

## Research Article

# Design and Experiment Analysis of a Hadoop-Based Video Transcoding System for Next-Generation Wireless Sensor Networks

**Haoyu Xu,<sup>1</sup> Liangyou Wang,<sup>1,2</sup> and Huang Xie<sup>3</sup>**

<sup>1</sup> Shanghai Advanced Research Institute, Chinese Academy of Sciences, Shanghai 201210, China

<sup>2</sup> School of Electronics and Information, Tongji University, Shanghai 201804, China

<sup>3</sup> School of Software Engineering, University of Science and Technology of China, Suzhou, Jiangsu 215123, China

Correspondence should be addressed to Haoyu Xu; [xuhaoyu@gmail.com](mailto:xuhaoyu@gmail.com)

Received 31 October 2013; Accepted 23 December 2013; Published 24 March 2014

Academic Editor: Zuqing Zhu

Copyright © 2014 Haoyu Xu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The next-generation wireless sensor network (WSN) has the capability of carrying hundreds of high-definition video streams, beside the feature of massive employment of energy-efficient nodes. However, several challenges are identified with respect to the video bearing, such as the different video formats, enormous size of “raw” video, and compatibility with heterogeneous terminal devices. The video transcoding system (VTS) is widely believed to address these challenges. This paper introduces a cloud-based, more specifically, Hadoop-based, video transcoding system to fulfill the vision of bearing hundreds of HD video streams in the next generation WSN, with a discussion on optimization of several significant parameters. This paper obtains three remarkable results: (1) there is an optimal value of the number of Mappers; (2) the optimal value is closely related to the file size; (3) the transcoding time depends principally on the duration of video files rather than their sizes.

## 1. Introduction

The next-generation wireless sensor network (WSN), in the broad sense, has two significant features. One is the massive employment of energy-efficient nodes that extends the networks’ working time from months to years. The other one is the dramatically increased throughput of the networks, thanks to the emergence of new wireless transmission technology operating on the microwave spectrum. IEEE 802.11ac is a case in point [1].

The increased link throughput, up to 1 Gbps, makes the next-generation WSN capable of carrying hundreds of high-definition video streams as cameras are deemed as one kind of optical sensors. This capability has been anticipated for a long time by many applications. In the future battlefield, thousands of mobile objects, such as helicopters, vehicles, tanks, and soldiers, will be installed with high-definition or standard-definition cameras. The next-generation WSN is suitable for transmission of videos captured by these cameras back to

the headquarter. Videos are going to be stored, managed, analyzed, and even redistributed to the related people. Similar application scenarios include large scale emergent rescue during natural disasters, security level scale up for a critical event, and prompt treatment to riots or chaos and so on.

To fully and efficiently realize the above application, several challenges are identified. The first one is that different video formats produced by the front-end cameras may complicate the video management and analysis task. Besides, the large size of the “raw” video brings severe pressure on the storage disks and processing power. Finally, it is obligatory that the videos redistributed to the users are compatible with and have the same quality on the heterogeneous terminal devices.

The video transcoding system (VTS) is widely believed to address these challenges. It takes large amounts of video files with various formats as inputs and videos with the uniform format and much less size as outputs. Single-machine-based approach and cloud-based approach are two

options to implement the VTS. Nevertheless, the cloud-based approach has several advantages over the single-machine-based approach, especially for the WSN application scenario. Scalability and fault tolerance are two major advantages, for instance. Scalability means that Hadoop nodes are able to easily join/leave the WSN if more/less video streams require transcoding. The VTS continues operating well even if some Hadoop-nodes fail to work, which is the frequently happening case for the WSN applications.

This paper aims to introduce a cloud-based, more specifically, Hadoop-based, video transcoding system to fulfill the vision of bearing hundreds of HD video streams in the next generation WSN, with a discussion on optimization of several significant parameters.

We carry out two sets of experiments: (1) various numbers of Mappers and (2) different file sizes with fixed video duration. Our results show that selecting a proper number of Mappers with respect to the size of video file will obtain optimal performance in terms of transcoding time. The transcoding time depends principally on the duration of video files rather than their sizes.

