

CloudMoV: A Mobile Social TV System using Cloud Services

¹Clarance Angel A, ¹Gokulakisan S, ¹Jaya Surya S, ¹Keerthivasan Asokan and
²Daya Kanimozhi Rani M

¹B. Tech Information Technology

²M. Tech Software Engineering

clarancemicheal92@gmail.com, gokulakisan.it@gmail.com,
suryarock08@gmail.com, akeerthivasan.it@gmail.com, mkanidaya@gmail.com

Abstract

In recent trends, the rapidly increasing power of personal mobile devices (smartphones, tablets, etc.) is providing much richer contents and social interactions to users on the move. This trend however is throttled by the limited battery lifetime of mobile devices and unstable wireless connectivity, making the highest possible quality of service experienced by mobile users not feasible. The recent cloud computing technology, with its rich resources to compensate for the limitations of mobile devices and connections, can potentially provide an ideal platform to support the desired mobile services. In this paper, the design of a Cloud-based novel Mobile Social TV system (CloudMoV) is proposed. The system effectively utilizes both PaaS (Platform-as-a-Service) and IaaS (Infrastructure-as-a-Service) cloud services to offer the living-room experience of video watching to a group of disparate mobile users who can interact socially while sharing the video. Given the battery life as a key performance bottleneck, the use of burst transmission from the surrogates to the mobile users, and carefully decide the burst size which can lead to high energy efficiency and streaming quality is advocated. Social interactions among the users, in terms of spontaneous textual exchanges, are effectively achieved by efficient designs of data storage with Big Table and dynamic handling of large volumes of concurrent messages in a typical PaaS cloud.

Keywords: CloudMoV, Surrogate

1. Introduction

The previously available set up box technologies do allow the users to watch the video concurrently from different location, but the mobile viewing experience remains a challenging task. The CloudMoV effectively utilizes the cloud computing resources to offer a living-room experience of video streaming for a group of disparate mobile users. The system also allows these users with spontaneous social interactions. The CloudMoV blends the co-viewing experience and social interaction by allowing the users to import or even watch an on-demand video from any video streaming website. CloudMoV seamlessly utilize agile resource support and rich functionalities offered by both an IaaS cloud and a PaaS cloud. The work mainly focuses on the following

1.1 Universal Streaming

CloudMov allows the user to import or download an on demand video from any video streaming site or a program provider at any time with altered formats.

1.2 Encoding Flexibility

Different mobiles have different sized displays so to support the video streaming in all mobiles are employed with a surrogate which downloads the video on behalf of the user and then formats it so that it is supported in the mobile.

1.3 Battery Efficiency

A burst transmission algorithm is used to allot burst sizes and takes care between high/low power consumption modes in the devices so that it increases the battery lifetime.

1.4 Co-viewing with Social Exchanges

A user can invite multiple friends to watch the same video and exchange text messages while watching.

2. Related Work

Although several mobile TV systems were implemented in the recent years they do not concentrate on mobile access. NoSQL Data Base is an online database technique that does not allow for ad hoc data fixing there first has to be some kind of query and manipulation language. Secondly, it is more difficult to fix data in distributed databases (like Project Voldemort or Cassandra) compared to data stores that run on a single node or have dedicated shards.

The case for VM-based Cloudlets in mobile computing” by M. Satyanarayanan, *et al.*, states that it is not yet widely supported in IPTV set-top boxes. The Chunking process creates thousands of small files which can be hard to manage. Adobe Flash is a good solution — popularized by YouTube — for embedding videos in a webpage when worried about client compatibility. Because of this broad compatibility it is also a good candidate as a fall back technology for any of the other two protocols. Nevertheless, the proprietary nature and the lack of adoption in the mobile market are two important disadvantages.

TS files mux the audio, video and data together. This means that multi-language support either comes at the cost of sending all the languages in the chunks or creating duplicate chunks with each language. Similarly for data PIDs, these are either muxed together or multiple versions of chunks are needed with different data PID's [1].

The concepts from Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading by S. Kosta, *et al.*, states that the VCS running on the physical hosts, the KVM Guests get restarted as the application agent calls the shutdown and start programs. Therefore, live migration is not supported without downtime for the KVM running under VCS control.

A KVM Guest is created on the secondary node where the service group is supposed to fail over. VCS does not support the virtio driver. Veritas Volume M.63 anager does not recognize disks exported to guests over a virtio bus and Volume Set agents cannot be used. Moreover, Disk Reservation agent cannot work with disks exported over a virtio bus [2].

