ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

# Dynamic Resource Allocation And Distributed Video Transcoding Using Hadoop Cloud Computing

Shanthi.B.R[1], Prakash Narayanan.C[2]

M.E, Department of CSE, P.S.V College of Engineering and Technology, Krishnagiri, Tamilnadu, India[1]

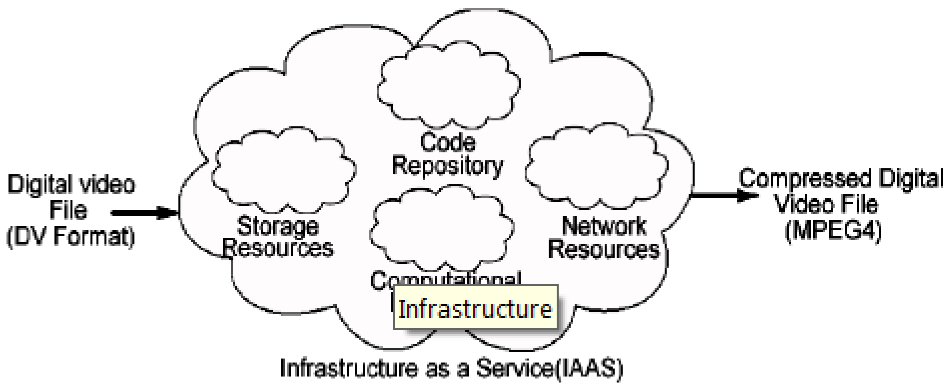
Assistant Professor, Department of CSE, P.S.V College of Engineering and Technology, Krishnagiri, Tamilnadu, India[2]

**Abstract:** Research work is carried out to design and optimize video transcoding using Hadoop. Cloud computing became quite a mature technology and is being adopted for every kind of applications in the real world; video transcoding is one of them. In this project, I propose a Hadoop Map Reduce based Distributed Video Transcoding System in a cloud computing environment that transcodes various video codec formats into the MPEG-4 video format. The cloud server provides and manages the applications and also storing the data remotely in cloud. Thus, the encoding time to transcode large amounts of video content is exponentially reduced, facilitating a transcoding function..I have proposed the solution that how video transcoding becomes smart and speeds-up due to the efficiency of cloud computing and allocate resources dynamically in cloud storage.

**Keywords:** Cloud Computing, Transcoding, Map Reduce, Hadoop, Dynamic Resource allocation, Windows Azure

## I. INTRODUCTION

Cloud computing is quickly becoming an effective and efficient way of computing resources and computing services consolidation In current and next decade, the technologies of cloud computing are expected to play a crucial role in motivating innovation in every industry like education, research, government, healthcare, media, Internet based business, manufacturing, video and audio providers etc. Cloud computing is an Internet based services, where we share some of the services like software, platform, infrastructure, storage, databases to computer or other devices on demand by the users. Services are sold on demand, for a minute/hourly basis, services are fully managed by the providers and consumer need only is a computer and Internet access.



ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

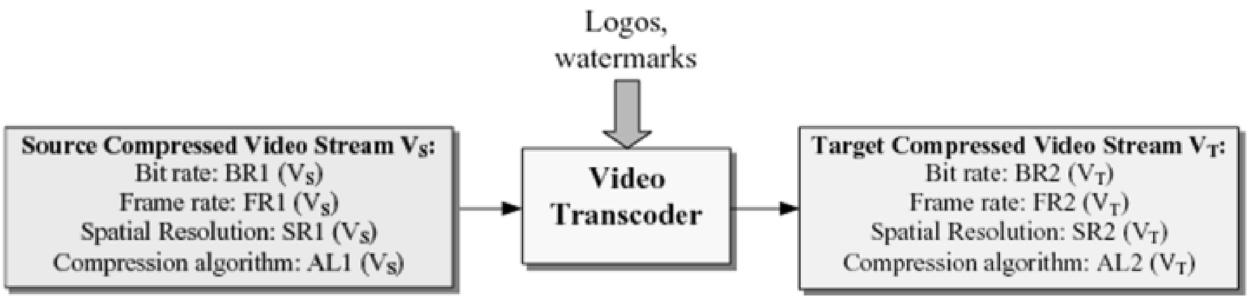
Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