The rest of the paper is organized as follows. Section 2 describes the related work. Section 3 discusses the whole VTS architecture and function components. In Section 4, we share our experiment design and result analysis. Section 5 concludes this paper and points out some future works.

## 2. Related Work

In recent years, the research area on video transcoding has gained more and more attention. A Hadoop-based distributed video transcoding system [2] is designed by using HDFS [3] (Hadoop Distributed File System) to store video resources and applying both MapReduce [4] and FFMPEG [5] technology to do transcoding. A video segmentation strategy on distributed file system (i.e., HDFS) is proposed, in which performance test is also conducted and the result shows that the 32MB size of segment has the most outstanding performance. However, there are still many works left untouched in the paper, such as other options of Hadoop parameters to optimize, for example, block size, cluster size, and so forth. In another paper [6], cloud-based smart video transcoding system is proposed. A cloud server is a logical server that is built, hosted, and delivered through a cloud computing platform over the Internet. It is responsible for providing and managing the user's applications, such as VOD on mobile phones, and storing video materials in the cloud. Service subscribers may access the computing resources by their own computers, smart phones, and so forth. The cloud computing technology could speed up video transcoding, owing to the benefits of parallel computing and MapReduce, that is, splitting original big video file to small portion and forwarding to different nodes in cloud environment and executing transcoding of each small portion in parallel. However, in this paper, no transcoding performance was evaluated. Similarly, another Hadoop-based distributed video transcoding system in a cloud computing environment is proposed to transcode various video codec formats into the

MPEG-4 video format [7]. In this system, MapReduce framework, HDFS (Hadoop Distributed File System) platform, and media processing library Xuggler are applied to implement it. Meanwhile, performance evaluation in a Hadoop-based distributed video transcoding system is also conducted [8]. In order to present optimal Hadoop options for processing video transcoding, the experiment data is collected with changing cluster size, block size, and block replication factor. However, these experiments are not enough to figure out the relationship between Hadoop and video transcoding, for example, how to set the number of Mappers for certain video files.

In this paper, we introduce a VTS with Apache Hadoop [9] (including MapReduce framework and HDFS platform, etc.), media processing library FFMPEG, and web server Apache Tomcat [10], in order to verify how it could speed up video transcoding for big video files. This paper also analyzes the experiment results with consideration of factors about the number of Mappers that do the transcoding work, and duration and size of video files, which have not been studied before. It fills the gap in the research field of Hadoop-based video transcoding and contributes greatly to the choice of optimal parameters.

## 3. Application Scenario and System Architecture

**3.1. Application Scenario.** Figure 1 illustrates the application scenario of the Hadoop-based VTS in the WSN.

Video capture devices, such as cameras, generate large amounts of videos and transfer them to Hadoop subclusters through the WSN. Then the video transcoding system deployed on the Hadoop cluster will transcode these video files into the MP4 files and store the transcoded files in the HDFS. The system clients, who may use smart watches, smart phones, and other wearable devices, are able to access the system web server to request for transcoded videos or receive recommended videos from the Hadoop cluster. They can also act as video content providers and then submit their own videos to launch new transcoding tasks.

There may be several Hadoop subclusters in one WSN. These Hadoop subclusters can be deployed on the same WSN node or different nodes. From a logical view point, all of these subclusters constitute the whole Hadoop cluster of one VTS. The decentralization property of Hadoop system matches quite well with that of WSN. Seamlessly integrating the Hadoop with WSN would make the solution more robust, scalable, and fault-tolerant.

**3.2. System Architecture.** This subsection describes the architecture of the video transcoding system. As shown in Figure 2, it consists of a web server and a Hadoop cluster. The web server accepts users' requests and invokes remote video transcoding function deployed on Hadoop cluster. In the following paragraphs, we will introduce each part of the video transcoding system.

