Australian Journal of Basic and Applied Sciences, 9(16) Special 2015, Pages: 449-456

ISSN:1991-8178

Australian Journal of Basic and Applied Sciences

Journal home page: www.ajbasweb.com

Implementation of Hadoop Video Transcoding and Open stack SWIFT Video Streaming as a Service

[1]D.Kesavaraja and [2]Dr.A.Shenbagavalli

[1]*AP/CSE, Dr.Sivanthi Aditanar College of Engineering Tiruchendur, India*
[2]*Professor & Head /ECE National Engineering College Kovilpatti, India.*

ARTICLE INFO

Article history:
Received 20 January 2015
Accepted 02 April 2015
Published 20 May 2015

Keywords:
Cloud Computing Video Streaming Hadoop
Open Stack SWIFT Map Reduce

ABSTRACT

Nowadays video is being produced and consumed in more component representation formats, more device types and over a variety of networks than ever. Transmission of video through network takes more time. So Video Streaming is a very important factor when the video is moved between various heterogeneous clients in the cloud environment. This proposed cloud system has four of the video compression standards such as Low Quality Encoding, Standard Quality Encoding , High Quality Encoding and High Definition. These standards achieve better compression performance with Quality These Four standards are embedded into the Hadoop Distributed File System and Open Stack implementation where trial runs are done. Using the HDFS Map Reduce functionality, the video is splited using 64 MB blocks (Segments of Streams) and processed separately for maintaining efficiency in a time based aspect. Swift Stack is an object storage system which includes an unmodified, 100% open-source release of Open Stack Swift at the core. In addition to Swift, Swift Stack provides extensive functionality for deploying, integrating, upgrading and managing single and multi-region Swift clusters, supported by Swift Stack 24x7. This paper is concerned with the performance comparisons of Hadoop and Open stack environment in video streaming.

To Cite This Article: D.Kesavaraja and Dr.A.Shenbagavalli., Implementation of Hadoop Video Transcoding and Open stack SWIFT Video Streaming as a Service. ***Aust. J. Basic & Appl. Sci.,*** *9(16): 449-456, 2015*

INTRODUCTION

Video content is being produced, transported and consumed in more ways and devices than ever. Meanwhile a seamless interaction between video content producing, transporting and consuming devices is required. The difference in device, network and video representation types result in the necessary requirements for an unified mechanism for video content adoption. One such mechanism is called transcoding. Video transcoding is a process of converting one signal representation of a video into another. Currently transcoding is being utilized for the purposes of bit-rate reduction in order to meet network bandwidth availability, resolution reduction for display size adoption, temporal transcoding for frame rate reduction and error resilience transcoding for insuring high Quality of Service(QOS)U. Erez et.al(2007)Web site et.al(2009).

Video Transcoder is a piece of software that decodes a given signal representation of a video and converts into another. The following figure shows the general task of the Video Transcoder

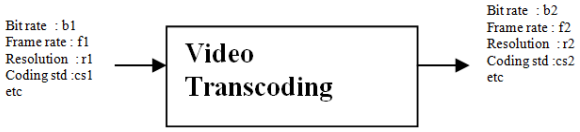

Fig. 1: Video Transcoding Model.

The above Figure illustrates the main idea behind transcoding and the common video transcoding operations. The task of a video transcoder could range from a simple task of converting the video format of the entire stream with a different codec (standard). The following four codes are considered for this proposed work.

- Low Quality – H263/MP3 (flv)
- Standard Quality – H264/AAC (mp4) 512 x 288
- High Quality – H264/AAC (mp4) 848 x 480

Corresponding Author: D.Kesavaraja, AP/CSE , Dr.Sivanthi Aditanar College of Engineering Tiruchendur , India, Tel: +91 9865213214; E-mail : dkesavaraja@gmail.com

• High Definition – H264/ACC (mp4) 1920 x 1080

These four methods have been chosen for its efficiency in compressing the video without the loss of vital data.

