

CLOUDDMSS: CLOUD-BASED DISTRIBUTED MULTIMEDIA STREAMING SERVICE SYSTEM FOR HETEROGENEOUS DEVICES

¹MYOUNGJIN KIM, ²CUI YUN, ³SEUNGHO HAN, ⁴HANKU LEE

^{1,2,3,4}Department of Internet & Multimedia Engineering, Konkuk University
E-mail: {tough105, ilycy, shhan87, hlee}@konkuk.ac.kr

Abstract- With the recent appearance of various heterogeneous smart devices and the expansion of social network services, services that support the production and sharing of social media data are actively provided. The stable provision of such services requires a transcoding technology supporting the N-Screen service and distributed streaming technologies for providing a quality of service (QoS)-based rich media streaming service. However, the transcoding and streaming services through the existing distributed computing environment places a substantial burden on the Internet infrastructure and computing resources by requesting vast computing resources. In addition, in the current situation where system expandability and stability are required, there are intrinsic difficulties such as the requirement of a variety of distributed management and operation policies and techniques including a data recovery policy in the event of content loss, an automatic recovery policy, an automatic content storage technique, and a cluster relocation technique. Therefore, in order to overcome such difficulties, in this paper, we have proposed a cloud-based distributed multimedia streaming service (CloudDMSS). Using the MapReduce framework, a core element technology of cloud computing, the proposed service enables high-speed transcoding and provides a stable streaming service by building a streaming service that is applied with a distributed streaming task distribution technique. In addition, with the application of the Hadoop policies and the Hadoop distributed file system (HDFS), the proposed service secures system expandability and stability.

Keywords- Streaming Service, Distributed Transcoding, Cloud Computing, Hadoop

I. INTRODUCTION

With the recent spread of social media-based social networking services (SNSs) and various heterogeneous smart devices, the multimedia-computing environment for supporting individual media services has been studied at a fast pace. In particular, as streaming services for individual media contents have become a major focus, cloud computing-based multimedia techniques that support content delivery, distribution processing, and storage, and quality of service (QoS)-based rich media services have emerged as new alternatives.

An efficient support for a media streaming service requires distributed computing-based transcoding and streaming that provide the distribution of contents. However, transcoding and streaming of a large volume of content consume vast amounts of computing resources (CPU, RAM, HDD, network bandwidth, etc.), substantially burdening the existing Internet infrastructure and computing resources. In addition, streaming in a distributed environment requires policies and techniques including data replication and automatic recovery after system shutdown and content loss, distributed content storage for the expansion of the system, cluster relocation, and namespace control of the file system. It is very difficult to realize the above policies and techniques while establishing a streaming service in a distributed environment with high availability and expandability considered.

In this paper, to overcome such limitations, we propose a cloud-based distributed multimedia streaming service (CloudDMSS) system, which is a Hadoop-based social media streaming system in a cloud-based environment. This service consists of three core modules.

The first module is the Hadoop-based distributed multimedia transcoding (HadoopDMT) module that transcodes a large portion of a media file into an MPEG4 file, which is a suitable format for heterogeneous smart devices. The second module, the Hadoop-based distributed multimedia streaming (HadoopDMS) module, copies and saves the transcoded contents in the Hadoop distributed file system (HDFS) by a unit block and automatically recovers them through NameNode when the files are lost or the system shuts down, securing the system's reliability and providing an efficient streaming service at the same time. The third module, the cloud multimedia management (CMM) module, schedules the task distribution for an efficient streaming distribution when transcoding is complete and manages the system on the whole.

Section 2 explains a service scenario of the CloudDMSS system. Section 3 presents the core architecture of our system and explains the core modules in detail. Section 4 discusses the issues with the system implementation and an early-stage prototype. Section 5 shows the workflow of CloudDMSS. Section 6 presents the conclusions and

the future research direction.

II. CLOUDDMSS SERVICE MODEL

The proposed cloud-based distributed multimedia streaming service facilitates the sharing of an individual's media (films, music, music videos, etc.) that are produced using mobile media services and SNSs with others and provides a high-quality streaming service. Fig. 1 shows the overall service scenario of the proposed model.

