A framework for video transcoding and management in cloud manufacturing

Jian Zhou* and Ying Zuo

School of Automation Science and Electrical Engineering,
Beihang University,
Beijing, 100191, China
Email: zjkermit@gmail.com
Email: zuoying226@126.com
*Corresponding author

Abstract: Cloud-based manufacturing has been defined as CMfg. It is a system built on cloud computing technology that can integrate a large number of manufacturing resources. In a CMfg system, there are huge amounts of data and one of them is the video used to monitor manufacturing processes and learn knowledge. In cloud platform, the video data generate fast and have diverse formats, so the need for unified transcoding and management is necessary. In this article, an elastic cloud framework for fast video transcoding process and management is proposed. The core of this framework is a distributed system based on Hadoop that is designed for massive data storage and parallel computing.

Keywords: cloud; Hadoop; video; distributed system.


Biographical notes: Jian Zhou received his BS degree in 2011 from the Department of Automation Science and Electrical Engineering at Beihang University, China. He is currently a Master student at School of Automation Science and Electrical Engineering in Beihang University, Beijing, China. His main research interests include service-oriented manufacturing and cloud manufacturing. He has been involved in the project of cloud manufacturing, and his main emphasis is on the cloud manufacturing system architecture and video processing.

Ying Zuo received his MS degree in School of Mechanical and Electronic Engineering in 2012, and is studying for a PhD degree in the School of Automation Science and Electrical Engineering at Beihang University. Her main research interests are in the areas of service-oriented manufacturing, energy saving and emission reducing. Her main emphasis is on the evaluation, analysis/simulation, and optimisation of energy saving and emission reducing of products life-cycle.
1 Introduction

CMfg (Li et al., 2010a, 2011; Zhang et al., 2014) is a new concept based on ‘manufacturing as a service’, drawing on the idea of cloud computing. CMfg is an advanced information technology, manufacturing technology and emerging networking technology. It can implement cross-product integration by creating services to embody the concept (Li et al., 2010b; Tao et al., 2011a).

The objective of CMfg is to realise the resource sharing and collaboration, in which the resource refers to the sum of resource elements involved in the entire life cycle of products. These resources include software resources, network facilities resources, equipment resources, information resources, knowledge resources and so on. So the data involved in these resources are varied. Currently, some scholars are focus on the traditional sense of resource and the dataset is mainly the text data. However, video as a multimedia resource under the network environment, its status is becoming more and more important. Especially in the cloud environment, the monitoring data of manufacturing process, the knowledge acquisition and the communication between remote users are all supported by video services.

In CMfg platform, there are large amounts of structured and unstructured data (Bao et al., 2012), such as device parameters, text data, log data, videos, etc. Various types of data are perceived access to manufacturing platform in the cloud through networking and virtualisation technology (Tao et al., 2011b). Absolutely, the data generated by sensors occupy the majority. However, from the perspective of the file size, the video resource occupies a large proportion in all of cloud resources. In CMfg, in order to realise the monitoring and perception of the cloud resources in real-time, and to realise that the user remote dispatch and interactive use of cloud resources, many techniques such as internet of things and CPS are adopted to realise the intelligent access function in manufacture and processing. Parameter acquisition of manufacturing systems usually uses sensor. However, as for cloud users, these parameters are not intuitive. So it is needed to adopt the necessary video techniques to assist manufacture or production. In these procedures, accordingly a large amount of video techniques adopted, a large amount of videos is produced. Compared with ordinary text resource, these video resources have the expression of larger volume, more information and easier to damage. Therefore, efficient storage modes should be taken into consideration. In addition, these video resources provided to cloud users are through cloud platform. In order to achieve the video services in the cross-platform and heterogeneous network, further requirements are proposed for video formats, distinguishability and code rates.

Therefore, distributed storage mechanism should be considered. The distributed storage mechanism can ensure that these video files can be effectively stored. Besides, the video needs to be transcoded to ensure that the video can be played in the cloud platform. Just like in CMfg platform, user from different terminals and networks through the platform accesses the same video resources. The cloud platform can be able to provide the same video services for user from different terminals and networks.