Video compression is an extension of image compression which also utilizes the temporal redundancy that exists among consecutive frames in addition to image level compression. Instead of only encoding individual frames independently, one can first predict a frame based on the previous frame and code only the difference in this prediction. Therefore to minimize the error and be able to get a better compression require an efficient way of prediction. When video content delivery becomes the prime important factor for consideration, then infrastructural investment increases exponentially Ali.et.al(2010)Anustup et.al(2012)Web siteet.al(2006). The major problem occurs, when time is a constraint and the video has to be presented to the user. Due to restriction of time, this might lead to poor Coding and encoding Cong et.al(2012). Again pre-processing infrastructure increases the cost and provides little flexibility. When there is a shift in the ratio of video flow into the stream, it increases the criticality of the structure. Cloud computing that shares resources as services to the user comes as a boon in such times of difficultyDidier et.al(1991)R. Gallager et.al(1962)Web site et.al(1962)Web site et.al(1962). The resources can be used in various metered way of access like hour based or day based and also is scalable as per the requirements of the userAdriana et.al(2010). The user need not worry about fine tuning the resource since that responsibility will be taken by the resource provider, thereby just making the user utilize the service providedD.Kesavaraja .et.al(2013)DanielDíazSanchez .et.al(2011)Daniel .et.al(2011).

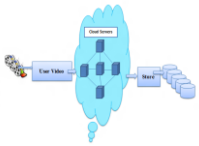

Fig. 2: General Cloud Video Architecture.

Figure 2 describes general architecture of Cloud Servers storing the video data into Database for future use.

This kind of service is called as Infrastructure as a service(IaaS). The resources are obtained by running a large number of virtual machines and forming a cluster that is called as the virtual clusterDominique .et.al(2013)Hai Zhong .et.al(2010). These virtual machines run in physical hardware.

Workload balancing is a methodology used in cloud for maintaining high level resource utilization. The resources are migrated in order to achieve better efficiency using live migration concept, in such a way that it is transparent and non -interruptible to the user.

Hadoop distributed file system provides a scalable, portable and distributed Infrastructure , which helps in performing the task of storing video split simultaneously across multiple nodes and Map Reduce helps in the completion of Coding process parallel Dean .et.al(2011)Dominique .et.al(2013)Web site .et.al(2010).

2. Video Block Split-up scheme:

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 video file and is splited into 64MB chunks, each Mapper process can access the streams in each split independently. However, when the input file is video file (bit stream) and is split into many streams, each Mapper process needs to interpret the bit stream 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 bit stream can be transcoded into a sequence of JPEG images that can be subsequently processed by video analytics Map Reduce jobs

Algorithm 1 : Pseudo for Agreement Splitter
User Upload the Video
Find the Video Size (S)
While S > 0 do
If S>64 MB Then
Split Video in to 64 MB Blocks
For i=0 ; i<abs(S/64)+1 ; i++
For each Blocki
Calculate Histogram of Last Frame
Find the Histogram for 2MB Backward Direction of Blocki
Calculate Histogram Difference
If HistDiff>UpperTheshold
Add it to Blocki+1
Remove the Frames from Blocki
Else
Continue
End If
Find the Histogram for 2MB Forward Direction of Blocki+1
Calculate Histogram Difference
If HistDiff>UpperTheshold

Add the Frames to Blocki
Remove it from Blocki+1
Else
Continue
End If
NearBlock[i]=Blocki
Else
NearBlock[0]=Video
End if
End While

The pseudo code above explains the method of implementation for the proposed system. The video is first measured for its size. If in case the size of the video, is less than 64MB, then the video is transmitted without any problem. If the size of the video exceeds the prescribed size, then the video is splited into 64MB blocks. It is then passed for finding the histogram for the last frame. In this part of the work we use two methods for finding the histogram values. Both the forward and backward methods for finding the histograms are used to obtain the approximation. If the histogram is above the threshold value depending upon the method of the histogram, the frame is either added or deleted from the block. This produces blocks of size nearer to 64MB. These blocks are then transmitted.

3.Video Transcoding:

Digitally recorded videos are categorized to be captured by video cameras and camcorders. We have decided to compress the videos. In order to store the videos we need to compress them, since Coding reduces a lot of complexities in storageRafael Pereiraet.al(2010). The encoding process also takes a lot of time for getting completed, thereby increasing the production timeRafael Silva Pereira.al(2013). These problems are overcomed by using the Infrastructure as the service component of cloud computing, which provides on demand scalable serviceJinchang Ren .et.al(2011)Web site..et.al(2011)Naeem Ramzan et.al(2010).

Some modification have been made to make FFMPEG work with the Video framework and the architecture proposed. The above figure shows the activity diagram of the original FFMPEG transcoder along with the changes applied to the transcoder.

