VOL. 10, NO. 5, MARCH 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

SEASONS: A SCALABLE 4-TIER CLOUD ARCHITECTURE FOR HANDLING HETEROGENEITY OF MOBILE DEVICES IN LIVE MULTIMEDIA STREAMING

Preetha Evangeline D. and Anandhakumar P.
Department of Computer Technology, MIT Campus, Anna University, Chennai, India
E-Mail: preethadavid4@gmail.com

ABSTRACT

Cloud based live Media streaming is becoming one of the prominent scope among researchers. The growing popularity of the cloud has moved the Multimedia technology from being traditional to be highly modernized. Mobile devices have replaced the traditional way of viewing videos through PC's. According to the recent survey there is an exponential hike in mobile video consumption, approximately 90% of consumers use a mobile devices to watch online videos when compared with 61% in 2012. Dealing with Device heterogeneity serves to be one of the biggest challenges for live media streaming as Mobile devices comes with various resolutions, OS's, audio and video codec. It is hard to adapt live streaming contents according to the specification of the device on the fly. This paper proposes SEASONS, a 4-Tier Novel architecture that handles Device heterogeneity and concentrates on providing efficient and scalable media dissemination system. The proposed system takes 9 sec on an average for transcoding on the fly and 14 sec on an average for end-to-end delivery of videos when heterogeneous devices are considered.

Keywords: cloud architecture, live multimedia streaming, device heterogeneity.

1. INTRODUCTION

Internet technologies have greatly advanced and the emergence of Cloud which provided better resource availability (processing power, storage, network bandwidth), accessibility from anywhere in the world has led users to move to the cloud era which has popularized live streaming widely. Live streaming enhanced with Cloud computing provides rich Media experience. The rapid development in social media applications has led to an enormous technical advancement of Mobile devices. The world is becoming busy and people are interested in using smart devices.

The traditional surfing through Internet has become obsolete and in recent days these kinds of activities (social network, video conferencing, live streaming and gaming) are accessed through mobile devices. According to recent survey nearly 35 mobile OS's, 130 screen resolutions and 90+ brands of mobile phones have been evolved and will exceed the count in the near future. Hence the heterogeneity of mobile devices is increasing at a vast rate. The video chunks are streamed live to the mobile devices and before it is delivered to the device it has to be transcoded which best suits the user device.

The case with video-on-demand is different where videos are being transcoded to more possible versions and then streamed to the mobile device. Another significance of VoD is the popularity hit of the frequently viewed and requested video. Videos that are requested frequently for viewing gets popular and those videos can be pre transcoded into various possible formats and can be readily streamed when requested by various heterogeneous devices. Whereas in the case of live streaming the scenario is 'Operation on the Fly'. This scenario has paved way for migrating to Cloud environment where the computational power, storage, bandwidth is enormously high. Here the live content is adapted and streamed according to the specification of the mobile devices handling the heterogeneity.

In traditional transcoding the content adaptation process is either done in the Client-side or Server-side. During Client -side transcoding, the device run out of storage space while storing the downloaded video for conversion and then the transcoded video. In Server - side transcoding the server gets overloaded with the content adaptation process while accommodating the videos into different formats. This is another one reason why transcoding is migrated to Cloud.

2. RELATED WORKS

Analysis of internet streaming service to mobile device (Yao Liu *et al*., 2013) specifies transcoding in the server side where videos can be converted into 40 different versions and streamed to the devices. This paper concluded that there exists a great demand of CPU resources when transcoding takes place online. Better replacement policies could be adopted for enabling optimized mobile streaming system.

Cloud assisted SVC (Zuqing *et al*., 2013) was proposed to handle the heterogeneity of the network. Here Cloud collects both the video resources and the network status information. The proposed Scalable Video Coding system encodes the video streams and stores it in multiple servers. The video server on receiving the request for streams forwards the query to the cloud. Here the cloud computed multi-path strategy and provides labels for individual streams. The limitation of this work is, every time the network status being advertised may lead to Overhead.