Compared with text data processing, video processing is more difficult, because existing video has lots of formats and video contains more information than text, so it has no convenient for storage and use. Furthermore, with the continuous evolution of computer systems, video volume grow faster and faster. Recent video files also have changed from a low capacity and low resolution to high capacity and high resolution (Xin et al., 2005). This increases the difficulty of video to store. In addition, these large
amounts of video files for the information processing require a lot of computing resources. So it also increases the difficulty of video to transcode. How to transcode and store video in a shorter period of time and provide video to user is the difficult point of this study. Further, the cloud platform should also have the ability to transcode the video to meet the needs of multi-terminal and heterogeneous network environments, as the network environment, the terminal type, the type and format of video content are usually different.

In order to develop a video transcoding and management system for CMfg platform, the volume of the video, computer storage and processing efficiency should be taken into account. Therefore, video transcoding techniques are required for delivering large amounts of video data in multiple formats to multi-terminal and heterogeneous network environments. In addition, with the emergence of new technology, Hadoop has been widely used in all walks of life, which puts forward new ideas to solve the above problem.

In this article, a framework for fast video transcoding and well management in CMfg environment is proposed. By using the Hadoop framework (White, 2009), Hadoop distributed file system (HDFS) (Shvachko et al., 2010), and MapReduce (Dean and Ghemawat, 2008), video processing and video storage can be combined, video transcoding speed has been improved and the safety of its storage has been ensured.

This paper is organised as follows. Section 2 presents the related works about video research in cloud manufacturing. Section 3 presents the framework. Section 4 describes the key technology for implementing video process and management. Section 5 talks about prototype system and case study. It is concluded in Section 6.

2 Related works

2.1 Overview of cloud manufacturing

How to integrate social resources, improve resource utilisation, reduce energy consumption and emissions so as to realise the service-oriented manufacturing in the manufacturing process is the background of cloud manufacturing. The architecture includes: physical resource layer, resource layer, service layer, the core service layer, application interface layer and user layer. In the physical resource layer, all the physical resources are connected to the network, which realise the comprehensive interconnection between physical resources. These resources include two-dimensional code labels, RFID tags and readers, camera, GPS, sensors, etc. For most of the data collected in the network, cloud manufacturing in the resource layer package accessed various resources into virtual manufacturing resources. The cloud platform uses cloud resource pool to store resource (Zhang et al., 2014).

2.2 Video in cloud manufacturing and its management requirements

As a kind of social production, each stage of cloud manufacturing is related to many complex processes. And the video as a multimedia resource under the network environment is occupying an important position. Not just because the video in the network environment can transmit more abundant information than the text, and the more
important thing is that in the cloud manufacturing process, the monitoring data of manufacturing process greatly dependent on video. However, video resources accessed in the form of a video stream cannot be directly stored in the cloud resource pool. In addition, users can only download video data from the cloud platform. Therefore, as a kind of resource in cloud manufacturing, further research need to be done to find a better way to provide video resources to user.

In cloud manufacturing, video data are regarded as regular file directly stored in the cloud storage resource pool. The rate of video utilisation is very low. In order to enable user to use these video resources without download, video online play service is necessary. So it is essential to transcode the original video format to a new format that is suitable for playing online. At the same time, the ability to provide different rates and resolutions are required to adapt to the heterogeneous network environment and terminal. In the cloud environment, to provide video transcoding and storage service, the premise is to manage video effectively.

First of all, the cloud platform should have the ability to store the large amount and the rapid growth of video resources. The cloud platform also should be able to provide user with online playing and download. Finally, the cloud platform operators should have the ability to review and delete video resources.

In addition, owing to the heterogeneity of video resources, video format, resolution and bit rate are varied. The cloud platform needs a unified transcoding of video resources. How to transcode video fast in the cloud environment and ensure the safety of storage is the focus of this study.