VIDEO transcoding performs one or more operations, such as bit rate and format conversions, to transform one compressed video stream to another. Transcoding can enable multimedia devices of diverse capabilities and formats to exchange video content on heterogeneous network platforms such as the Internet. One scenario is delivering a high-quality multimedia source (such as a DVD or HDTV) to various receivers (such as PDAs, Pocket PCs, and fast desktop PCs) on wireless and wireline networks. Here, a transcoder (placed at the transmitter, This means that if you want to look at a specific document you have housed in the cloud, you must first establish an internet connection either through a wireless or wired internet or a mobile broadband connection. The benefit is that you can access that same document from wherever you are with any device that can access the internet. These devices could be a desktop, laptop, tablet, or phone. This can also help your business to function more smoothly because anyone who can connect to the internet and your cloud can work on documents, access software, and store data. Imagine picking up your smart phone and downloading a .pdf document to review instead of having to stop by the office to print it or upload it to your laptop. This is the freedom that the cloud can provide for you or your organization.

Receiver or somewhere in the network) can generate appropriate bitstream threads directly from the original bitstream without having to decode and re-encode. To suit available network bandwidth, a video transcoder can perform dynamic adjustments in the bit-rate of the video bitstream without additional functional requirements in the decoder. Another scenario is a video conferencing system on the Internet in which the participants may be using different terminals. Here, a video transcoder can offer dual functionality provide video format conversion to enable content exchange, and perform dynamic bit rate adjustment to facilitate proper scheduling of network resources. Thus, video transcoding is one of the essential components for current and future multimedia systems that aim to provide universal access




This paper provides a comprehensive survey of video transcoding techniques. We discuss various research issues arising in transcoding and illustrate them using an architectural approach. An architecture, which can be implemented in hardware or software, shows various algorithmic modules, as well as their operations. We present several transcoding architectures with varying levels of efficiency and functional modules. We categorize these architectures and present various examples within a category. We discuss various outstanding issues and provide future directions.

## II. RELATED WORKS

Owing to the increasing popularity of mobile media services, the development of distributed video transcoding approaches that exponentially reduce transcoding time and ensure video quality has become a challenge. The efforts in the investigation of such approaches have thus increased. In this paper, our system that utilizes HDFS, MapReduce framework, and Xuggler based on a cloud server is presented. Therefore, in this section, we describe HDFS and MapReduce framework.

Hadoop, inspired by Google's MapReduce and Google File System, is a software framework that supports data-intensive distributed applications handling thousands of nodes and petabytes of data. It can perform scalable and timely analytical processing of large data sets to extract useful information. Hadoop consists of two important frameworks: 1) Hadoop Distributed File System (HDFS), like GFS, is a distributed, scalable and portable file system written in Java. 2) MapReduce is the first framework developed by Google for processing large data sets.

The MapReduce framework provides a specific programming model and a run-time system for processing and creating large data sets amenable to various real-world tasks. This framework also handles automatic scheduling, communication, and synchronization for processing huge datasets and it has fault tolerance capability.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

