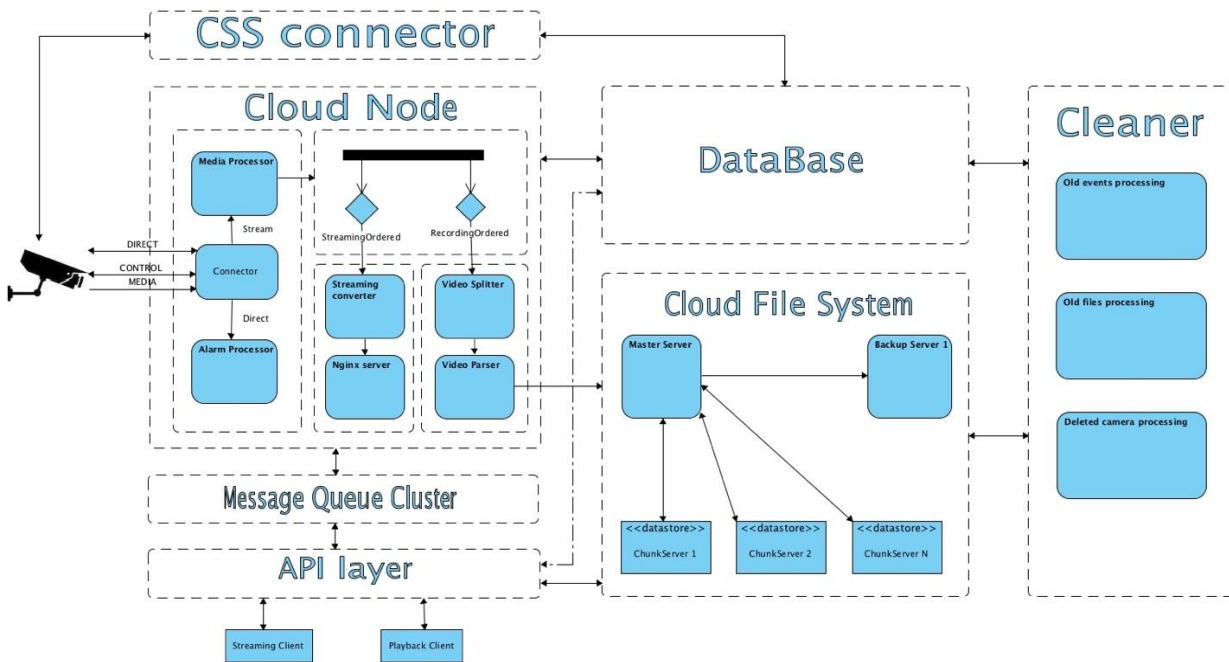


Fig. 3. Cloud Surveillance System modules

Let's explore the recording process. Every surveillance camera provides continuous endless video stream. That means that we need to split video data to fragments, because we cannot save endless file to DFS. The other side well described in official FFmpeg [4] knowledge base: "Normally, a MOV/MP4 file has all the metadata about all packets stored in one location (written at the end of the file, it can be moved to the start for better playback by adding faststart to the movflags). A fragmented file consists of a number of fragments, where packets and metadata about these packets are stored together. Writing a fragmented file has the advantage that the file is decodable even if the writing is interrupted (while a normal MOV/MP4 is undecodable if it is not properly finished), and it requires less memory when writing very long files (since writing normal MOV/MP4 files stores info about every single packet in memory until the file is closed). The downside is that it is less compatible with other applications."

Thus, Video Splitter processing live stream and split it to video segments. Video Parser waits for complete segments and when it's ready (fig. 4), start simple processing:

- 1) Coping segment to DFS;
- 2) Analyzing segment for any kind of alarm (movement, fire alarm, etc);
- 3) Storing file information in database.

Only these 3 steps could lead to a huge CPU load in real time system [5] and cause Node fail.