2.3 Video transcoding frameworks

Transcoding is a compute-intensive process and some methods to improve its efficiency have been proposed (Morrison, 1997; Youn et al., 1999). Most methods focus on solving the issue that how to utilise the video information to speed up the video transcoding process. Other researchers put the emphasis on utilising multiprocessor to parallel transcoding (Shen et al., 1995; Sambe et al., 2005; Deneke, 2011) or cloud transcoding (Pereira et al., 2010; Kim et al., 2012, 2013; Garcia et al., 2010). In fact, the transcoding technique is comparative maturity, but the issue of how to combine the existing video processing technology and distributed architecture is not researched so much. Furthermore, video transcoding framework in CMfg still shoulders the video storage responsibility. In this research, a framework on distributed video transcoding and management is proposed, and the framework would be used for video transcoding, storage and service in CMfg. To achieve the purpose, the framework combines MapReduce programming paradigms and open source video processing technology FFmpeg together.

3 Video transcoding and management framework

As mentioned in the previous sections, existing video processing technology and distributed processing framework can provide a new thought for distributed video transcoding. The purpose of video transcoding and management is to transcode various heterogeneous videos into the videos fit for playing in network, which contributes to users from different terminals and networks accessing quickly.
In CMfg, video types mainly include real-time monitoring video in the manufacturing process and sample video offered by users. Real-time monitoring video is used to assist the manufacturing, which enables users to monitor the process. Sample video is used for resource consumers to offer instructions.

The two types of video are finally stored in the cloud platform and it can provide users online video on demand services, so it is necessary to transcode these videos. In view of combining video storage and video transcoding contributing to improving the video transcoding process and service, the follow framework is adopted to manage the video.

**Figure 1** Framework of video transcoding and management

The proposed architecture of video transcoding and management is shown in Figure 1, which primarily consists of four layers. Communication between the layers is via web service. In this study, the system build web service in the representational state transfer (REST) (Fielding, 2000) style. By using the REST style, the system can work in a fully distributed environment. The four-layered structure is as follows:

**Application Layer**
- Consumer
- Operator
- Provider

**Core Layer**
- Basic Service
  - User management
  - Authority management
  - System management
- Video Management Service
  - Video Upload
  - Video Review
  - Video Download
  - Video Delete
- Video Process Service
  - Resolution Adjustment
  - Frame Rate Adjustment
  - Format Transformation
  - Data Rate Adjustment

**Middle Layer**
- Retrieve Video Resolution
- Retrieve Video Format
- Video Segmentation
- Retrieve Video Frame Rate
- Retrieve Video Data Rate
- Parallel Transcoding
- Read-Write Database
- Submit MapReduce Job
- Video Stitching

**Data Layer**
- Relational Database Management
  - MySQL-Proxy
  - Connection Pool
  - Master-Slave
  - Transaction
- Hadoop Distributed File System
  - Zookeeper
  - DataNode
  - NameNode
  - Replication
Data layer (D-layer): This layer provides a data storage service. All the related data in the system go through this layer. It also provides data support for the entire video transcoding and management system. In this study, it mainly includes the original video data, the data in the entire life cycle of video transcoding and the data for system operation support. It can be divided into two sub-layers as follows:

a. Relational database management layer (RDBM-layer), it is mainly used to support the core layer and provide web services to the middle layer. The actual data are forwarded through the middle layer.

b. HDFS-layer, it is the core of video transcoding and storage, all operations on video are handled in this layer. This layer is in charge of processing the original video files and the storage of video files. It also uses web services to provide consistent services.

Middle-layer (M-layer): The M-layer, as a logical layer between the C-layer and the D-layer, is responsible for the direct operation of D-layer. It transmits a REST request for video transcoding to D-layer and then forwards the recall data from D-layer to the C-layer.

Core-layer (C-layer): It provides video services for the application layer. It primarily includes the basic service (such as user management, authority management, system management, etc.), the video management service (e.g., video upload, video download, video review, etc.), and the video process service (e.g., video resolution adjustment, video frame rate adjustment, video format transform, video data rate adjustment, etc.).

Application-layer (A-layer): Application layer is mainly responsible for providing video data for different users such as consumers, operators and providers. It can provide a uniform service interface to other systems.

4 Key technologies for implementing video transcoding and management

In the previous section, the video transcoding and management framework are introduced briefly. This section will focus on the referred technologies and details.