The concepts from Cloudstream: Delivering high-quality streaming videos through a cloud-based SVC proxy by Z. Huang et.al, states that the mechanics, while fairly well understood, mean that a significant amount of data “housekeeping” code exists in the application tier to manage cache locations and invalidation logic. This increases the complexity and brittleness of the overall application, especially as its features and functions grow over time. It also does nothing to improve the raw write throughput of the underlying database [3].

3. Architecture and Design

The CloudMov focuses on two major things such as improving the quality of the video being streamed as well as improving the battery power efficiency. The architecture of the CloudMov with its modules is being discussed in the following.

3.1 Key Modules

From Figure 1, a surrogate is created for each user. The surrogate acts as an intermediate between the mobile device and the video source. It is responsible for burst size allocation, sending of packets, segmentation of videos and social interaction. It consists of the following parts:

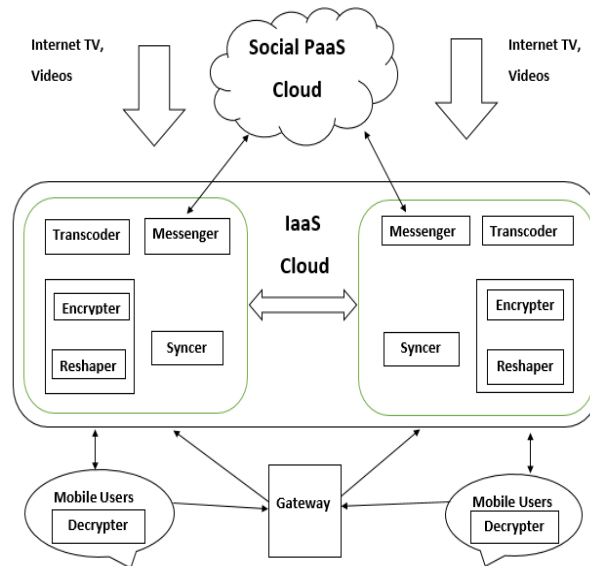


Figure 1. Architecture

3.1.1 Transcoder

It is used to encode the video in the appropriate format which is supported by the mobile device. In the system, the transcoder encodes the video in H.264 format which is a common high standard for streaming video these days.

3.1.2 Reshaper and Encrypter

The reshaper helps in segmentation of the video and then sends to the mobile device based on the request. In order to achieve high power efficiency the burst size are allotted based on the network capability.

3.1.3 Social Cloud

The prototype will be implemented in the GAE but in the future it can be implemented in any of the other platforms. This serves as an example for the portable nature of the system.

3.1.4 Messenger

It is responsible for the social interaction between the users. It periodically queries the social cloud for the social data. The data is sent by the cloud in plain text files which is a light weighted format. The surrogate asynchronously delivers the message with low traffic.

3.1.5 Syncer

The syncer guarantees that the viewing progress of the user is stored in the surrogate such that co-viewing experience is possible. For this to happen, the syncer periodically retrieves the playback information about the video being streamed by the user.

3.1.6 Mobile Client

A mobile device which supports HTML 5 browser is enough in order to support Cloud Mov. The mobile device must also support HTTP Live Streaming Protocol.

3.1.7 Gateway

The gateway is responsible for the authentication of the users. The user login credentials are stored in a permanent MySQL database. Whenever the user logs in, a surrogate is assigned from the VM pool. The VM pool is update frequently and then on log out, the VM instances are destroyed.

3.2 Loosely Coupled Interfaces

The different modules of the Cloud MoV are connected together using the HTTP protocol which is a universal standard for internet communication. Without any knowledge about the underlying technology the surrogates from the IaaS cloud communicates with the mobile devices using HTTP.

For social interaction the system uses asynchronous communication. The messages are the routed via the cloud and then encrypted with the video segments. Thus one packet from the cloud consists of the video information and the message from the user encrypted in it.

3.3 Pipelined Video Processing

The surrogates are designed in such a way that it is able to support both stored as well as on demand videos. It encodes the video directly and then sends it to the reshaper for segmentation and transmission so that as soon as the first segment is received the playback will be started. The IaaS platform in which the surrogates are deployed represent an ideal platform for implementing such jobs.