First part of figure 3 illustrates the main operations that the transcoder implementation (FFMPEG) goes through when dealing with audio and video streams. The main idea is to segment the format header for the stream metadata and to find an encoder from the libavcodec library to deal with the encoding.

The second part of the figure illustrates the modification made in-order to utilize the original transcoder in the context of the distributed one. Since the cloud transcoder uses segmented streams as an atomic unit of data distribution among multiple nodes, each transcoding instance (process) needs to wait until there is no transcoding tasks (no segmentation) left to be done.

The video is encoded by being splited into various segments and is then distributed over the Cloud environment. A major issue that needs to be considered is the synchronization of the video splits while merging Navdeep et.al(2012). To observe the time required for the breaking up of video, the process is done using two methods. The video files are encoded in a specific format by the Map Reduce framework for delivering the content to the user efficiently.

In this proposed work we use Histogram Based Video Shot Detection. Shot based video detection helps in reducing the length of the video into different shots. It detects number of shots by using histogram differencing and using local and global thresholds. In this proposed work we use FFMPEG Video Coding.

4 .Hadoop Distributed File System:

Algorithm 2:Video Map Operation
beginmap(String key, String value):
// key: Video name
// value: Video Header and contents
Split the Video into 64 MB Blocks (Nsh)
for each blocks in Nsh in value:
CompressIntermediate(s, "format");
end
Algorithm 3:Video Reduce Operation
begin reduce(String key, Iterator values):
// key: a shot
// values: a list of Compressed Blocks CNsh
Video v1;
for each v1 in Compressed Blocks CNsh
Emit(AsMergedVideo(v1));
Store to Cloud Database
end

The above algorithm describes the basic activity of Hadoop distributed video encoding procedure.

Figure 4 describes the basic Cloud video Coding Architecture using Map reduce frame work . In the first phase we perform video shot detection and the Mapping of Video Transcoding using four standards, finally in the reduce phase the compressed video files are merged and we receive the final compressed video in Quimper fine Vasileios et.al(2009)D. Dardari et.al(2004). The compressed video is perfectly stored in cloud database

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

Video Decoder Map Reduce Job: This Map Reduce 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.

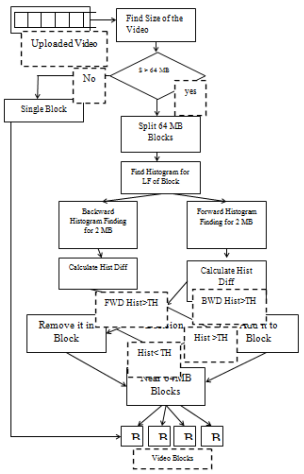

Fig. 3: Activity Diagram of Video Encoding System.

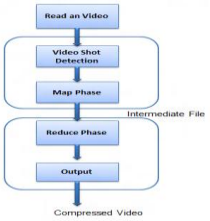

Fig. 4: Flow graph for Hadoop Video Coding.

5. HDFS Video Transcoding:

`The normal coding methods take a single video and compress them using the processing power that remains with them. This methodology increases the downtime and thereby affects the efficiency of video based services. In order to overcome this problem we propose a new strategy wherein the video is splited on shot based detection method and then using the power of Hadoop Distributed file system, we process the splits by using the concepts of Map and Reduce by implementing it inside the HDFS. This in turn reduces the time taken for the entire process thereby increasing the efficiency Rajkumar Buyya.al (2011) Rakesh Kumar Jha et.al (2011) Shivnath Babu et.al (2010) Smitha Sundareswaran et.al (2012).

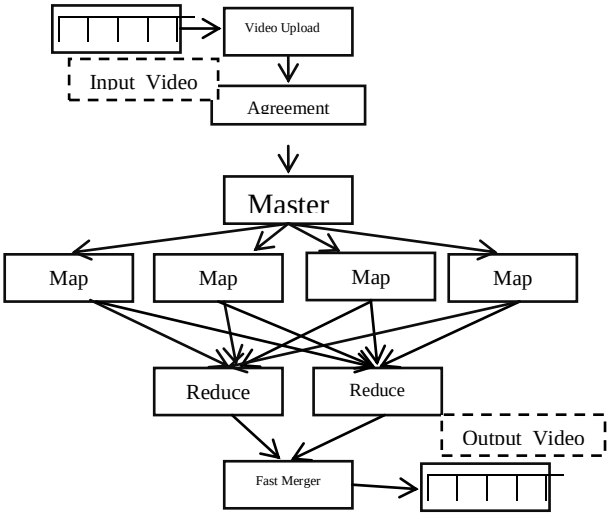

Fig. 5: Proposed Work Architecture of HDFS Video Transcoding.