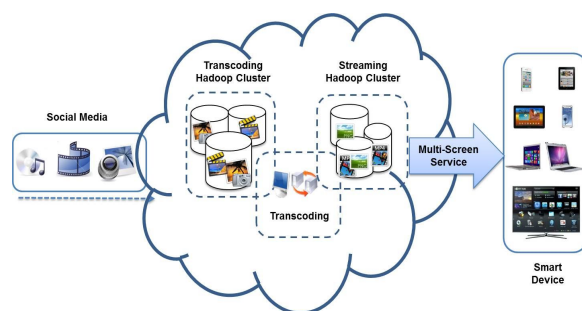


Fig. 1. CloudDMSS service model and scenario

When a user uploads content to the CloudDMSS for sharing his contents with ours, the pertinent content, first, is stored in Transcoding Hadoop Cluster of the dual Hadoop clusters in the system. The service model designed in this study embraces N-Screen services that can play the said content in multiple devices after a transcoding process. For a substantial time reduction in transcoding the uploaded large-volume individual media, this service model provides a distributed media transcoding function that is applied with a distributed framework—Hadoop MapReduce. Once the transcoding is complete, for a streaming service, the transcoded content is automatically transferred to Streaming Hadoop Cluster. The transferred content is serviced to users via multiple streaming servers that are contained in the cluster. At the time of service, to reduce the content delay and the network traffic bottleneck occurring in the streaming servers, the streaming resource-based connection (SRC) algorithm is applied for an efficient distribution to the streaming servers, which is a QoS-based distributed streaming service.

III. SYSTEM ARCHITECTURE OF THE CLOUDDMSS SYSTEM

This system consists of three core modules: the Hadoop-based distributed multimedia transcoding (HadoopDMT) module, the Hadoop-based distributed multimedia streaming (HadoopDMS) module, and the cloud multimedia management (CMM) module.

The HadoopDMT module is a Hadoop framework-based distributed transcoding module that

transcodes a large volume of a media file into an MPEG4 file, which is an appropriate format for heterogeneous smart devices. First of all, various types of multimedia files that are generated in a certain time unit are segmented to a unit of 64-MB blocks and stored in the HDFS, after which two copies of each block are generated and stored in another node. Hadoop's NameNode manages the copied blocks. If a system error and a data loss occur, NameNode detects them and runs an automatic recovery. The mappers of the MapReduce framework transcode the segmented blocks in parallel and the reducers combine the segmented blocks, thereby generating the complete transcoded content.

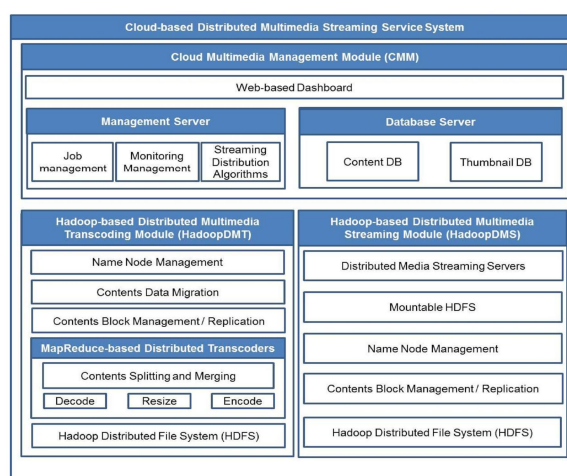


Fig. 2. System architecture of CloudDMSS

The main role of the HadoopDMS module is to provide constant streaming. HadoopDMS consists of multiple streaming servers and content storage servers in which the transcoded contents are stored. The transcoded contents in HadoopDMT are automatically transferred to the content storage servers in HadoopDMS. The storage method and the distribution management policy of HadoopDMS are the same as those of HadoopDMT. When streaming is requested, HadoopDMS selects the best streaming server through the SRC algorithm, connects the user and the selected server, and provides streaming.