4.1 D-layer: data management for the proposed video transcoding system

The D-layer provides the entire video processing with relevant data support. From the aspect of storage type, the D-layer is divided into relational data storage and video data storage. As for different storage types, different storage methods are adopted. Shown in Figure 2, it is the framework of data storage, which is mainly comprised of two parts. One is a MySQL (Widenius and Axmark, 2002) cluster used for relational data storage. The other is a Hadoop cluster for video data storage and transcoding.
4.1.1 Storage part

The two clusters are used in this study to enhance the performance and the high availability of the storage. For the Hadoop cluster, it mainly consists of three types of node. They are NameNode, DataNode (Shvachko et al., 2010) and Zookeeper (Hunt et al., 2010) node. Hadoop uses a block division mechanism in which it will store data in fixed-size blocks of data. These data blocks distribute in different DataNodes in HDFS. Setting the replication property allows data disaster recovery. DataNodes are used as common storage for blocks. DataNodes send periodic heartbeats and block reports and handles commands from the NameNode. To maintain configuration information and distributed synchronisation, Zookeeper is also adopted in the system.

For the MySQL cluster, according to CAP theorem (Gilbert and Lynch, 2002), it is impossible for a distributed computer system to simultaneously provide all three of the following guarantees: consistency (all nodes see the same data at the same time), availability (a guarantee that every request receives a response about whether it was successful or failed) and partition tolerance (the system continues to operate despite arbitrary message loss or failure of part of the system).

In a distributed system, partition tolerance is essential. So in order to guarantee the high availability, the proxy mechanism is adopted. Considering the read operation is the majority, the master is used to write data and the two slaves is used to read data. Consistency is realised through data synchronisation between the master and the slaves.
4.1.2 Transcoding part

The main idea for video transcoding process is to use distributed computers doing the similar transcoding task in parallel and each computer transcode a different part of the video. Compared with stand-alone transcoding process, it will accelerate the transcoding time, reduce CPU and memory usage in a single computer. In this study, FFmpeg (Tomar, 2006), the open source video processing software, is adopted. Through the deployment of FFmpeg in each node of Hadoop clusters, the MapReduce process can conveniently carry out video transcoding.

So the first step is to segment a large volume video file into several fixed-size video segments. However, unlike plain text file commonly used in Hadoop, the constitution of video files are more complicated. Video includes a rapid sequence of coded frames. Each frame is one picture. Moving picture experts group (MPEG) divides coded frame into three classes, which is shown as follows:

- **I-frame (intra coded frame)** is an independent frame. The frame is coded independently of all other frames that mean it does not connect any other frame in the front and rear. They appear at periodic intervals and cannot be configured by other frames.

- **P-frame (predictive coded frame)** is associated with previous I-frame or P-frame. Each P-frame comes from the front of the frame, but changes between the previous frame and the current frame cannot cover a large part. P-frame can be generated by the previous I-frames and P-frames.

- **B-frame (bi-directionally predicted frame)** is associated with the previous and rear I-frame or P-frame, but there is no association between two B-frames.

Considering the above-mentioned three categories of frames, in order to avoid video distortion, choosing the appropriate position of the frame to cut the video is very important. Because P-frames and B-frames cannot be decoded without I-frames, the cutting point must base on the I-frame. The frames between two consecutive I-frames are referred to as a GOP (Haskell, 1997). That is to say, video segment contains an integer number of GOP units. Otherwise the video cannot be decoded in the cutting position due to missing I-frames.

Another aspect to be considered is the storage mechanism of HDFS. In HDFS storage system, it introduces a concept of the file system block and the input split, the block is the smallest unit of storage. HDFS define the size of 64 MB. Usually, HDFS will slice the input data into many input splits and each input split will produce a map process. To store file, HDFS data are stored in blocks and each block will be replicated several copies to other nodes. In the file hdfs-site.xml, the value of replications can be set.

Assuming the size of the cut video segment is $Y$ and the size of HDFS block is $X$. Then each video segment can be stored in $Y/X$ nodes. When the video segment is transcoded, HDFS must transport data to one node through the network. Clearly, network data exchange will result in efficiency decreased. From this perspective, to decrease the load of the network in transcoding process, making $Y$ less than $X$ is better. In addition, if the size of the video segment is much smaller than the size of HDFS block. For the same video file, the numbers of blocks become more. This will lead to the addressing and scheduling more complex. So in the video process, video is cut into equal-sized pieces and the size is close or equal to the size of HDFS block.
Along the way, one or more of video transcoding tasks are splitting into two parts, exactly two MapReduce processes. In the first MapReduce task, the original video is divided into many segments and each segment is equal to HDFS block. The map process is responsible for the actual video file cutting and the reduce process is responsible for the persistence of information. Specifically, the map process will cut video files and generate universally unique identifier (UUID) for the original video file. Then UUID as the key and the sequence number of the video segment as the value are transferred to reduce process. The reduce process will save the information into a database.