6. Open Stack SWIFT:

Swift Stack is an object storage system which includes an unmodified, 100% open-source release of Open Stack Swift at the core. In addition to Swift, Swift Stack provides extensive functionality for deploying, integrating, upgrading and managing single and multi-region Swift clusters, supported by

Swift Stack 24x7Web site et.al(1962),Web site et.al(1962).

Objects and files are written to multiple drives, and the Swift software ensures the data is replicated across a server cluster. By default, Swift places three copies of every object in as unique-as-possible locations -- first by region, then by zone, server and drive. If a server or hard drive fails, Open Stack Object Storage replicates its content from active nodes to new locations in the cluster.

The system, which is accessed through a REST HTTP application programming interface (API), can scale horizontally to store petabytes of data through the addition of nodes, which typically equate to servers. Open Stack Swift software is based on Cloud Files technology developed by Rackspace Hosting Inc. Rackspace and NASA initiated the project and co-founded the community that develops and maintains OpenStack software, which includes compute, storage and networking components for building cloud computing services.

Digitally recorded videos are categorized to be captured by video cameras and camcorders. We have decided to compress the videos. In order to store the videos we need to compress them, since Coding reduces a lot of complexities in storageD. Dardari et.al(2004). The encoding process also takes a lot of time for getting completed, thereby increasing the production timeS.N. Diggavi et.al(2008). These problems overcome by using the Infrastructure as the service component of cloud computing, which provides on demand scalable serviceW.H.R. Equitz et.al(1991)U. Erez et.al(2007)Web site et.al(2009)Web site et.al(1997).

Some modification have been made to make FFMPEG work with the Video framework and the architecture proposed. The above figure shows the activity diagram of the original FFMPEG transcoder along with the changes applied to the transcoderDidier et.al(1991),R. Gallager et.al(1962).

Figure 8 shows the video Streaming mechanism that is followed in the proposed system. In case the streaming quality is higher, then it require the encoding and strategy for this situation is decided. Also smoothness as required is introduced in to the system.

7. Video Quality Measurement(PSNR):

The Average Peak Signal to Noise Ratio is often used to measure the quality of video and it is calculated as in the following equation. Also compression performance is evaluated by measuring the visual quality of reconstructed image in terms of PSNR against various compression ratio. The PSNR is used as objective quality measure. The phrase Peak Signal-to-Noise Ratio, often abbreviated PSNR is used for the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. PSNR is usually expressed in terms of the logarithmic decibel scale. The PSNR is most commonly used as a measure of quality of reconstruction of lossy compression codecs (e.g., for image compression). The signal in this case is the original data, and the noise is the error introduced by compression. When comparing compression codecs, it is used as an approximation to human perception of reconstruction quality, therefore in some cases one reconstruction may appear to be closer to the original than another, even though it has a lower PSNR (a higher PSNR would normally indicate that the reconstruction is of higher quality). It is most easily defined via the mean squared error (MSE) which for two m×n monochrome images I and K where one of the images is considered a noisy approximation of the other and is defined as:

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left[I(i,j) - K(i,j)\right]^2 \quad (1)$$

The PSNR is defined as:

$$PSNR = 10.\log_{10}\left(\frac{MAX_I^2}{MSE}\right) = 20.\log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right) \quad (2)$$

Here, is the maximum possible pixel value of the image. When the pixels are represented using 8 bits per sample, this is 255. Typical values for the loss in lossy image and video compression are between 30 and 50 dB where higher is better. Acceptable values for wireless transmission quality loss are considered to be about 20 dB to 25 dB.

8. Performance measure:

Measuring the performance of the system is a very important task to measure the effectiveness of the proposed strategy.

Taking into consideration that the performance evaluation should be done in multiple perspectives, we have compared the functionality with videos that represent different sizes. Also a minimal comparison of the process with the normal method that is being used is done.

Considering many scenarios in the process is a good practice. we have considered implementing the process in varying number of Virtual machines, so that the effect of the same input when high scalability is available, can be visualized.