**(1) Hadoop Cluster.** Hadoop is an open-source software framework that derives from Google's Map/Reduce and

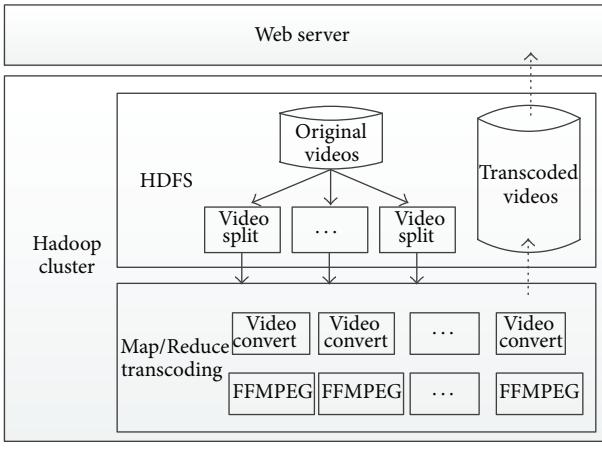
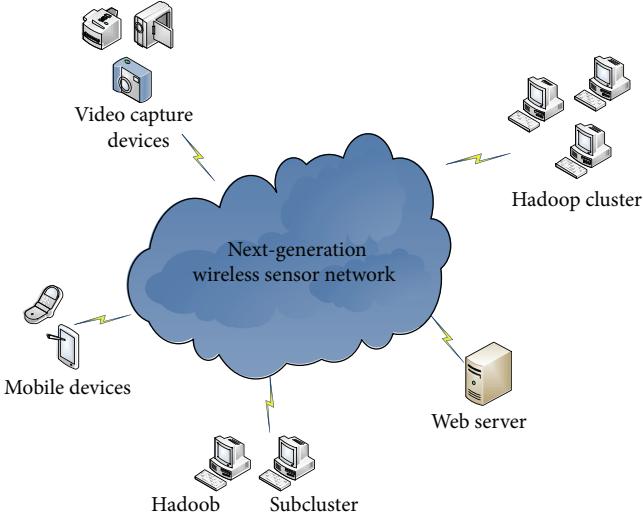


FIGURE 2: System architecture.

Google File System papers. It provides strong distributed computing capability and scalable storage capacity for huge data processing based on the Hadoop cluster composed by a set of commercial and low cost machines. In the Hadoop cluster as shown in Figure 3, it incorporates a master/slave model; that is, a Hadoop cluster usually includes a single machine designated as a master node and all others act as slave nodes. Based on this model, Hadoop provides a distributed file system known as HDFS and the Map/Reduce function, which are used for data storage and data processing.

(2) **HDFS**. HDFS is a distributed file system implemented in the Hadoop cluster. It stores large data files across dozens of machines. The master node, also known as the NameNode, is the centerpiece of HDFS. It keeps a directory tree for all the files in HDFS and tracks the location of each part of data files. Moreover, it divides large computation tasks, files

into small ones, and distributes them to slave node, known as DataNode. The slave nodes provide storage space and computing resources for HDFS. Large data files are split into several parts and stored across slave nodes. And computation tasks are executed on slave nodes. The results are collected by the master node and combined into an output form.

In our system, the HDFS is mounted on the web server using FUSE tool. From a logical view, it can be treated the same as the local file system. The original videos and transcoded videos are stored in HDFS. The NameNode and DataNodes can colocate with one WSN node or connect with different WSN nodes.

(3) **Map/Reduce Model**. Map/Reduce is the programming model for processing parallelizable problems across large datasets with some distributed algorithms running on a cluster of machines. It involves a Map procedure and a Reduce procedure. As indicated in Figure 4, the input data of Map/Reduce model will be split into small datasets. Then these small datasets are mapped into some Map processors' work space and handled. A shuffle/sorting mechanism will collect results of the Map procedure and organize them into lists. The Reduce procedure will run on these lists and transform the results into the final output data.

As a popular implementation of Map/Reduce model, Hadoop uses the master node to complete the Map and Reduce procedure. In the Map procedure, the master node divides the input, such as large data files or computation tasks, into small ones and then generates (key, value) pairs. Depending on the shuffle and sorting mechanism, it distributes them to the slave nodes. After small tasks are finished, the master node collects results from slave nodes to accomplish the Reduce procedure.