In the second MapReduce task, the segments of video will be transcoded and merged. The map process is responsible for transcoding video segment, since each video segment is stored in a block, all the process does not produce data exchange through the network. In the cluster environment, the process will be fully executed in parallel, so it will be very fast. The reduce process will receive information from the map process. Then, all the video segments will be merged into a new video. The new merged video is still stored in HDFS.

As showed in Figure 3, one DataNode of the Hadoop cluster is used to preprocess the video. In this part, the system first collects video metadata that include video format, the length of video, the size of file, video definition, frame rate, etc. Then, the video file is cut into many segments and each segment contains a part of source video. For every segment, a map process is created for video transcoding in a DataNode. Finally, all transcoded segments are merged in one DataNode.

Figure 3 Video segment, transcode and merge
4.2 M-layer: business logic processing and request dispatcher

The main purpose of this layer is to separate data, view and procedure. This layer encapsulates all programme operations on the data layer, which make the system easier to expand. Besides, to working in a distributed environment, all interfaces are provided by REST application programming interface (API). And all the return data are formatted into JavaScript object notation (JSON) (Crockford, 2006) data, which makes the layer completely independent of all layers.

As shown in Figure 4, the M-layer accepted REST uniform resource identifier (URI) request, then submit video processing tasks to Hadoop cluster and transmit data request to MySQL cluster. In Hadoop cluster, these original videos are eventually converted to H.264/MPEG-4 AVC format (Schwarz et al., 2007). This layer has four kinds of responsibilities as following:

- **Routing forwarding as dispatcher:** The dispatcher captures all requests. From the beginning, the dispatcher gets the REST URI, and then parses the URI. Finally, forwards the request to the appropriate REST action.

- **Operating Hadoop cluster as client:** All operations on Hadoop cluster are implemented by Hadoop REST action, including submitting MapReduce tasks, uploading file to HDFS, deleting file in HDFS, downloading file from HDFS, appending file to HDFS, etc. The MapReduce process is illustrated in Figure 4. The video need to be processed is segmented into multiple parts and each part is corresponding to a map process. Every map process transcodes one part of the video and adjust the video parameters, and then all parts merged into a reduce process.
A framework for video transcoding and management in cloud manufacturing

- **Operating MySQL cluster as client**: All operations on MySQL cluster are carried out by MySQL REST action, including inserting operation, deleting operation, updating operation, and selecting operation.

- **JSON data response as web service**: By using aspect-oriented programming (AOP) and JSON-lib library, all of the data requested through the web service prior to transmission are encapsulated into JSON format.

### 4.3 C-layer: basic services provider and video services provider

The C-layer provides all external data call interfaces and all interfaces are also in REST style. As shown in Figure 5, in accordance with the REST style convention, request method in hyper text transport protocol (HTTP) has semantic meanings. For example, POST method means add, PUT method means modify, GET method means retrieve, and DELETE method means delete. In this layer, a dispatcher is also configured. All the REST requests are preprocessed in the dispatcher firstly. Then the request method in the HTTP header is extracted. According to the REST URI and HTTP request method, the dispatcher forwards the actual request to the appropriate action. All functions are as follows:

- **Add resources**: Using POST method, it includes uploading video, importing video, adding video description information, creating video category, creating user, creating user authority, creating video stream interface, adding video stream address (using RTMP protocol), and submitting video processing job to Hadoop cluster, etc.

- **Delete resources**: Using DELETE method, it includes deleting video, deleting video description information, deleting video category, deleting user, deleting user authority, deleting video stream interface, deleting video stream address, etc.

- **Modify resources**: Using PUT method, it includes modifying video description information, modifying video category, modifying user information, modifying user authority, modifying video stream interface, modifying video stream address, reviewing video status.