Table2 shows the comparison of the file Coding process when VMs of 4, 8, and 16 is used comparing the performance with that of the normal method.

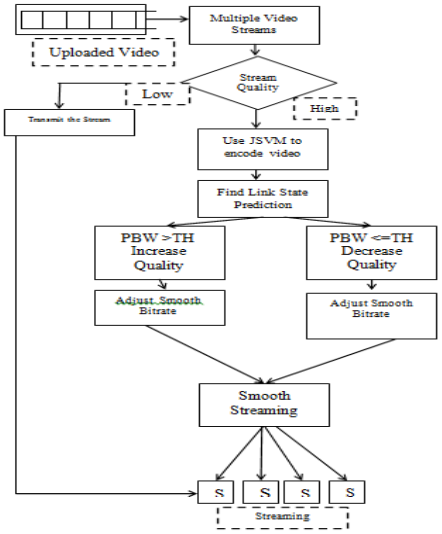

Fig. 6: Activity Diagram of Video Streaming System.

Table I: Video Transcoding.

Input Video File	Input Video File Size	Compressed Video Size
Video1.avi	723MB	249 MB
Video2.avi	1.34GB	430 MB
Video3.avi	20GB	7.2 GB
Video4.avi	84GB	32.12 GB
Video5.av.i	127GB	41.23 GB
Video6.avi	240GB	71.23 GB

Table II: Video coding in different number of virtual machines.

Input Video File	Input Video File Size	Time			
		Coding Using Normal Method	Coding Using Hadoop 4 VM	Coding Using Hadoop 8 VM	Coding Using Hadoop 16 VM
Video1.avi	723MB	18 min 32 sec	12 Min 12 sec	11 Min 21 sec	6 min 12 sec
Video2.avi	1.34GB	22 min 12 sec	13 min 32 sec	9 min 09 sec	4 Min 13 sec
Video3.avi	20GB	30 min 13 sec	15mi 12 sec	11 min 7 sec	7 min 16 sec
Video4.avi	84GB	45 min 14 sec	23 min 22 sec	15 min 31 sec	9 min 11 sec
Video5.avi	127GB	62 min 12 sec	31 min 30 sec	24 min 11 sec	11 min 11 sec
Video6.avi	240GB	74 min 52 sec	40 min 31 sec	32 min 22 sec	12 min 31 sec

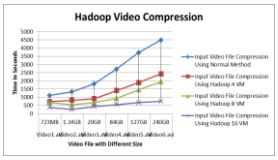

Fig. 7: Hadoop Video Coding Ease of Use.

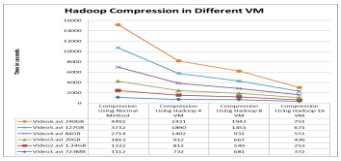

Fig. 8: HadoopCoding in Different Number of Virtual Machines.

Figure 7 and 8 shows the graphical view of the process with the different sizes of video files and with varying number of VMs.

The Following Figure 9 illustrates the quality of the transcoded output.

Figure shows that the quality of the transcoded video and their PSNR between 34 dB to 46 dB . It shows that if bit rate and PSNR are directly propositional to each other ie, if bit rate increases PSNR also increases vice versa , where higher PSNR is better

Cloud environments are useful in their own right , however, when they are joined together, the benefits an service provider experiences are

substantial. Although the environment will be complex, an enterprise will see substantial synergies by joining an OpenStack private cloud with an Apache Hadoop environmentWeb site et.al(1962).

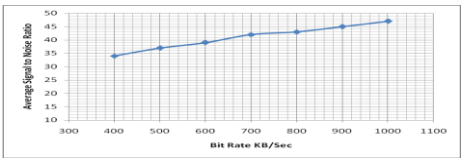

Fig. 9: Average Signal to Noise Ration Graph.

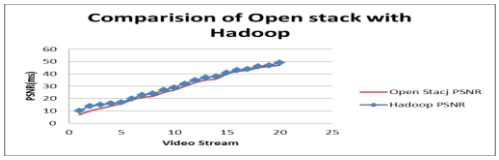

Fig. 10: Comparision of Openstack with Hadoop.

This deployment model is best used in an organization looking to pilot a private cloud through a storage pool while concurrently using in-house big data technologies. Best practices dictate that you deploy your big data technologies to your production data warehouse environment first, then build and configure your private cloud storage solution. When you have your Apache Hadoop MapReduce technologies successfully joined to your data warehouse environment and your private cloud storage pool is built and working correctly, then you can integrate the private cloud storage data with the scheduled Hadoop MapReduce environment.