The CMM module is a core module for the CloudDMSS system; it manages and controls all the functions executed in the overall streaming process and monitors the system use rate and the progress of a task. The CMM module also forms and provides a Web-based interface that enables the easy control and monitoring of the overall functions. In particular, the management server of the CMM module efficiently distributes tasks. The two most popular streaming distribution methods are round robin (RR)—scheduling in the order in which the streaming requests are received—and least connection (LC)—scheduling with the least connected server.

Because these two methods fail to take into account the use rates of the streaming servers and the traffic of streaming transmission, their streaming distribution, which is largely influenced by computing resources, is not very efficient. Therefore, in consideration of the system use rate and the traffic rate of the streaming transmission of each streaming server, the proposed system uses the SRC algorithm and provides efficient distributed streaming.

IV. IMPLEMENTATION AND PROTOTYPE OF CLOUDDMSS

In this study, we used a cloud cluster consisting of 28 nodes for the implementation of the CloudDMSS system architecture. Each node consisted of an Intel Xeon quad-core 2.13-GHz processor, 16-GB DDR memory, and 1-TB HDD, and the operating system was Ubuntu 10.04 LTS. The 28 nodes were connected through a 100-Mbps Ethernet. Fig. 3 shows the architecture of the CloudDMSS system.

The CMM module was constructed using a Tomcat-based management server and a My-SQL-based content management DB server. The HadoopDMT module was built with a NameNode server and 12 DataNode servers and used the HDFS. The HadoopDMS module was constructed using three NginX-based streaming servers and ten Hadoop DFS-based content storage servers. A single server cluster was applied with a dual-Hadoop DFS structure to distribute the load of a task in the transcoding and streaming processes.

Fig. 4 presents a service prototype of CloudDMSS. This service is compatible with all browser types. The image on the left in Fig. 4 shows a sample shot for the personal computer (PC)-based service, and that on the right shows a sample shot for the mobile-based service. Once the user selects a thumbnail image, he can enjoy the corresponding transcoded content on various heterogeneous devices via streaming.

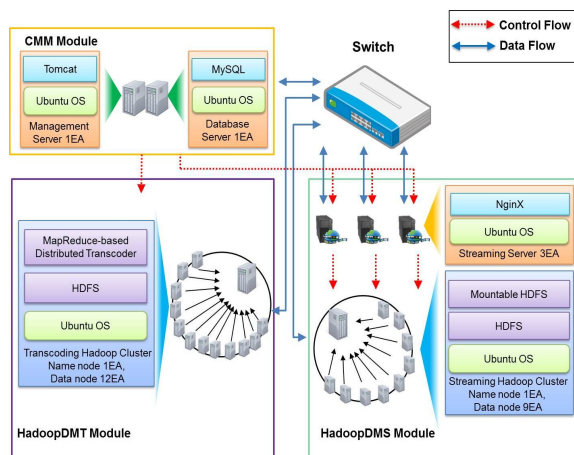


Fig. 3. System architecture of CloudDMSS

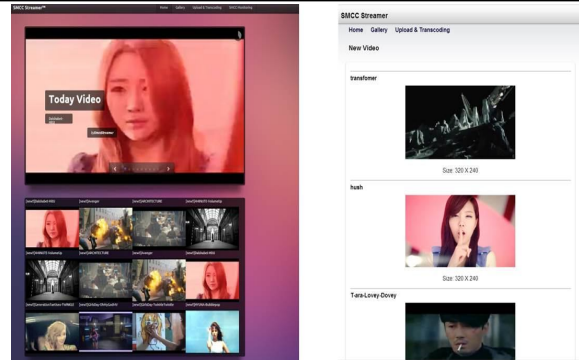


Fig. 4. Prototypes of the CloudDMSS service

V. WORKFLOW OF CLOUDDMSS

In this section, we define the workflow of the sequential tasks in CloudDMSS during the streaming service deployment process in the overall system. Figure 5 shows the overall workflow of CloudDMSS. The CloudDMSS workflow is divided into two parts: content transcoding processing and distributed streaming processing.