VOL. 10, NO. 5, MARCH 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

CMTS (Seungho *et al.*, 2014) Cloud Multimedia Transcoding System which overcomes the limitations of Cloud Multimedia Data Conversion System (CMCS), such as performance delay and transcoding failure. Transcoding failure tends to occur when large volume of video files could not be converted. In the proposed CMTS, Hadoop Distributed File System and Map Reduce technology is used. Here the video files are created through various heterogeneous device and they are transferred to CMTS through web interface. The files are divided and stored according to HDFS policies. Predefined format and size that are specified by the user are accounted and the video files are transcoded. The advantage of this system is parallel processing that takes place on the divided streams. The limitation of the work is there isn't any method discussed at the time of Churn failure.

GMSME (Garnet Message System Micro Edition) (Sangmi Lee *et al.*, 2003) which was actually developed to integrate Mobile devices into Garnet collaborative System which is a universal accessible collaborative environment. This paper handles Application aware transcoding. Another challenge mentioned in this paper is reducing the size of the file over wireless network. To handle this issue, the Master of the session initiates the change in the collaborative feature; hence all the Mobile users need not fix their new features and update it again. The underlying transcoding operations are performed based on the changes initiated by the Master. Here transcoding is carried out in proxy based middleware which can be a limitation for large scale peers and hence Scalability issue may occur. Even for converting a file to a comparatively smaller size, the complete file has to be downloaded and converted which causes significance overhead on the network of proxy middleware.

3. PROPOSED WORK

This paper proposes a 4-Tier Cloud based architecture illustrated in Figure-1.

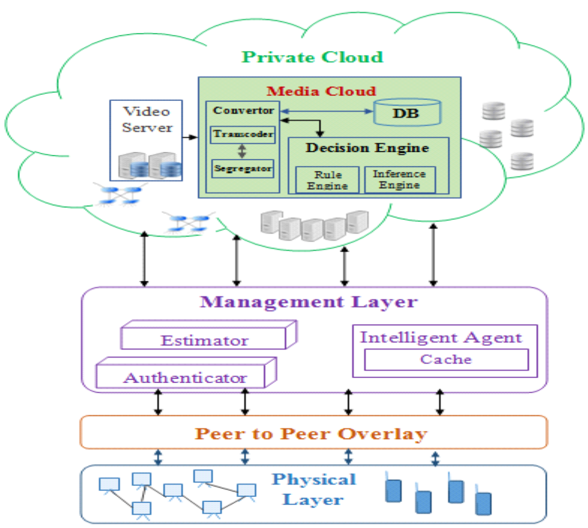

Figure-1. Seasons: 4-Tier architecture.

It consists of (Bottom-up approach) the Physical layer (PC's, laptops, Mobile devices) organized in a peer to peer fashion. The second layer is the p2p dynamic overlay of the underlying physical layer. The third layer is the Management layer which serves as the interface between the upper cloud layer and the lower p2p layer. There comes the topmost layer which is a private cloud having an additional crowning of Media Cloud. The overall objective of the proposed work is (i) to handle the heterogeneity of mobile devices on the fly for live video streaming, (ii) to minimize the end-to-end delay and (iii) to improve the playback continuity of the streaming video.

3.1 Physical layer

The physical layer consists of end users who are interested in viewing the video content that is being streamed. As peer-to-peer network architecture seems to be highly scalable for large scale network, it is used for organizing the peers and moreover communication and content distribution is efficient between Multi-users in case of peer-to-peer network. This layer connects Heterogeneous mobile devices, laptops, tablets and PC's which are used by the users for viewing multimedia content. Here the devices may join and leave the network anytime and keeping track of the participating node seems to be a challenging task.