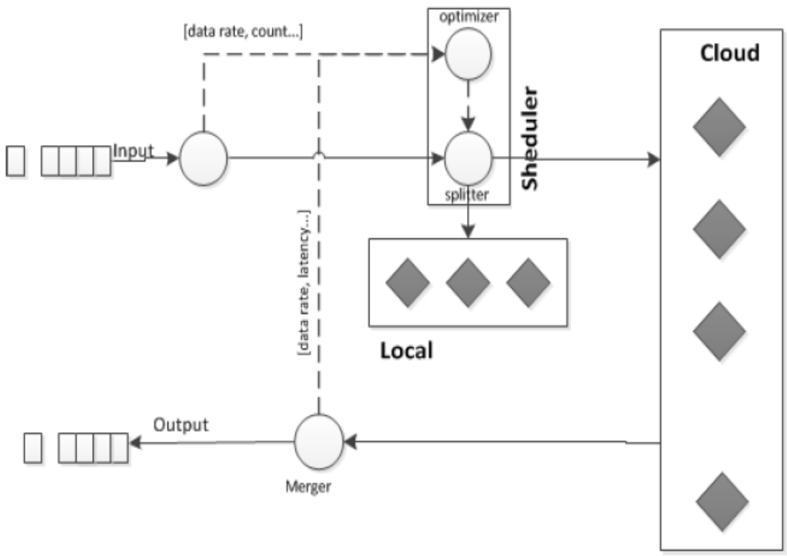
Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

## III. PROPOSED SYSTEM ARCHITECTURE

A Hadoop Map Reduce based Distributed Video Transcoding System in a cloud computing environment that transcodes various video codec formats into the MPEG-4 video format. The users may access the computing resources by using computer, tablet, notebook, Smartphone, pad computer or other devices. The cloud server provides and manages the applications and also storing the data remotely in cloud. For performance evaluation, I focus on measuring the total time to transcode a data set into a target data set for three sets of experiments. I have proposed the solution that how video transcoding becomes smart and speeds-up due to the efficiency of cloud computing and allocate resources dynamically in cloud storage and also analyze the experimental results, providing optimal Hadoop Distributed File System and MapReduce options suitable for video transcoding.

In order to process such large-scale video datasets, we are using the Hadoop MapReduce framework. I will dive further into the distributed video transcoder part of the framework that ingests the video into Hadoop, decodes the bit stream chunks in parallel and produces a sequence file (which is much more amenable for video analytics in Hadoop).

Hadoop framework stores large files in a distributed file system (HDFS: Hadoop Distributed File System) as small chunks of certain block size (typically 64MB) across a cluster of commodity machines. Given this framework, when the large input file to be processed is a text file and is split into 64MB chunks, each Mapper process can access the lines in each split independently. However, when the input file is video file (bitstream) and is split into many chunks, each Mapper process needs to interpret the bitstream chunk appropriately to provide access to the individual decoded video frames for subsequent analysis. In the following section, we will describe how each of the splits (64MB chunks) of a video bitstream can be transcoded into a sequence of JPEG images that can be subsequently processed by video analytics MapReduce jobs.




Architecture of Distributed Video Transcoder

Popular video compression formats such as MPEG-2, H.264 have a hierarchical structure in the bitstream. This hierarchical structure inherent in video bitstreams makes it possible to decode arbitrary input chunks. For concreteness, we choose MPEG-2 video as our example input video format (which is also the chosen format for DVDs, Digital Video Broadcast, possibly widely used surveillance video recording format).Our prototype distributed video transcoder breaks the task of decoding each chunk into the two following MapReduce jobs.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

Video Sequence Header MapReduce Job: This MapReduce job is essentially looking for video sequence level information that is present only in the first chunk of a large video file.

Video Decoder MapReduce Job: This MapReduce job uses the output of previous job as its input to configure the Video Decoder object to decode each chunk and writes the decoded frames in Hadoop-friendly Sequence File format as <key,value> pairs.

In the following subsections, we will describe at a high-level some elements of the MPEG-2 bitstream and in detail the two MapReduce jobs.

Video Sequence Header MapReduce Job

The goal of this MapReduce job is to extract the sequence level information from the first 64MB chunk of MPEG-2 video file and write this information as a text file in HDFS. The sequence information is available only once in the bitstream and contains information that is essential to set up a MPEG-2 video decoder object. Since our input is a Video file, we implemented a new FileInputFormat with its own record reader. The record reader corresponding to this FileInputFormat provides the following key-value pair to each Map process: <LongWritable, BytesWritable> where the input key is the byte offset into the file, and the BytesWritable is a byte array holding the video bitstream for the whole chunk. The record reader reads beyond the block boundary until it sees a GOP boundary. This ensures that all frames are decoded. Within each Map process, the key value is compared against zero to determine if it is the first chunk of the video file. If it is, then the bitstream is parsed to get the sequence level information and written to a HDFS file with the following name: "input_video_filename_sequence_info.txt". There is no need for a Reduce phase as we get the sequence information from the Map phase.