(4) **Video Transcoding**. The format of a video includes many parameters, for example, bit rate, frame rate, spatial resolution, coding syntax, content, and so forth. Video transcoding is the technology used to convert a video from one format to another one. There are numerous video transcoding mechanisms. Some main transcoding techniques are briefly introduced as follows [11–13].

(a) **Bit-rate transcoding**: it is usually applied to reduce the bit rate with the same complexity and quality if possible. The likely scenario is to convert the video resources for television broadcast and Internet streaming. There are some bit-rate transcoding architectures: open-loop transcoders, cascaded pixel-domain transcoders, and DCT-domain transcoders.

(b) **Spatial and temporal transcoding**: apart from bit-rate transcoding, spatial and temporal transcoding is also used to convert compressed video for communication networks video viewer. There are a multitude of challenges, one of which is how to derive a new set of motion vectors. Many relevant researches have been performed to solve them. In addition, some architectures are proposed to enhance its performance, such as DCT-domain architecture, a hybrid DCT/pixel-domain transcoder architecture.

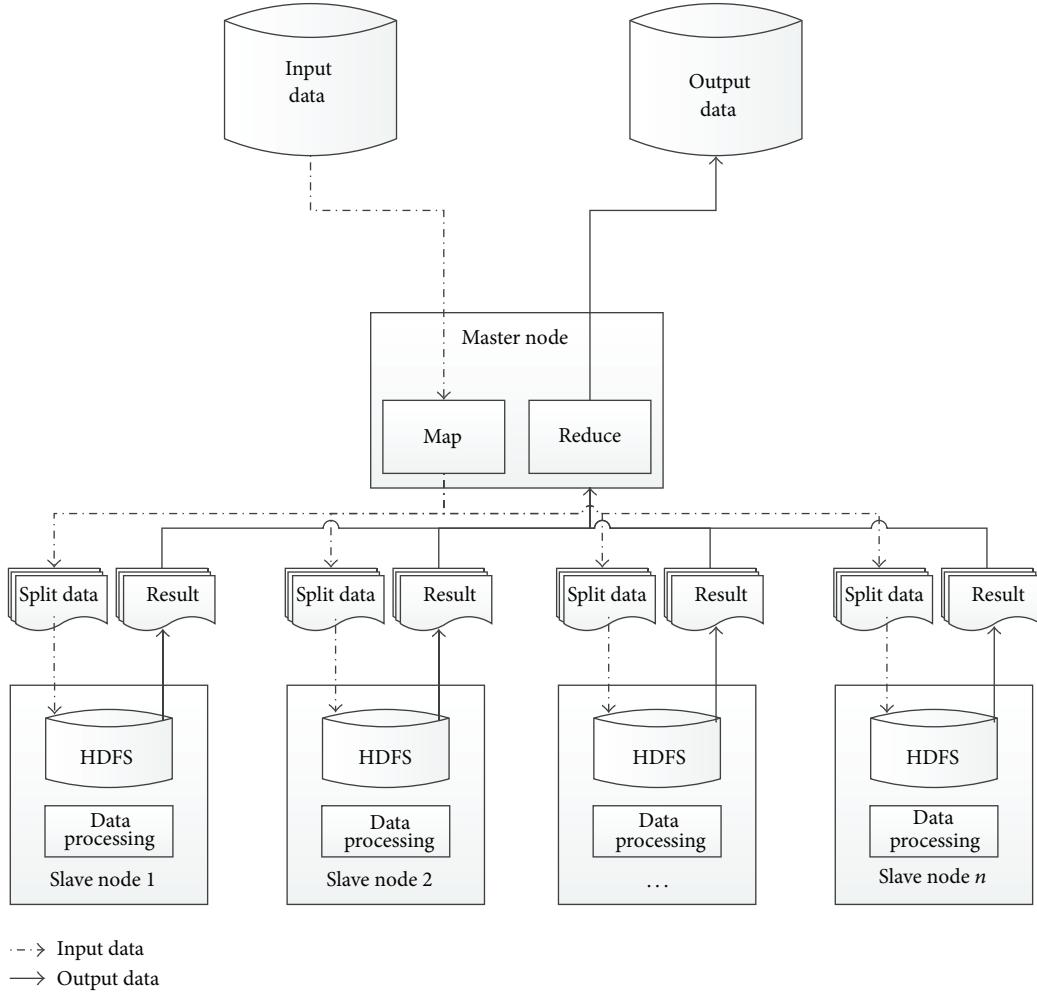


FIGURE 3: Hadoop cluster work flow.

(c) Standards transcoding: coding standards, for example, MPEG-2, MPEG-4, and so forth, are one of the important characteristics of video. Besides the conversion in bit rate and resolution, coding standard will also be changed from one to another in some applications.