3.2 P2P overlay

Overlay is a logical or a Virtual network that is constructed on top of the underlying physical network. In this proposed architecture the reason for building the overlay network is due to the dynamic nature of the participating nodes. With the nodes that are actively online in the network the overlay is constructed. An overlay may be a structured or an unstructured one which can handle the joining and leaving of nodes anytime in a network. As there are many works done using an unstructured overlay, this proposed work uses a structured overlay in which a Hybrid self- balancing tree is constructed which handles the dynamic load of the nodes.

3.3 Management layer

The management layer serves as the intermediate between the bottom and the top layer. This layer consists of three modules (i) Estimator, (ii) Authenticator and (iii) Intelligent agent.

Estimator

The role of the estimator is to calculate the capacity of the overall peer and estimate the amount of required resources which is to be obtained from the cloud for better data dissemination in order to reduce the end-to-end delay and improve the playback continuity of the video streams.

Authenticator

The authenticator authenticates the incoming node when newly joining the network. The node has to provide its individual identity in order to participate in the session. The authenticator keeps track of the online duration time of a node and how frequently the node has requested to watch the session.

Intelligent agent

The intelligent agent communicates with the user device gathering the heterogeneous information and specification of the system such as screen resolution, OS's and audio/video codec formats. Hence this intelligent agent maintains the list of all specifications and forwards the request along with the specification details to the above Cloud layer. It not only keeps track of the incoming device specification but it maintains the transcoded format of the output videos to every individual specification. The intelligent agent maintains a cache which consists of frequent system specifications and its matching transcoded output format.

In future incase if the incoming request specification matches with the existing information in the cache directory, the intelligent agent sends a clue message to the cloud about the format the video should be transcoded. This makes the complex transcoding operations much easier thereby reducing the delay of conversion on the fly. The steps to calculate and keep the cache directory updated with frequently requested streaming format is given below.

Input: R_1, R_2, R_3, $R_{4,\ldots\ldots}$ R_K are the various available streaming formats.

Output: The Rearrangement of Streaming formats in most frequently requested order.

For each R= 1, 2, 3, ... k do the following

Step-1. Find the number of request '**nr**' processed for the current streaming format.

Step-2. Find the weight '**w**' of current streaming format which varies between the ranges 0.1 to 1 based on its compatibility
with other specification devices.

Step-3. Rank is calculated by multiplying the number of request and weight of the current format.

V = nr * w

Step-4. Rearrange the formats such that $V(R_1)>V(R_2)>V(R_3)>V(R_{4.})\ldots\ldots.V(R_k)$

3.4 Private cloud layer

The topmost layer in the proposed 4-Tier architecture is the Private Cloud layer. As the paper concentrates on live streaming for critical applications such as e-learning, telemedicine, and video conferencing the streaming rate and Quality of Service should be highly maintained. Nowadays the trend in multimedia streaming has become cloud assisted. This private cloud is set to provide resources such as Computational nodes and Storage nodes. The underlying physical layer is restricted with limited bandwidth and fluctuations in the network may lead to interrupted multimedia streaming. There comes the cloud enriching the underlying network with more computational power and storage for the backup in case of churn failure. Not only for providing storage and computational resources, the proposed private cloud is enhanced with the media cloud which takes care of the transcoding part online on the fly.

Media cloud

Media cloud is termed to the outstanding highlight of the proposed work. The scope of bringing media cloud embedded into the private cloud is to transcode the incoming live videos from the video server according to the incoming system specification. This is quite a critical challenge in the case of live streaming, since the delay parameter is to be considered. If transcoding is time consuming then obviously dissemination of video streams will not maintain its continuity. Hence content adaptation is carried out inside the cloud for faster conversion process.

The media cloud maintains the database of all available screen resolutions, audio codec, video codec and adaptive streaming output formats. Once the request is forwarded to the media cloud, the incoming video chunk is converted based on the device specification. The converter module consists of two operations such as Segregation and transcoding. Here the streaming video is converted to multi bitrate renditions of various sizes and qualities. The segregator divides the video streams and analyzes which codec is used. The converter in parallel discuss with the decision engine which operates based on rule and inference based. The transcoder functions as an adaptation engine which reformats the video contents to cater the capabilities of the device that is being used for viewing the content.