The parsing of the bitstream to get sequence information can either be done in Java or C code. In our implementation, we used libmpeg-2 (which is an open source MPEG-2 video decoder) to parse the bitstream through Java Native Interface (JNI). The hadoop framework supports JNI by allowing the user of the framework to submit a shared object file (*.so file) through "-files" command line option when the job is submitted.

Video Decoder MapReduce Job

The goal of this MapReduce job is to decode the individual 64MB chunks and create a sequence file containing the decoded video frames as JPEG images from each chunk. The InputFileFormat and the record reader are the same as in the first MapReduce job in the previous section. Hence, each Mapper's input <key,value> pair is <LongWritable, BytesWritable>.

Each mapper reads the sequence information file in HDFS (which is the output of the previous MapReduce job) and passes that information along with the bitstream buffer that came as input BytesWritable. The actual decoding of the bitstream is done in C using libmpeg2 through Java Native Interface.

Then, in the Map process we convert the decoded frames to jpeg images and create the following <key,value> pair as output of the Map process. The output key of the map process encodes the input video filename and chunk number as "input_video_filename_chunkid". The corresponding output value is BytesWritable and encodes the decoded frame number and the jpeg bitstream of the decoded frame. Note that the "chunk_id" can be derived from input key value which is the byte offset into the video file.

Each reducer simply writes the decoded frames from all the chunks into a sequence file as output format. Here is an example that illustrates the input and output <key,value> pairs for this MapReduce job. Let us assume the input video file is named "example.m2v".

1Input <key,value> for Mapper: <LongWritable, BytesWritable>

2 e.g <67108864, BytesWritable>

3Output<key,value> from Mapper: <Text, BytesWritable>

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

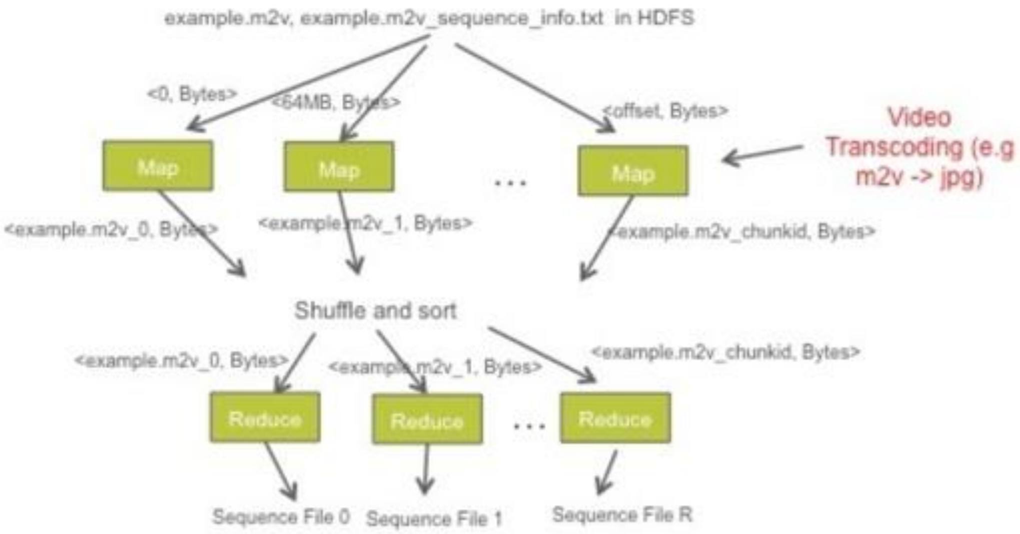
Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

4 e.g <example.m2v_1, BytesWritable>

The reducer takes in <Text, BytesWritable> pairs and writes these pairs into a sequence file. Figure 2 below illustrates the Video Decoder Map-Reduce job.