- **Retrieve resources**: Using GET method, it includes searching video, getting video resolution, getting video format, getting video frame rate, getting video data rate, exporting video, downloading video, getting video stream address, getting video stream interface, etc.

### 4.4 A-Layer

As shown in Figure 6, the related user can be divided into the following sub-layers.

#### a Provider

Providers can create video stream address. They can bind monitor equipment and cameras with the stream address easily. Thus, other user can access the video stream for viewing through the platform. Providers also are able to upload videos or import video files to the platform. These video files can be used as a learning resource or a guide resource provided to other user. Then user can watch the video via a PC or mobile terminals online.
5 Prototype system and case study

In this section, the whole process including the video uploading, transcoding and publishing is imitated. This prototype system, as one part of CMfg, is responsible for dealing with videos. In addition, this section verifies the advantages of cluster transcoding compared with stand-alone transcoding, and shows the effect of different segment size for video transcoding speed in HDFS.
A framework for video transcoding and management in cloud manufacturing

To transcode and manage videos, a video transcoding and management system is developed. In this platform, uploaded video can be transcoded, stored and managed. In the test environment, four virtual machines as one test cluster, each machine is equipped with one CPU of 2.4 GHz and 2G memory. Through adjusting the default file block size of Hadoop, the time consumption of transcoding video is tested. The size of the testing video is 332 MB and its format is AVI.

As shown in Figure 7, firstly, user uploads video to the cloud platform, then the cloud platform transcodes video, and finally, the operator reviews the video. The detail steps of this process are illustrated as follows:

1. The user first creates a classification and then fills in video classification information and parameters. If the classification already exists, it can be directly used, as shown in Figure 7(a).

2. According to the classification, the user can choose to upload video or import video, as shown in Figure 7(b) and Figure 7(c).

3. When video is uploaded successfully, the state of the video is unaudited. The system will trigger transcoding tasks at a periodic time. At that time, Hadoop cluster will transcode the video and write the successful transcoding information to the MySQL cluster, as shown in Figure 7(d) and Figure 7(e).

4. The operator search and check videos submitted by all users, as shown in Figure 7(f).

5. If the state of successful transcoded video is still unaudited, the operators can review it, only the users who passed the review can use the video, as shown in Figure 7(g).

6. As for illegal video, the operator can delete it, as shown in Figure 7(h).
Figure 7  Video transcoding and review process (see online version for colours)

Figure 8  Total transcoding time cost in different segment size (see online version for colours)
The figure shows the time consumption of distributed transcoding. When the number of DataNode is 3, the time consumption of transcoding represents a trend of decreased first and then increased along with the increase of the size of the block. The reason is that the resources of the cluster could be maximised used.

As can be seen from the figure, because of the testing video size less than 512 MB, which means the speed of using single transcoding is 847 s, but adopting the distributed method written in this paper, four nodes cluster Hadoop can reduce about 45% of time, greatly improving the transcoding efficiency.

6 Conclusions and future works

How to transcode and manage video is a challenging problem in CMfg. This study proposed a framework of elastic cloud system for fast video transcoding and management. In order to implement the proposed framework, a four-layered structure (i.e., data layer, middle layer, core layer, and application layer) system based on web service is proposed, a prototype application system is developed, and a case study is conducted to validate the feasibility of proposed framework.

The two contributions of the proposed framework are as follows.

1. According to the characteristic of the video, video resources in CMfg are effectively utilised. At the same time, using Hadoop framework ensures the safety of video storage, and using MapReduce technology increases the speed of video processing.

2. Adopting the REST style of web service, the system could use multiple servers to provide distributed service and could be integrated in any systems.

The proposed framework can be further improved and used in many applications, such as mobile apps, video management and processing systems, social network site (SNS) systems, cloud manufacturing system and so on.

However, research on video transcoding and management system is just beginning, and the relevant theories, technologies, and applications are immature so further research is required. In future, the following work will be emphasised:

1. Verify effectiveness of the framework in more nodes of the cluster environment.

2. Improve the ability of the system to handle more video formats, frame rates and data rates.

3. Optimise MapReduce algorithm.

4. Increase the video processing speed.
References