Rule engine

The Rule Engine maintains a set of adaptation rules based on the device specification, display specification like size, pixel, etc., and user preferences. The rule that is framed is used to make appropriate decisions based on various aspects. After the segregator breaks the video into chunks, the rules are being implemented and the output characteristics of the resulting content is determined based on modality and appropriate adaptation is executed.

Inference engine

When a new device with varying specification sends the query, the request is transmitted to the inference engine which makes use of the existing adaptation rules, maps the appropriate format and converts the incoming chunks intelligently. And the output format which has been newly evolved is saved for future references.

4. EXPERIMENTAL SETUP

In our experimental set up, we have used 60 Acer Veriton X4630G personal computer each with the configuration of 64 bit Windows 7 operating system architecture, i5 processor, 4GB RAM and 500 GB of hard drive capacity, one Dell precision 690 workstation of specification Quad-core Intel Xeon 5300 series processor with upto 1333 MHz front side bus and 2x 4GB shared cache, upto 32GB, Mobile phones of brands such as Nokia

VOL. 10, NO. 5, MARCH 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

with the screen resolution of 480x800 and 240x320, Blackberry with 320x240 and Micromax with 480x854 along with various specifications were used as heterogeneous devices. These devices were connected as peer-to-peer with physical network strength of 100Mbps. Some devices were switched off and switched on over the fly in order to maintain the dynamic nature. With the devices that are active, a tree based overlay was constructed which was evaluated by joining and removing the nodes from the network.

4.1 Private cloud setup

Private cloud set up is done using Openstack. The reason behind openstack is massive scalability and this can be used to develop a private as well as public cloud. We set up openstack using juju and maas in ubuntu 14.04 and nodes were added in cluster. We set the streaming rate to be 500kbps using http as the streaming protocol. Both Helix server and VLC were used to stream videos and the video chunks were divided into 20Kb were each chunk corresponds to 0.03 seconds of streaming rate. In our experiment two scenarios were considered: (i) Joining and leaving of nodes and (ii) Different size of chunk.

5. RESULT AND DISCUSSIONS

The experimental results are tabulated below. Here the streaming rate of the video is maintained constant for all the devices with varying specifications. Graphs were plotted for the time taken for transcoding and the overall end-to-end delay for all the devices. Figure-2 shows the result for transcoding time as well as the time taken for the complete delivery of streams to the devices. Figure-3.a shows the resulting graph of the varying transcoding time taken by Heterogeneous devices and homogeneous devices. Same devices with equal configuration were streamed and the results remain to be constant whereas a heterogeneous device shows the increase in delay during transcoding. Figure-3.b shows the resulting graph of start-up time in relation in to the number of chunks used. It also shows the total transcoding time achieved while using the different chunks. Here the total transcoding time gets reduced, possibly due to smaller chunks.

Table-1. Comparison of heterogeneous devices.

Brand	OS	Resolution	fps	Audio format	Video format	Streaming rate	Transcoding delay	End-to-end delay
Blackberry 8520	BlackBerry OS 5.0	320 x 240	320p@24fps	MP3/eAAC+/WMA/WAV	MPEG-4	500kbps	4.087 sec	8.11 sec
Micromax A177	Android OS, v4.2.2 (Jelly Bean)	480x 854	720p@30fps	MP3/eAAC+/WMA/WAV	MPEG4, H.264, H.263	500kbps	7.013 sec	10.06 sec
Nokia lumia 520	Microsoft Windows Phone 8	480 x 800	720p@30fps	MP3/WAV/eAAC+/WMA	MP4/H.264/WMV player	500kbps	6.752 sec	11.8 sec
Nokia asha 510	Nokia Asha software platform 1.0	240 x 320	320p@15fps	MP3/WAV/AAC	MP4/H.263 player	500kbps	3.473 sec	6.4 sec
PC 1	WINDOWS 7	1280 x 1024	1024p@40fps	WMA,PCM,AAC,MP3	MPEG-4/2,AVI, WMV	500kbps	9.3 sec	14.3 sec
PC 2	Ubuntu 14.04	1920 x 1080	1080p@60fps	FLAC,MP3,AAC,AC3	MPEG-2, H.264	500kbps	10.16 sec	14.01 sec