## IV. MAP-REDUCE

A Framework is for running applications on large clusters of commodity hardware. First, the implement of InputFormat that transfers original video data sets stored on HDFS. The InputFormat plays two significant roles. The first role is to provide information about the number of map tasks for the MapReduce framework in advance. Therefore, the map tasks are prescheduled in the MapReduce framework. The second role is to read a record that is transferred to map () of a map class from the original video datasets. This function is to perform by the RecordReader. The RecordReader provided from FileInputFormat is designed to read one line from a source file and pass it to map ().Next, the implementation of Mapper is to process each record received from the RecordReader. Mapper receives a video file name and byte stream of the video file as key and value pairs from RecordReader. The key and value pairs are processed by map () in a parallel and distributed manner. This transcoding processing is carried out by the Xuggler and JAI media libraries.

Map/Reduce is a programming model to process insanely large data sets, initiallyimplemented by Google. The Map and Reduce functions are pretty simple to understand.

Map(list) –> List of Key, Value

The Map function will process a data set and splits the same to multiple key/value pairs

Aggregate, Group

The Map/Reduce framework may perform operations like group,sort etc on the output of Map function. The Grouping will be done based on the Keys and the values for a given key is passed to the Reduce method

Reduce(Key, List of Values for the key) -> Another List of Key,Value

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

The Reduce method may normally perform a aggregate function (sum, average or even other complex functions) for all values for a given key.

The interesting aspect is, you can use a Map/Reduce framework like Apache Hadoop, to hierarchically parallelize Map/Reduce operations on a Big Data set. The M/R framework will take the responsibility of multiple operations, including deploying the Map and Reduce methods to multiple nodes, aggregate the output from the map methods, passing the same to the reduce method, taking care of fault tolerance in case if a node goes down etc

## V. HADOOP AND HADOOP STREAMING

Apache Hadoop is a very mature distributed computing framework that provides Map/Reduce functionality.

Hadoop streaming is a utility that comes with Apache Hadoop, and the utility allows you to create and run map/reduce jobs with any executable. Your Map and Reduce executables will read/write data from/to the console if you are using Hadoop Streaming, and Hadoop will do the required piping of data to your mapper/reducer.

From a Hadoop head node, you can run the following command to initiate your map/reduce job. Don't bother too much as later you'll see that the Hadoop dashboard we are using in Azure will provide us a decent UI to create the Map/Reduce command while we create a new Map Reduce Job.

Presently, a Developer Preview version of Hadoop is available on Windows Azure, so you can head over tohttps://www.hadooponazure.com/ and sign in to create your own Hadoop clusters, to run Map/Reduce jobs. Once you sign up, you can use the Javascript Interactive console to interact with your Hadoop cluster.

## VI. SETTING UP YOUR HADOOP CLUSTER IN AZURE

You can easily setup your Hadoop Cluster in Azure, once you are signed in to https://www.hadooponazure.com/ . Request a cluster, and give a name to the same. I'm requesting a cluster with the name Stackanalyzer, so that I can access the same @stackanalyzer.cloudapp.net for remote login etc. You need to wait few minutes till your cluster is prepared. Once the Cluster setup is done, you can access the cluster dashboard. You can access the cluster via the Javascript/Hive interactive console, via Remote Desktop etc. In this example, we'll use the Javascript console for uploading our mapper, reducer and data files to the cluster.

## VII. CONCLUSIONS

I conclude a Hadoop Map Reduce based Distributed Video Transcoding System in a cloud computing environment that transcodes various video codec formats into the MPEG-4 video format. Our system ensures uniform transcoded video quality and a fast transcoding process by applying HDFS and MapReduce, the core techniques in cloud computing enabling technologies. Moreover, our system overcomes the difficulties related to emerging & merging policies in distributed video processing as well as fault tolerance and load balancing management in large-scale distributed systems by obeying Hadoop policies. In the experiments section, i measure the total transcoding time for three sets of experiments, and experimentally verified the excellent performance of our system in video transcoding processing using distributed techniques.