First, we introduce the workflow required for content transcoding processing to stream content to heterogeneous devices. Figure 6 shows a diagram of the workflow of content transcoding processing. Users and administrators can access our user interface via CMM to upload their own original multimedia content. After an upload task is complete, they select the user-requested options (resolution, frame rate, bit rate, codec, and container) for transcoding and then request a content transcoding task from a MapReduce-based distributed transcoder of CMM. Immediately after the task is requested, CMM prepares a preprocessing procedure for the task. The preprocessing procedure comprises two steps. First, CMM creates an SSH shell in both HadoopDMT and CMM to allow remote command execution and secured content migration between both components. Second, the file names of the transcoded and original contents are the same, so CMM creates a new folder to store the transcoded contents to avoid any content loss due to file redundancy. After the preprocessing procedure is complete, a management server module in CMM performs the transcoding task by operating the MapReduce-based distributed transcoder in a distributed manner. Subsequently, the management server module commands a thumbnail extraction task from a thumbnail image extractor in HadoopDMT, and the extractor extracts thumbnail images of the transcoded content using FFmpeg supported by Xuggler libraries. In the next step, the transcoded content and extracted thumbnail images are migrated to HadoopDMS via a content migrator in HadoopDMT. After migration, the original multimedia content, transcoded content, and thumbnail images are deleted by the management

server, which then registers the thumbnail images to a thumbnail database in a CMM database server, as well as content information, including the file name and location, to a content database in the same server.

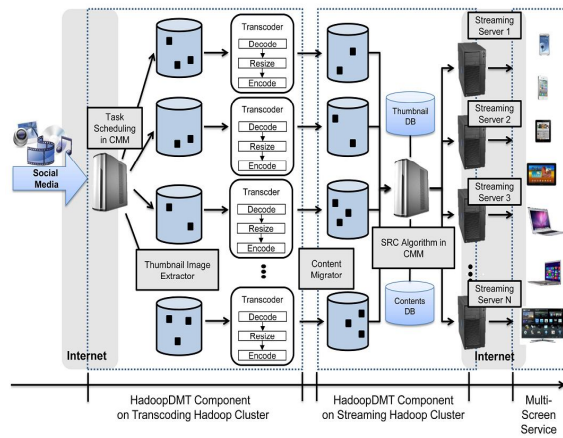


Fig. 5. Overall workflow of CloudDMSS

Second, the distributed streaming processing workflow with a streaming job distribution scheme is the most important part of the overall processing flow. Figure 7 illustrates the distributed processing workflow. Thumbnail images are updated periodically in a Web interface and the metadata of the thumbnail images are registered each time a transcoding task is completed. When users select and request media content by clicking a thumbnail image, CMM obtains the location and file name of the selected media via a content database. CMM then requests the status of streaming servers via a system status collector and selects the optimal streaming server from those in HadoopDMT based on the SRC algorithm, as described in section 3.2.3. The content streaming service begins after the optimal streaming server is selected.