Our video transcoding mechanism, by itself, is based on the spatial and temporal transcoding technique together with standards transcoding technique. We conduct video transcoding from AVI to MPEG-4 and meanwhile from  $800 \times 480$  to  $640 \times 240$ . More specifically, we use a Mapper to divide large video files into small ones. Afterwards, these small files are distributed to be stored in HDFS. There is no Reduce procedure in order to eliminate the time consumption. All of the transcoding work is done on the slave nodes. We use the open-source software FFMPEG to split videos and do transcoding work. FFMPEG is a free multimedia software framework used to handle various multimedia data. It also provides some library files related to audio/video codec. In addition, it can be compiled under most operating systems.

## 4. Experiments

**4.1. Experiment Settings.** Figure 5 illustrates the experiment platform deployed on three Dell servers and one PC. The Hadoop cluster is composed of three Dell servers, which serve as one NameNode and two DataNodes. It does not hold lots of meanings if tens of DataNodes are deployed considering that the most usual application locations of WSN are outdoor, in mobile status, or even in barren areas. Each Hadoop cluster node is running on the Linux OS (Debian 3.2.46 x86\_64). No matter the NameNode and DataNode are, each Hadoop-node is equipped with two Intel Xeon 8 core 2.00 GHz processors with 64 GB registered ECC DDR memory and 3 TB SATA-2. The web application is deployed on Tomcat web engine running on a PC machine, which is equipped with Linux OS (Ubuntu Server 3.5.0 x86\_64), Intel Core 2 Duo CPU 3.0 GHz with 2 GB registered DDR memory, 320 GB SATA-2. Java 1.6.0\_37, Apache Tomcat 7.0.39, Hadoop-0.20.2, and FFMPEG 1.0.6 are the other components used in the platform.

Several video data sets are used for the system performance evaluation. The video data sets are generated by emerging replications of the original file using the Format