Conclusion:

Currently the advances in cloud computing has provided us with new ways of utilizing computational resources. Though there are unlimited resources in a cloud based service scenario, the resources cannot be infinite. There has to be a limit to which the extent of the elasticity property can be utilized. This can be solved by optimization of the chunks generated to be limited to an extent that restricts based on the Nodes available in the cloud source. The performance of the clusters in a Hadoop Map reduce does not depend on the hardware it works upon. The performance of the Hadoop map reduce shall be increased ,Some of the parameters that can be fine-tuned are cluster specifications and processing complexity. Using the HDFS Map Reduce functionality, the video is splited using 64 MB blocks (Segments of Streams) and processed separately for maintaining efficiency in a time based aspect. This system helps in reducing the size of the video slice thereby providing opportunity for efficient Streaming in quicker time using Openstack SWIFT

REFERENCES

Adriana Garcia Kunzel, HariKalva, BorkoFurht, 2010 “A Study of Transcoding on Cloud Environments for Video Content Delivery” 978-1-4503-0168/10, 13-18, ACM

Ali, C., Begen, TankutAkgul and Mark Baugher, 2010. ” Watching Video over the Web Part 1: Streaming Protocols”.

AnustupChoudhury and Gerard Medioni, 2012 “A Framework for Robust Online Video Contrast Enhancement Using Modularity Optimization”, IEEE Transactions on Circuits and Systems for Video Technology, 22-9.

Web site, 2013, ApacheHadoop http://hadoop.apache.org/mapreduce/

Cong Wang, Ning Cao, KuiRen and Wenjing Lou, 2012” Enabling Secure and Efficient Ranked Keyword Search over Outsourced Cloud Data”, IEEE Transactions On Parallel And Distributed Systems, 23-8.

Web site, 2006 Converting video formats with FFMpeg, Linux Journal archive- Issue 146, June, pp. 10

Kesavaraja, D. and Dr.A.Shenbagavalli, 2013. Cloud Video as a Service [VaaS] with Storage , Streaming , Security and Quality of service : Approaches and Directions, International Conference on Circuits, Power and Computing Technologies [ICCPCT-2013]

DanielDíazSanchez, , FlorinaAlmenarez, Andrés Marin.,2011 DavideProserpio, and Patricia Arias Cabarcos, “Media Cloud: An Open Cloud Computing Middleware for Content Management” IEEE Transactions on Consumer Electronics, 57-2.

Daniel Gmach, LudmilaCherkasova, 2011. HP Labs, Palo Alto, CA (USA) and Jerry Rolia HP Labs, Bristol (UK) “Resource and Virtualization Costs up in the Cloud: Models and Design Choices”, 978-1-4244-9233-6/11, IEEE.

Dean, J., Ghemawat, S. MapReduce, 2004. Simplified Data Processing on Large Clusters. In OSDI

Dominique, A., Heger, 2013. “Optimized Resource Allocation & Task Scheduling Challenges in Cloud Computing Environments”, dheger@dhtusa.com

Hai Zhong, Kun Tao, Xuejie Zhang, 2010. “An Approach to Optimized Resource Scheduling

Algorithm for Open-source Cloud Systems", 978-0-7695-4106-8/10, 124-129, IEEE.

Hsiao-Ying Lin and Wen-GueyTzeng, 2012." A Secure Erasure Code-Based Cloud Storage System with Secure Data Forwarding", IEEE Transactions on Parallel and Distributed Systems, 23-6.

Web site, 2014. http://www.sauronsoftware.it/projects/jave/

Hui Kang, Yao Chen, Jennifer L. Wong, 2011."Enhancement of Xen's Scheduler for MapReduce Workloads" 978-1-4503-0552-5/11/06, 8-11, ACM.

Jeffrey Dean and Sanjay Ghemawat, 2010. "MapReduce: Simplified Data Processing on Large Clusters"

Jiann-Liang Chenz, SzuLin Wuy, YanuariusTeofilusLarosa, PeiJia Yang and Yang Fang Li, 2011. " IMS Cloud Computing Architecture for High Quality Multimedia Applications", 978-1-4577-9538, page 1463-1468, IEEE.