3.4 Burst Transmissions

The burst transmission focuses on three major things. The power state, transmission mechanism and the burst size of the video.

3.4.1 3G Power States

3G services suffers from lack of radio resources which results in the regulation of radio resources by the Radio Resource Control (RRC). There are several 3G carriers. Each 3G carrier customizes and deploys the complex states in their own network. The energy consumption is mainly based on the state in which the user is working on and not the volume of data that is being transmitted.

The 3G carrier works on high-power and low-power state. The user may be transmitted from a high-power state to a low power state when the traffic is more and vice versa. Timeouts are set to guarantee performance in delay and energy consumption.

3.4.2 Transmission Mechanism

The CloudMoV majorly focuses on energy consumption. The HTTP Live Streaming Protocol is used by the mobile devices to send the requests for the next segment of the video. The mobile device works in a high power state when it is receiving a segment and then transfers to a low power state when it is not receiving a segment.

3.4.3 Burst Size

The burst size is determined by the amount of segment that is being transmitted in one burst. When the user switches from one channel to the other the burst size already allocated will be wasted. Since the focus is primarily on energy consumption smaller burst sizes are used. But on the other hand when transmission occurs frequently the burst size must be relatively higher. Another reason for using a larger burst size is to avoid frequent state transition.

4. Proposed System

The system assigns a surrogate to each user who logs in to it. This surrogate allows us in assigning the burst size for the video streaming based on the user header values. The header values consists of the battery details, 3G power states and the video format that is being supported by the mobile client. The following are the steps that are being performed for video streaming.

4.1 Surrogate Allocation

Each user is provided with a user id and password for authentication. The user uses this log in information to login to the system. The login details are stored in a MySQL database. Whenever the user logs in, the user is assigned a VM instance from the VM pool. This VM instance is known as the surrogate. This surrogate is helpful in retrieving the header values of the mobile client. These header values consists of information about the power states, battery efficiency *etc.* Based on these header values the burst size for each user is allocated using the Burst Transmission Algorithm.

4.2 Burst Size Determination

The portal collects the device configuration information by examining the “User-Agent” header values, this information will be sent to its surrogate for decision making of the video encoding format. The surrogate on basis of the information received allocates the burst size using the Burst Transmission Algorithm. The surrogate also helps in transcoding the videos in the required format. The videos are stored in a default H.264 format. This H.264 video is then transcoded on the format which is supported by the user mobile device.

4.3 Video Streaming

When the mobile client subscribes to a video, the playlist is first downloaded. The surrogate by using the reshaper segments the video into multiple segments. The video is segmented as several 60 second segments. Then from the surrogate the video streams are delivered as multiple segments. The client starts to play the video as soon as the first segment is received.

If there is a text message then it is encrypted along with the video segment and then sent to the user. This encrypted message and video is decrypted by the mobile client. When the user plays a video his/her playback is recorded so that when another user watches the same video, the video is played from the same point. This helps in establishing a co-viewing experience.

5. Prototype Implementation

The system focuses on increasing the power efficiency of the mobile nodes during video streaming. The system uses the H.264 as a video formatting standard instead of the usual MPEG-2 which increases the efficiency of the system.

5.1 H.264

In common with earlier standards (such as MPEG1, MPEG2 and MPEG4), the H.264 standard does not explicitly define a CODEC (encoder / Decoder pair). Rather, the standard defines the syntax of an encoded video bit stream together with the method of decoding this bit stream. In practice, however, a compliant encoder and decoder are likely to include the functional elements.

Whilst the functions shown in these are likely to be necessary for compliance, there is scope for considerable variation in the structure of the CODEC. The basic functional elements (prediction, transform, quantization, entropy encoding) are little different from previous standards (MPEG1, MPEG2, MPEG4, H.261, H.263); the important changes in H.264 occur in the details of each functional element.

5.2 MPEG

Digital video data is encoded as a series of code words in a complicated manner that causes the average length of the code words to be much smaller than would be the case, if for example, each pixel in every frame was coded as an 8 bit value. This is also known as data compression. The standard allows for the encoding of video over a wide range of resolutions, including higher resolutions commonly known as HDTV.