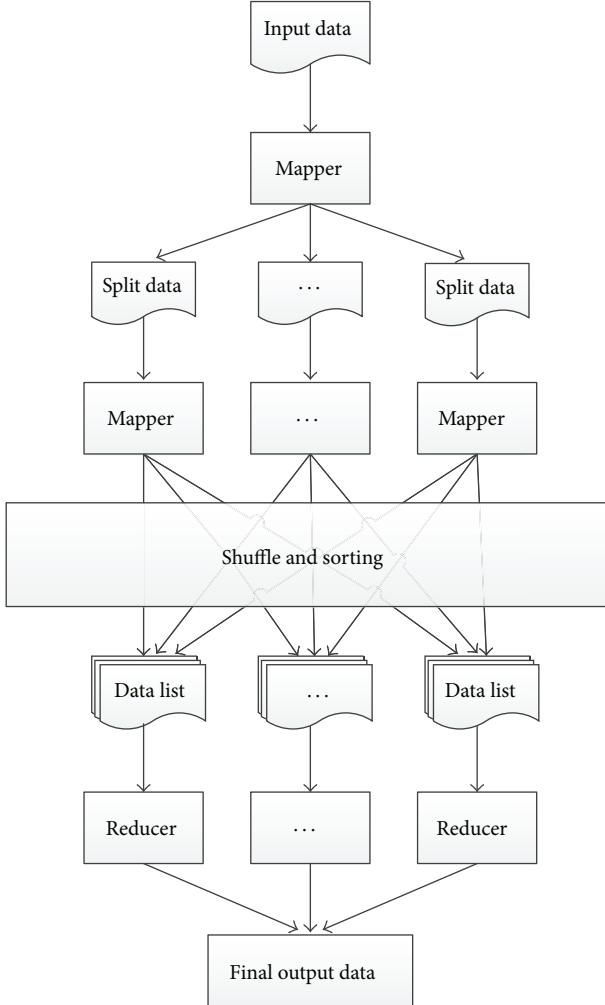


FIGURE 4: Map/Reduce model.

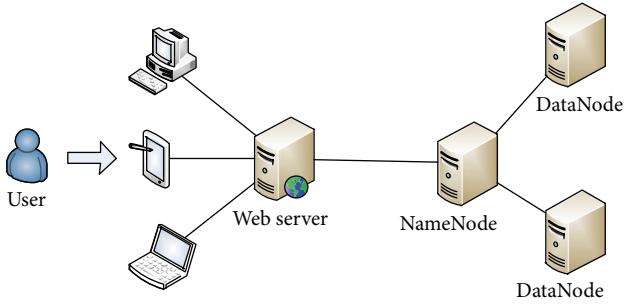


FIGURE 5: Experimental platform.

Factory [14]. Each original file size is 80MB. Table 1 lists the parameters for original and transcoded video file.

**4.2. Experimental Results and Analysis.** This subsection focuses on discussing and analyzing the experiment results. The transcoding time consumption is employed as the performance metric. In the experiments, we choose parameters

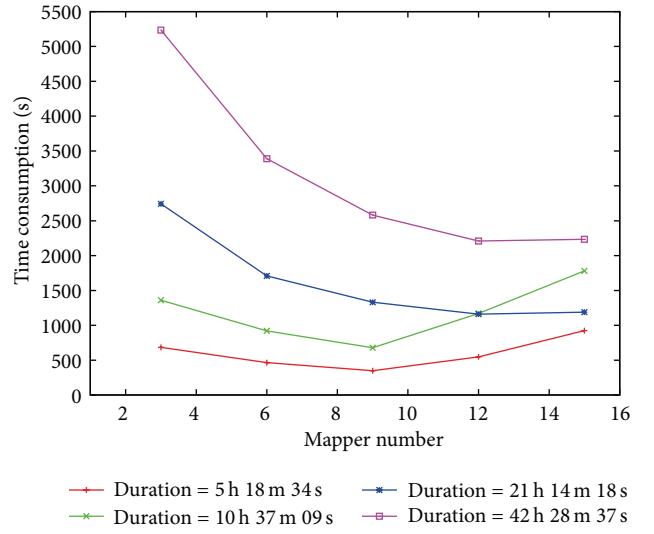


FIGURE 6: Effect of number of Mappers.

TABLE 1: Experiment settings.

Parameter	Original single file	Transcoded file
Codec	DivX	MPEG-4
Format	AVI	MP4
Duration	31 min 21 sec	31 min 21 sec
Resolution	800 × 480	640 × 240
Frame rate	15 fps	15 fps

about the number of Mappers that do transcoding work and duration and size of video files. We carry out a series of experiments based on these factors.

(1) *Effect of Number of Mappers.* We design the first set of experiments by changing the number of Mappers while fixing file size and duration. The range of Mappers' number is 3, 6, 9, 12, and 15. Table 2 shows the video data sets used in these experiments and Table 3 lists the experiment results.

Figure 6 indicates the effect of number of Mappers. For the files whose durations are 5 h 18 m 34 s and 10 h 37 m 9 s, as the Mappers' number increases, the time consumptions of transcoding decrease at the beginning and reach the lowest point when the number is 9. Later, they increase obviously as the number of Mappers is greater than 9. For the files whose durations are 21 h 14 h 18 s and 42 h 28 m 37 s, time consumptions go down when Mappers' number increases.