VOL. 10, NO. 5, MARCH 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

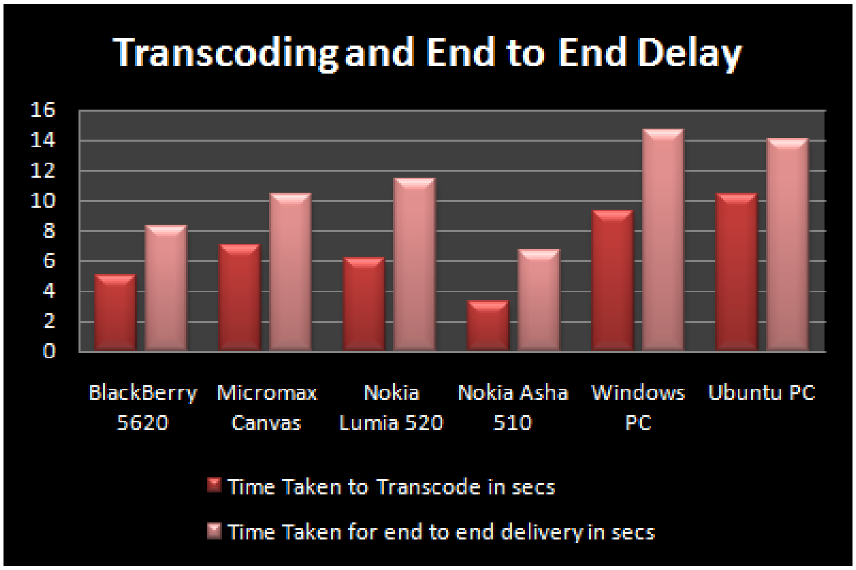

Figure-2. Comparative chart of transcoding and end-to-end delay.

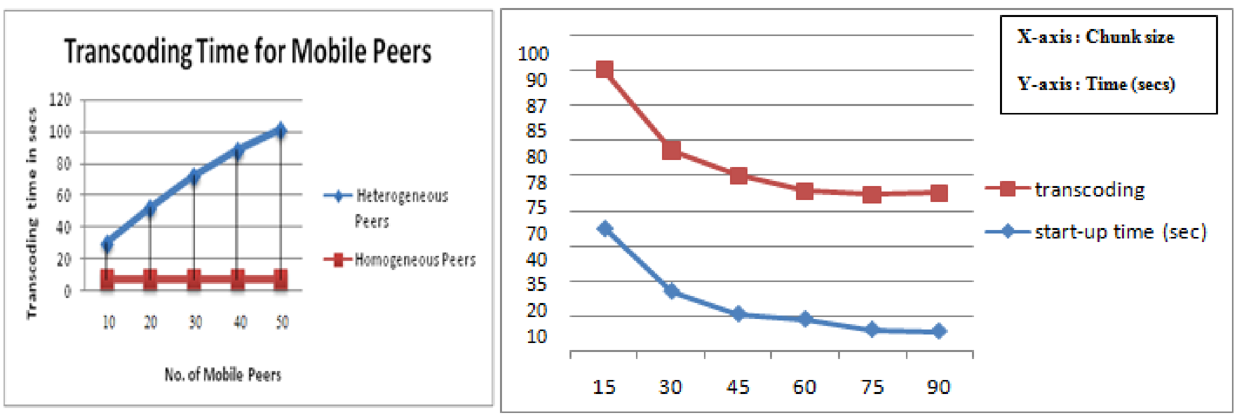

Figure-3. (a) Transcoding time for various peers: (b) Chart plotted for start-up time.

