A Survey On Distributed Video Management Cloud Platform Using Hadoop

Surbhi Saxena
Mtech Scholar (Computer Science and Engineering)
Oriental Institute Of Science and Technology
Bhopal, India

Pankaj Pandey
Assistant Professor (Computer Science and Engineering)
Oriental Institute Of Science and Technology
Bhopal, India

Abstract—This paper presents the literature review on distributed video management cloud platform using Hadoop. Due to complexities of big video data management, such as immense processing of large amount of video data to do a video summary, it is challenging to effectively and efficiently store and process these video data in a user friendly way. Based on the parallel processing and flexible storage capabilities of cloud computing, in this paper we are discussing practical massive video management platform using Hadoop [1], which can achieve a fast video processing (such as video summary, encoding, and decoding) using MapReduce, with good usability, performance, and availability. Red5 streaming media server is used to get video stream from Hadoop distributed file system, and Flex is used to play video in browsers. A user-friendly interface is designed for managing the whole platform in a browser server style using J2EE. In addition, we discuss some related work and compare it with our surveyed work. In addition to presenting these works, we also discuss possible extensions.

Keywords—Hadoop, J2EE, video processing, MapReduce, video management

I. Introduction

With the progress of Internet of things (IOT), streaming media applications such as security and traffic video surveillance are deployed rapidly. Streaming media data contains considerable useful information. These massive audio and video data are unstructured data that traditional approaches can't handle storage and processing efficiently. The majority of existing approaches use centralized storage, such as disk array, which is not scalable for efficient massive video data processing.

In order to alleviate these problems, researchers are beginning to make use of the relatively new cloud computing technologies to explore video storage and processing. For example, work on using Hadoop [2], [3] to accelerate video processing time utilizing MapReduce programming style [4], [5], and similar work called Split & Merge architecture was proposed in [6] for high performance video compression. However, these existing work lack details on how to design a versatile video management platform in a user friendly way, and how to effectively use Hadoop to fine tune performance of video processing in order to build a practical and usable video cloud platform.

In this paper, we survey a previous work which gives a solution that seamlessly integrates J2EE, and Flex with Hadoop to provide a user friendly way for video management, and also that integrate an open source media server to make our solution economical. H.264 video files stored on HDFS [7], [8] are transcoded to common video format using FFmpeg. Users can download the transcoded files and play it with a video player directly. Furthermore, Gaussian mixture model (GMM) is used to implement video background subtraction. A web based interface is designed using J2EE to provide users an easy way to manage the platform, where Struts2, Spring and Hibernate are integrated.

II. BODY

A. ARCHITECTURE AND DESIGN OF THE VIDEO MANAGEMENT PLATFORM

![Deployment diagram of the platform]

As discussed in the introduction, the video management platform should have a browser server style which can upgrade usability for normal users. Also it will make use of...
Hadoop platform to make use of the parallel and distributed computing and storage capabilities. The architecture of the video platform described with a UML deployment diagram is shown in Figure 1.

The distributed cloud platform mainly consists of four parts: Hadoop clusters, Red5 streaming server clusters, Web servers and database servers. Video stream from video capture devices (e.g. cameras) is stored on HDFS via Web server [9]. Large amount of persistent video data (e.g. in smart surveillance applications) need a huge storage space, but it's not necessary to store all the video from cameras, so some processing is done on the DataNode. For example, video summary can extract specified video that we care about and remove useless video that contain little information.

The background subtraction can separate background and foreground effectively, which makes it easier to implement video summary. Meanwhile, video encoding and decoding is an important method to ensure efficient video transmission. Red5 streaming media server uses RTMP, RTMPT, and RTMPE to play audio and video when a users' request is sent to a Web server. There are two important components with a Web server. The first one is FFmpeg, which is a transcoding tool to transcoded RTSP stream from cameras to RTMP stream. Another one is Flex, which can get real time stream from a Red5 server. All user information is stored on MySQI database servers. Finally, a user-friendly graphical interface is designed for an easier operation in a browser.

B. IMPLEMENTING THE VIDEO CLOUD PLATFORM

There are three important modules in Hadoop: HDFS, YARN and MapReduce. HDFS is the Hadoop distributed file system which can provide high reliability, high scalability and high throughput data services. YARN is responsible for unify management and scheduling of cluster resources. And MapReduce is a distributed computing framework that provides efficient, fault-tolerant, and scalable parallel processing capabilities.

i. DISTRIBUTED STORAGE

HDFS uses a master-slave configuration which includes a NameNode and a number of DataNodes. The NameNode (master) is a central server, and it manages the file system name space and clients' access to files. The DataNode (slave) is the storage space that stores various data. Additionally, Zookeeper cluster is used to manage the active NameNode and the standby NameNode to ensure reliable hot-swap between two NameNode when the active NameNode is not working. HDFS has a strong system scalability which supports adding or removing nodes dynamically. Whenever a larger storage space is needed, new nodes can be added online like nothing happens and data can be redistributed to old and new nodes automatically, i.e., the storage space of Hadoop cluster can be infinite large theoretically. Meanwhile, HDFS stores files on blocks with limited size, which can be adjusted according to your own configuration. So the storage strategy of HDFS is to cut those large files to little pieces and scatter them over the storage space in the cluster. HDFS stores files on partition with automatic block replication to implement high fault tolerance and its backup number can also be configured.