In the Hadoop system, a Mapper is designated to process a split part of video file and is invoked by a DataNode. The more the Mappers, the more the transcoding tasks distributed to DataNode, which accelerates the transcoding process. On the other hand, there are also queues in each DataNode keeping the disengaged transcoding tasks waiting, which retards the transcoding process. There should be a "Laffer curve" just as the case in economics [15].

The curve depicted in Figure 6 reveals that there exists an optimal value of the number of Mappers for the Hadoop-based VTS and the value is closely related to the size of files.

TABLE 2: Video data sets for performance evaluation.

File size	1 GB	2 GB	4 GB	8 GB
Total duration	5 h 18 m 34 s	10 h 37 m 09 s	21 h 14 m 18 s	42 h 28 m 37 s

TABLE 3: Transcoding time with different Mappers' numbers.

Mapper	Video data sets			
	1 GB/5 h 18 m 34 s	2 GB/10 h 37 m 09 s	4 GB/21 h 14 m 18 s	8 GB/42 h 28 m 37 s
3	684 s	1359 s	2743 s	5235 s
6	464 s	921 s	1709 s	3390 s
9	349 s	678 s	1330 s	2581 s
12	547 s	1170 s	1160 s	2210 s
15	925 s	1781 s	1189 s	2234 s

TABLE 4: File sizes and results—first file.

First file size	Time consumption
0.7 GB	317 s
1.0 GB	349 s
1.3 GB	309 s
2.0 GB	340 s

TABLE 5: File sizes and results—second file.

Second file size	Time consumption
1.7 GB	578 s
2.0 GB	678 s
2.9 GB	576 s
4.0 GB	601 s

The optimal value is 9 for the video files whose durations are 5 h 18 m 34 s and 10 h 37 m 9 s in experiment settings of this paper, while, for the other two cases, the optimal value should be around 12.

(2) *Effect of Duration and Size.* The second set of experiments tries to explore the relationship of the transcoding time with the size and duration of video files. In these experiments, we specify the Mappers' number as 9. By fixing the duration of two video files and changing their sizes, we investigate the transcoding time consumption. Tables 4 and 5 indicate the system performance with respect to various file sizes. The durations of two video files are 5 h 18 m 34 s (Table 4) and 10 h 37 m 09 s (Table 5), respectively.

Figure 7 demonstrates that time consumptions fluctuate a little when file sizes increase and meanwhile the durations are fixed. The time consumption of video transcoding depends principally on the duration of video files rather than their sizes. It suggests that duration-based splitting mechanism would be more controllable than the size-based method.

## 5. Conclusion and Future Work

In this paper, we propose a Hadoop-based VTS integrating several key components including HDFS, Map/Reduce,

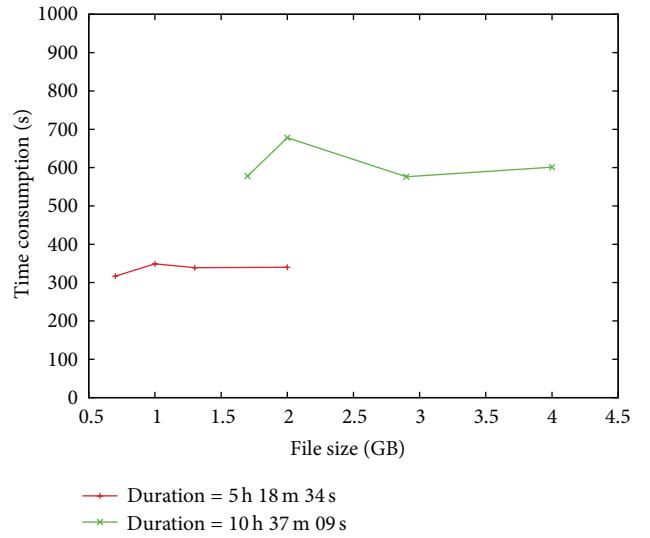


FIGURE 7: Effect of duration and size.