6. CONCLUSION AND FUTURE SCOPE

The contribution of this paper is novel 4-Tier Cloud Architecture which consists of physical layer of end users, p2p overlay, intermediate management layer and the final private cloud layer. The objective of the proposed system is to handle the heterogeneity of the mobile devices for live streaming by performing content adaptation on the fly. Various Content adaptation system were already developed for Video-on-Demand but in our proposed work Cloud based transcoding system is developed for live streaming which is a real challenge and has high scope for future work. From the experimental results the proposed system takes 9 sec on an average for transcoding and 14 sec on an average for end-to-end delivery of videos when heterogeneous devices are considered. This proves the performance of the proposed system to be highly efficient when compared with the existing systems.

The future work would be enhanced by varying streaming rate and size of the chunk. This system would be further developed with Hybrid cloud consisting of both private and public cloud set up where more peers would join the network and increase the complexity of transcoding and overall streaming delay.

REFERENCES

Yao Liu, Fei Li, Lei Guo, Bo Shen, Songqing Chen, Yingjie Lan. 2013. Measurement and Analysis of an Internet Streaming Service to Mobile Devices. In: IEEE Transaction on Parallel and Distributed Systems, vol. 24. pp. 2240-2250.

Zuquing Zhu, Suoheng Li, Xiaoliang Chen. 2013. Design QoS-Aware Multi-Path Provisioning Strategies for Efficient Cloud-Assisted SVC Video Streaming to Heterogeneous Clients. In: IEEE Transactions on Multimedia, vol. 15. pp.758-768.

Seungho Han, Myoungjin Kim, Yun Cui. 2014. CMTS: Cloud Multimedia Transcoding System. In: International

Journal of Advanced Computational and Networking, vol. 2. pp. 21-23.

Sangmi Lee, Sung Hoon Ko, Geoffrey Fox. 2003. Adapting Content for Mobile Devices in Heterogeneous Collaboration Environments. International Conference on Wireless Networks. p.211-217.

Y.Liu, L.Guo, F.Li, S.Chen. 2012. A Server's Perspective of Internet Streaming Delivery to Mobile Devices. In: IEEE INFOCOM. P.1332-1340.

Y Liu, F Li, L Guo, B Shen, S Chen. 2013. Effectively minimizing redundant Internet streaming traffic to iOS devices. In: IEEE INFOCOM. pp. 250-254.

Y Liu, F Li, L Guo, Y Guo, S Chen. 2011. BlueStreaming: towards power-efficient internet P2P streaming to mobile devices. In: Proceedings of the 19th ACM international conference on Multimedia. pp. 193-202.

Z. Huang, C. Mei, L. Li, and T. Woo. 2011. CloudStream: Delivering high-quality streaming videos through a cloud-based SVC proxy. In: IEEE INFOCOM. pp. 201-205.

M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon. 2007. I Tube, YouTube, Everybody Tubes: Analyzing the World's Largest User Generated Content Video System. In: Proc. of ACM IMC.

Y. Liu, L. Guo, F. Li, and S. Chen. 2011. An Empirical Evaluation of Battery Power Consumption for Streaming Data Transmission to Mobile Devices. In: Proc. of ACM Multimedia.

Y. Liu, F. Li, L. Guo, Y. Guo, and S. Chen. 2011. BlueStreaming: Towards Power-Efficient Internet P2P Streaming to Mobile Devices. In: Proc. Of ACM Multimedia.

Error! Hyperlink reference not valid.

http://idgknowledgehub.com/mobileidg/idg-mobile-survey/

Netflix. http://www.netflix.com.

Air Video. http://itunes.apple.com/app/id306550020.

Flash HTTP Streaming. http://www.adobe.com/products/httpdynamicstreaming/.

Online-convert. http://www.online-convert.com.