To access video files stored in HDFS, users should login to the platform through a client browser. Then they can manage videos, like upload, download, and delete video files if they have such rights. It's hard for users to operate HDFS directly. A user-friendly interface is designed using the browser server architecture style, and J2EE technology is used to implement it to provide a simple and easy operation environment. When a user requests are sent to a Web server via http protocol, this Web server will call APIs to send reading and writing requests to the HDFS cluster. Then Web server receives the response from HDFS and display it in the browser. As shown in Figure 2, the Web server acts as a relay in this process.

![Fig 2. Work flow of user requests.](image)

When uploading video files to HDFS cluster, a logged in user will upload video files to the Web server, and then a UploadMovieToHDFS() method is used to upload files to the HDFS cluster. After this, the Web server deletes the cached files. When downloading video files from a HDFS cluster, the Web server queries files address information and apply DownloadFromHDFS() method to get files from the HDFS cluster. When deleting video files, the request is sent to the Web server, then the persistence layer query the file address information and apply DelMovieFromHDFS() method to delete the file from the HDFS cluster.

ii. VIDEO PLAYING

Red5 is an open source streaming media server [9] based on Flash streaming media service [10] developed in Java. It uses RTMP, RTMPT and RTMPE to play audio (MP3) and video (FLV, MP4, etc.) [11], [12]. When the request of playing video is sent by users, the Web server will transcoded RTSP stream from cameras to RTMP stream using FFmpeg and sends it to Red5 streaming media server. Then the Web server can get real time video stream from Red5 streaming media server using Flex. The process is shown in Figure 3.

![Fig 3. Work flow of video transcoding](image)

The transcoded video stream from camera is stored on HDFS. A mount is made using fuse-dfs between Red5 streaming media server and HDFS cluster and a
directory named Streams for the mount directory is created. The process is shown in Figure 4.

![Work flow of video playing](image)

Fig 6. Work flow of video playing.

The Red5 streaming media server scans FLV files from the Streams directory and transcodes them to video stream. The Web server receives video stream from the Red5 streaming media server and plays it in a browser.

### iii. BACKGROUND SUBTRACTION

One of the most important video processing tasks is background subtraction. There are lots of algorithms to achieve this. The mixture of Gaussians (MOG) is used widely. In surveyed work an improved Gaussian model designed by Zivkovic is used [14]. The advantages of this algorithm are that the number of Gaussian components is constantly adapted per pixel and that it can perform shadow detection.

### III. RELATED WORK

Myoungjin Kim et al. proposed a Hadoop-based distributed video transcoding system in a cloud computing environment, which can transcode various video format to MPEG-4 [15]. They design and implement the platform efficiently using MapReduce framework. Comparing with their work, besides transcoding, in survey work a user-friendly interface to operate HDFS easily, details on how to use Hadoop efficiently by tuning different parameters are also presented.

In [16], the authors proposed a Hadoop-based storage architecture for massive MP3 files. They used classification algorithm in pre-processing module to merge small files into sequence files. They confirm that the introduced efficient indexing mechanism is a good solution to the problem of small files. Survey work focus on Hadoop-based storage and processing architecture for big video data, and usability is one of their focuses.

Lin et al. [17] proposed a prototype of cloud based video recording system. The system can provide scalable video recording, data backup and feature detection. They also used HDFS to store video data, which is similar with ours. In survey work, they apply J2EE technology to develop a user-friendly interface for HDFS, and they used Red5 to play video. Additionally they consider fine tuning Hadoop is an important task for practical big data applications.

In [18], the authors studied and analyzed the open source Red5 streaming media server and they used Flex and Red5 streaming media server for playing video successfully. In surveyed research, they mainly concentrate on Hadoop based management of video data using Flex and Red5 streaming media server. In their work they combine Flex, Red5 and Hadoop together to implement their platform.

Liu et al. [19] present a framework for video playing and video storage based on Hadoop. Their framework provides high availability services, which could support concurrent access and playing streaming media in mobile terminals. Survey work is similar with their work, but they target a user friendly big video data management platform and they also present their experiences on using J2EE for the management of video data, and experiences on optimizing the video management platform with Hadoop.

In [20], the authors presented a video monitoring system that can meet the users’ demands of searching video, uploading video, downloading video and transcoding video. They also used FFMpeg to transcode video. In survey work, they focused on the usability of the video management platform, plus the experiences on integration of J2EE, Flex, and Hadoop.

In survey previous work [21], they did some preliminary research on video monitoring platform, where Hadoop based processing was explored. In this paper, they focus on a user-friendly video management platform with its architecture, and sharing our experiences on fine tuning the performance of Hadoop based video processing.

### IV. CONCLUSIONS AND FUTURE WORK

In this paper, we presents the literature review on distributed cloud platform for storing and managing massive video data based on the integration of Hadoop, Red5 streaming media server and J2EE. In which the platform is economical because the cluster can be built by personal computer, and the software packages are open source. They have proposed to use Red5 streaming media technology to realize the video playback while downloading. They have also evaluated the performance of the platform extensively and it shows that the platform has good performance. They have also found that the parameter setting of Hadoop can greatly affect the performance of the video processing.

They are experimenting to use the presented approach for smart transportation system in Qingdao City, China. Additionally they are adopting an online processing and offline processing combined approach [5] in order to make full of the cloud computing potentials. Another direction is that they plan to use deep learning to do knowledge mining from video data [22].

### REFERENCES