## VIII. FUTURE WORKS

I can further optimize these splits, analyzing what is the optimum amount of chunks to be generated, which certainly vary according to the different data types (text, images, etc).Performance of Hadoop MapReduce jobs can be improved without increasing hardware costs, by tuning several key configuration parameters for cluster specifications, input data size and processing complexity.

Lot of research work is still going on to optimize the resources of Cloud computing based upon scheduling, elasticity and scalability. Future work includes the experiments with public Cloud and with different set of inputs.

## IX. ACKNOWLEDGEMENT

The success and final outcome of this project required a lot of guidance and assistance from many people and I am extremely fortunate to have got this all along the completion of my project work. Whatever I have done is only due to such guidance and assistance and I would not forget to thank them. I respect and thank Mr. B. SAKTHIVEL M.E., Head of the Department of Computer Science and Engineering for providing all support and guidance which made me to complete the project on time. I take this opportunity to express my profound gratitude and deep regards to my guide Mr. C.PRAKASH NARAYANAN M.E., Assistant Professor or his exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessing, help and guidance given by him time to time shall carry me a long way in the journey of life on which I am about to embark .Lastly, I thank almighty, my parents, husband and children for their constant encouragement without which this assignment would not be possible.

## REFERENCES

[1] ChenHao Chengdu and QiaoYing Chengdu ,"Research of Cloud Computing based on the Hadoop platform", 978-0-7695-4501-1/11 $26.00 © 2011 IEEEDOI 10.1109/ICCIS.2011.213.

[2] Dawei Jiang, Anthony K. H. Tung, and Gang Chen, "MAP-JOIN-REDUCE: Toward Scalable and Efficient Data Analysis on Large Clusters", IEEE, Vol.23, No.9, Sep 2011.

[3] Feng Lao, Xinggong Zhang and ZongmingGuo Institute of Computer Science & Technology, Peking University, Beijing 100871, P.R. China . "Parallelizing Video Transcoding Using Map-Reduce-Based Cloud Computing" 978-1-4673-0219-7/12/$31.00 ©2012 IEEE.

[4] Hong-Qing, Yan-jieZhai, "System Design of Cloud Computing Based on Mobile Learning", 978-1-4244-8005-0/1 0/$26.00 ©20 1 0 IEEE, Department of Educational Technology, Henan Normal University, Xinxiang.

[5] HongyuZhao, Yongqiang and Liyou Yang Network Center of Logistics Academy, "Research on Distance Education Based on Cloud Computing" 978-1-4577-0208-2/11/$26.00 ©2011 IEEE.

[6] Ishfaq Ahmad, Xiaohui Wei, Yu Sun, Student Member, and Ya-Qin Zhang "Video Transcoding: An Overview of Various Techniques and Research Issues "(2005, IEEE).

[7] Kalaiarasi.S and D.Sangeetha, Department of IT, Anna University. "Dynamic Heterogeneous data Processing in IaaS Cloud", ISBN: 978-1-4673-1601-9/12/$31.00 ©2012 IEEE.

[8] Ke-yan Liu, Tong Zhang2, Lei Wang ,HP Labs China, "A New Parallel VIDEO UNDERSTANDING AND RETRIEVAL SYSTEM", 978-1-4244-7493-6/10/$26.00 ©2010 IEEE.

[9] Myoungjin Kim, Seungho Han, Yun Cui, Hanku Lee, and Changsung 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, Nov 2012.

[10] Myoungjin Kim, Yun Cui1, Seungho Han and Hanku Lee , International Journal of Multimedia and Ubiquitous Engineering "Towards Efficient Design and Implementation of a Hadoop-based Distributed Video Transcoding System in Cloud Computing Environment " , , Vol. 8, No. 2, March, 2013.