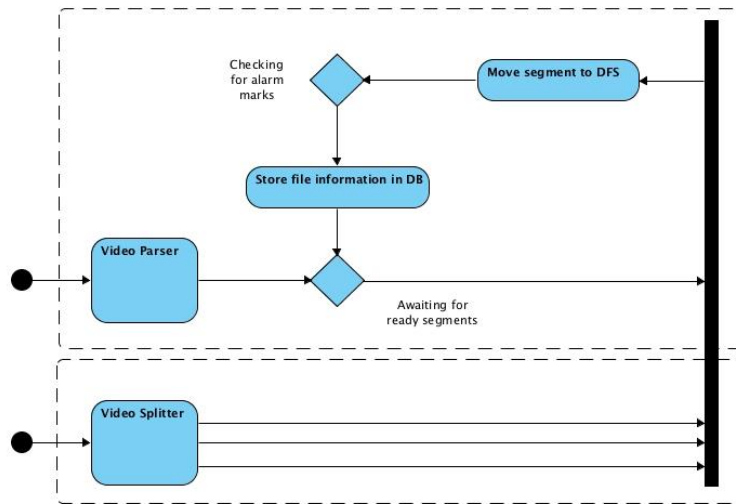


Fig. 4. Recording & video processing scheme

Cleaner – is the simplest, but heavy loaded module of cloud system. The main goal of the service is to remove old events and files from our system. But in case of Video Parser, which operates with video fragments from only 1 Node, Cleaner should process files and events from all over the system. This makes this service no less interesting.

7. Cloud approach to the surveillance system

Well, described surveillance system is distributed, replicated, failover, redundant. But where is the Cloud? To answer this question, we need to rollback to Cloud service models [6] and check them one by one (fig. 5).

Service Models	Cloud Stack	Stack Components	Who Is Responsible
SaaS PaaS IaaS	User	Login Registration Administration	Customer
	Application	Authentication Authorization User Interface Transactions Reports Dashboard	Customer
	Application Stack	OS Programming Language App Svr Middleware Database Monitoring	Vendor
	Infrastructure	Data Center Disk Storage Servers Firewall Network Load Balancer	Vendor

Fig. 5. Cloud service models

The basic service model is IaaS (Infrastructure as a Service), which means that we can request data center to provide us predefined server with network connection. This is the hardest way to build own cloud solution. After we got a bare metal we can choose a cloud platform from a variety of solutions. One of the popular are: Apache Cloud Stack, Xen Cloud Platform,

VmWare vCloud, Proxmox. After we choose cloud software platform we can install and manage everything by our self. This way provides best flexibility of configuration and options, but difficult to manage and maintain.

The second level is PaaS (Platform as a Service). In this case we receive pre-installed software for our solution (operating system, database server or even distributed failover file storage). This way more preferred, because it is much easier to maintain. At the moment PaaS service is promoted by huge corporation like Amazon (Amazon Web Services), Google (Google Cloud Services), Microsoft (Azure Cloud).

The third level is the level of Cloud Surveillance System customers. As a result of Cloud-based infrastructure and distributed, redundant surveillance software model customers receive a well stable, secure, reliable Cloud Surveillance System.

8. Conclusion

The article describes the main nuts and bolts of video processing under conditions of cloud service models and principals. The author found main properties of failover surveillance system and proposed methods, which allowed reaching failover and redundant system attributes. It was proposed the methods how to spread the load among computing resources. The software basic structure and key processing mechanisms were described.

REFERENCES

1. Aitchison R. Pro DNS and BIND / Aitchison R. – New York: Apress, 2005. – P. 165 – 167.
2. Дятлов Е.И. Использование облачных технологий видеонаблюдения в рамках программы интеллектуального города / Е.И. Дятлов // Inudoco'16. – Slavutych, 2016. – P. 222 – 227.
3. Connolly T.M. Database Systems: A Practical Approach to Design, Implementation, and Management / T.M. Connolly, C.E. Begg. – [6 ed.]. – Pearson Education Limited, 2015. – 785 p.
4. FFmpeg multimedia framework [Електронний ресурс]. – Режим доступу: <https://www.ffmpeg.org>.
5. Дятлов Е.И. Балансировка нагрузки в распределенных вычислительных системах / Е.И. Дятлов // Системи обробки інформації: зб. наук. праць. – Х.: Харківський університет Повітряних сил імені Івана Кожедуба, 2015. – Вип. 136. – С. 128 – 134.
6. Kavis M.J. Architecting the cloud. Design decisions for cloud computing service models (SaaS, PaaS and IaaS) / Kavis M.J. – Wiley, 2014. – 104 p.

Стаття надійшла до редакції 30.05.2016