Jinchang Ren, Jianmin Jiang, and Juan Chen, 2009." Shot Boundary Detection in MPEG Videos using Local and Global Indicators, IEEE Transactions on Circuits and Systems for Video Technology, 19-8.

Web site, 2013. Mencoder – http://www.mplayerhq.hu

Naeem Ramzan, EmanueleQuacchio, Toni Zgaljic and Stefano Asioli, Luca Celetto, EbroulIzquierdo, FabrizioRovati, 2010. "Peer-to-Peer Streaming of Scalable Video in Future Internet Applications"

Navdeep, S., Chahal and Baljit S. Khehra, 2012. A Comparative Study for Optimization of Video File Coding in Cloud Environment , International Journal of Computer Applications (0975 – 8887), 60-13.

ParthaPratimMohanta, Sanjoy Kumar Saha, 2012. "Model-Based Shot Boundary Detection Technique Using Frame Transition Parameters", IEEE Transactions on Multimedia, 14-1.

Pengye Xia, S.H., Gary Chan, 2011." Optimal Bandwidth Assignment for Multiple-Description-Coded Video", IEEE Transactions on Multimedia, 13-2.

Rafael Pereira, Karin Breitman, 2010. " An Architecture for Distributed High Performance Video Processing in the Cloud", 3rd International Conference on Cloud Computing, page, 482-489, IEEE

Rafael Silva Pereira, Karin, K. Breitman, 2013. "Video processing in the Cloud" SpringerBriefs in Computer Science, ISBN: 978-1-4471-2136-7.

RajkumarBuyya, James Broberg, AndrzejGoscinski, 2011. "Cloud Computing, Principles and Paradigms, Wiley, ISBN: 978-0-470-88799-8, page 123-151.

Rakesh Kumar Jha, Upena, D. Dalal, 2011. "On Demand Cloud Computing Performance Analysis With Low Cost For QoS Application", International Conference on Multimedia, IMPACT-2011, 978-1-4577, IEEE

ShivnathBabu, 2010. Towards automatic optimization of MapReduce programs, In: the 1st ACM symposium on Cloud computing, pp: 137-142. ACM Press, New York.

SmithaSundareswaran, Anna C. Squicciarini and Dan Lin, 2012." Ensuring Distributed Accountability for Data Sharing in the Cloud", IEEE Transactions on Dependable and Secure Computing, 9-4.

Vasileios, T., Chasanis, Aristidis C. Likas and Nikolaos P. Galatsanos, 2009. " Scene Detection in Videos Using ShotClustering and Sequence Alignment", IEEE Transactions On Multimedia, 11-1.

Venkatesa Kumar, V., S. Palaniswami "A Dynamic Resource Allocation Method for Parallel Data Processing in Cloud Computing" ISSN 1549-3636, page 780-788, Journal of Computer Science, 8(5).

Dardari, D., M.G. Martini, M. Mazzotti and M. Chiani, 2004. Layered video transmission on adaptive ofdm wireless systems. EURASIP J. Appl. Signal Process.,1557-1567.

Diggavi, S.N., A.R. Calderbank, S. Dusad and N. Al-Dhahir, 2008. Diversity embedded space-time codes. IEEE Transactions on Information Theory, 54.

Equitz, W.H.R. and T.M. Cover, 1991. Successive refinement of information. Information Theory, IEEE Transactions on, 37(2): 269-275.

Erez, U., M.D. Trott and G.W. Wornell, 2007. Rate less coding for Gaussian channels. Arxiv preprint arXiv:0708.2575.

Web site., ETSI. Digital Video Streaming, 2009. Framing structure, channel coding and modulation for digital terrestrial television, EN, 300-744.

Web site.,1997,IEEE/ACM Transactions on Networking (TON), 5(6):784-803.

Didier Le Gall, 1991. MPEG: a video compression standard for multimedia applica¬tions. Commun. ACM, 34(4): 46-58.

Gallager, R., 1962. Low-density parity-check codes. Information Theory, IRE Transactions on, 8(1):21—28,

Web site., 2012. https://swiftstack.com/openstack-swift/

Web site., 2012 .https:/ /www. openstack.org/summit/san-diego-2012/ openstack-summit-sessions/ presentation/ openstack-swift-introduction n-architecture-overview-and-use-cases

Web site., 2012. http://www.ibm.com/developerworks/cloud/library/cl-openstack-deployhadoop/