FFMPEG, and Tomcat, with a discussion on several significant parameters. Three prominent results are achieved through the experiments: (1) it is clear that there is an optimal value of the number of Mappers; (2) the optimal value is closely related to the file size; (3) the time consumption of video transcoding depends principally on the duration of video files rather than their sizes.

The inherited distribution property of the Hadoop system seems quite harmonious with the decentralization attribute of next-generation WSN. However, the research exploring their relationship and integrating them into a turn-key solution for many practical problems is still in its preliminary stage. As one of the first papers pioneering in this direction, this paper enlightens several directions for future work.

First of all, more experiments will be carried out not only in the area of standards transcoding and spatial transcoding, but also in the field of bit-rate transcoding, to meet the service requirements of the next-generation WSN, such as converting the video resources for video broadcast or streaming.

Secondly, some efforts have to focus on WSN's network planning and routing protocol optimization in order

to integrate seamlessly with Hadoop system. The normal Hadoop system generally operates in an indoor and machine-friendly environment with wired connections. However, the WSN system always works in outdoor locations with tough surroundings, such as severe interferences or extremely high or low temperatures, and so forth. Accordingly, the communication protocols among Hadoop nodes also have to be redesigned with the consideration of WSN characteristics.

Besides, some theoretical research will be done to find the optimal value of the Mappers' number. The mathematical model is going to be constructed, taking the Hadoop cluster size, block size, video file size, and block replication factor into account.

Finally, it is necessary to implement the Hadoop-based online or real time VTS for the next generation WSN, beside the offline version proposed in this paper. It makes the live video broadcasting, multicasting, and P2P streaming possible in the WSN with the online VTS at hand.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgment

This work is supported by Science and Technology Commission of Shanghai Municipality Research Project "Research of key technologies and applications of FOD detection in aircraft movement area" (Project nos. 13511503200 and 13511503202).

## References

- [1] O. Bejarano, E. W. Knightly, and M. Park, "IEEE 802.11ac: from channelization to multi-user MIMO," *Communications Magazine, IEEE*, vol. 51, pp. 84–90, 2013.
- [2] F. Yang and Q. W. Shen, "Distributed video transcoding on Hadoop," *Computer Systems & Applications*, vol. 20, no. 11, pp. 80–85, 2011.
- [3] HDFS, [http://en.wikipedia.org/wiki/HDFS#Hadoop\\_distributed\\_file\\_system](http://en.wikipedia.org/wiki/HDFS#Hadoop_distributed_file_system).
- [4] MapReduce, <http://en.wikipedia.org/wiki/MapReduce>.
- [5] FFmpeg, <http://www.ffmpeg.org/>.
- [6] A. A. Chandio, I. A. Korejo, Z. U. A. Khuhro et al., "Clouds based smart video transcoding system," *Sindh University Research Journal*, vol. 45, pp. 123–130, 2013.
- [7] M. Kim, Y. Cui, K. Lee et al., "Towards efficient design and implementation of a Hadoop-based distributed transcoding system," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 8, no. 2, pp. 213–224, 2013.
- [8] N. S. Chahal and B. S. Khehra, "Performance evaluation of a Hadoop-based distributed video transcoding system for mobile media service," ASTL Volume 2, Information Science and Technology(Part 1), 2012.
- [9] The Apache Hadoop Project, <http://hadoop.apache.org/>.
- [10] The Apache Tomcat Project, <http://tomcat.apache.org/>.
- [11] Y. Wu, A. Vetro, H. Sun, and S. Y. Kung, "Intelligent multi-hop video communications," in *Presented at the IEEE Pacific Rim Conference on Multimedia*, Beijing, China, 2001.
- [12] S.-F. Chang and A. Vetro, "Video adaptation: concepts, technologies, and open issues," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 148–158, 2005.
- [13] J. Xin, C.-W. Lin, and M.-T. Sun, "Digital video transcoding," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 84–97, 2005.
- [14] Format Factory, <http://www.pcfreetime.com/>.
- [15] Laffer curve, [http://en.wikipedia.org/wiki/Laffer\\_curve](http://en.wikipedia.org/wiki/Laffer_curve).