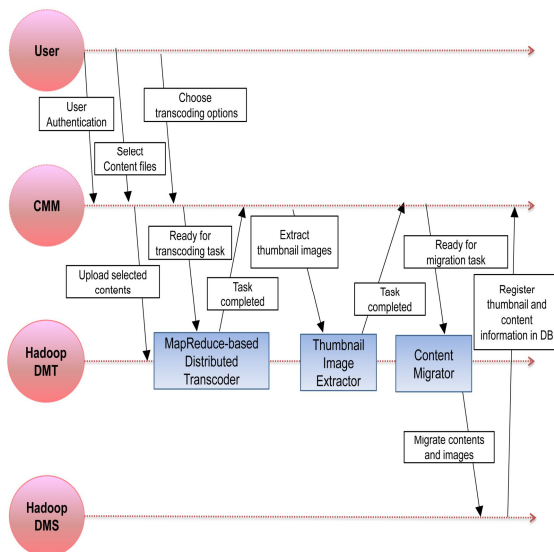


Fig. 6. Diagram showing the workflow of content transcoding processing

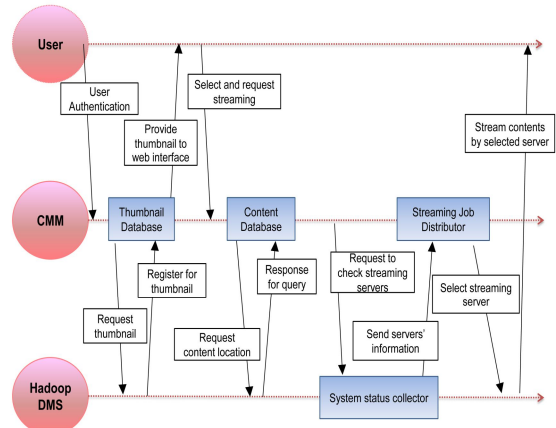


Fig. 7. Diagram showing the workflow distributed streaming processing

CONCLUSIONS AND FUTURE STUDY

In this paper, we proposed the CloudDMSS system, a cloud-based distributed multimedia streaming system, for providing a stable social media-based streaming service. The proposed system includes a high-speed distributed transcoding function, which is a core cloud computing technique using the MapReduce framework, and includes a distributed streaming function based on the SRC algorithm in consideration of the use rate of a streaming resource. The biggest benefit of this system is that the system stability and expandability can be secured by introducing the Hadoop-based distributed management and file management system policies across the system. In the future, for a stable operation and expansion of the proposed system, we will confirm its performance by reviewing the enhancement techniques, verifying the transcoding performance, and including a performance test in which the system is set up in a commercial cloud.

REFERENCES

- [1] S. Kim, "Design of a cloud-based media service platform," *Information and Communications Magazine*, vol. 30, no. 6, pp. 55-61, 2013.
- [2] W. Zhu, C. Luo, J. Wang, and S. Li, "Multimedia cloud computing," *IEEE Signal Processing Magazine*, vol. 28, pp. 59-69, 2011.
- [3] Apache Hadoop, <http://hadoop.apache.org>.
- [4] S. Han, M. Kim, C. Yun, and H. Lee, "A streaming task distribution algorithm for distributed media servers in cloud computing environment," *Proceedings of Conference of the Korean Institute of Information Scientists and Engineers*, pp. 1337-1339, June 2013.
- [5] M. Kim, S. Han, Y. Cui, H. Lee, "CloudDMSS: robust Hadoop-based multimedia streaming service architecture for a cloud computing environment," *Cluster Computing*, In Press.
- [6] M. Kim, S. Han, Y. Cui, H. Lee, C. Jeong, "A Hadoop-based multimedia transcoding system for processing social media in the PaaS platform of SMCCSE," *KSII Transactions on Internet and Information Systems*, vol. 6, no. 11, pp. 2827-2848, November 2012.

- [7] Z. Quyang, F. Dai, J. Guo, Y. Zhang, "VTrans: A distributed video transcoding platform, Lecture Notes in Computer Science, vol. 7733, pp. 532-534, January 2013.
- [8] C. Yang, Y. Chen, Y. Shen, "The research on a P2P transcoding system based on distributed farming computing architecture," In. Proc. Of KESE, pp. 55-58, December, 2009.
- [9] H. Sanson, L. Loyola, D. Pereira, "Scalable distributed architecture for media transcoding," vol. 7439, pp. 288-302, September 2012.
- [10] A. Kala Karun, K. Chitharanjan, "A review on Hadoop – HDFS infrastructure extensions, article no. 6558077, pp. 132-137, April 2013.
- [11] M. Kim, S. Han, C. Yun, H. Lee, "Distributed multimedia streaming service algorithms over cloud computing environment," WIT Transactions on Information and Communication Technologies, vol. 49, pp. 793-801, November 2013.
- [12] J. Dean, S. Ghemawat, "MapReduce: Simplified data processing on large clusters," Communications of the ACM, vol. 51, no. 1, pp. 107-113, January 2008
- [13] D. Jiang, B.C. Ooi, L. Shi, S. Wu, "The Performance of mapreduce: An in-depth study," Proceedings of the VLDB Endowment, vol.3, no. 1, September, 2010.

★ ★ ★