Each 8x8 array of pixels is known as a block. A 2x2 array of blocks is termed a macro block. Compression is achieved by using well-known techniques of prediction (motion estimation in the encoder, motion compensation in the decoder), 2 dimensional discrete cosine transform(DCT) performed on 8x8 blocks of pixels, quantization of DCT coefficients, and Huffman and run/level coding. Pictures called I pictures are encoded without prediction. Pictures termed P pictures may be encoded with prediction from previous pictures. B pictures may be encoded using prediction from both previous and subsequent pictures.

The battery and the power efficiency while using the MPEG-2 and the H.264 video formatting standards the following changes have been identified.

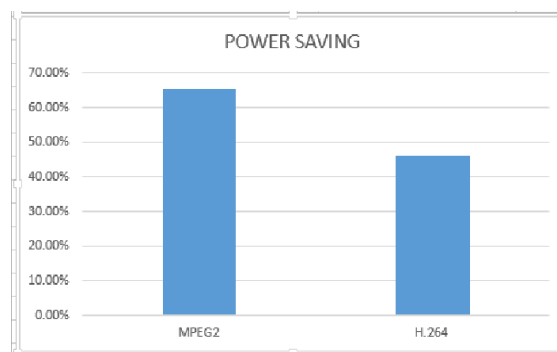


Figure 2. Power Consumption Graph

The Figure 2 depicts that the H.264 video formatting standard is being used for better power efficiency. In the older video streaming techniques, the formatting standard used was MPEG-2. The power consumption of both the video formatting standards are depicted in the above graph. The power consumption of MPEG-2 format is 65.1% whereas for

H.264 formatting standard it is 46.2%. This describes the lower power consumption of H.264 while compared to the MPEG-2 standard

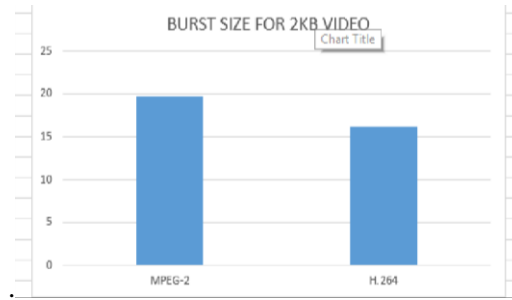


Figure 3. Burst Size Comparison

The Figure 3 represents the varying burst sizes of the MPEG-2 and the H.264 video formatting standard. Let us consider a 2kb video, the burst size for that video will be 19.7 whereas for the same video the burst size for H.264 format will be 16.2 which is lower.

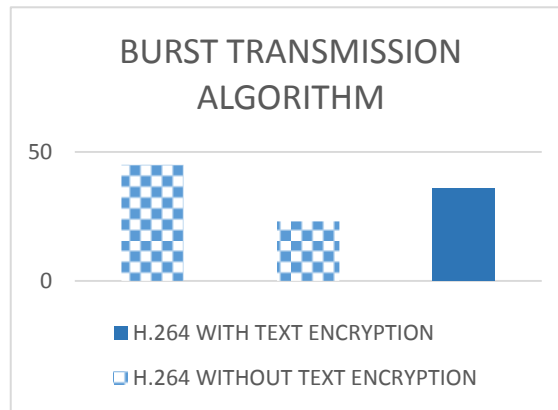


Figure 4. Burst Size Comparison With and Without Text Encryption

The Figure 4 helps in the comparison of the burst when the text is embedded and when the text is not embedded. When sending the video segment and the text separately two transmissions occur with a bigger burst size. Since the encryption of the text and video is done it can be sent using a single transmission so that the burst size is relatively low.

6. Conclusion And Future Work

The proposed system is soon to become a trend in mobile TV system. The primary focus of the system is improving battery efficiency. The system uses the H.264 video format as a standard in the burst transmission algorithm. The system also segments the video into multiple 60s segments so that burst size of the video increases. The social interaction is done using a syncer which allows us to merge the video segment and the plain text in a single packet. By sending the video segment and the message in a single packet, the efficiency of the system increases to quite an extent.

The work maybe extended in the future by using the upcoming latest video formatting standards (ie) instead of the H.264 the newer video standards maybe used to increase the efficiency of the system. By configuring the HTTP Live Streaming Protocol battery efficiency can be increased. Leaky Bucket Algorithm is an alternative to the Burst transmission algorithm but the efficiency change yet remains unknown.

References

- [1] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for VM-based Cloudlets in mobile computing," *IEEE Pervasive Comput.*, vol. 8, (2009), pp. 14–23.
- [2] S. Kosta, A. Aucinas, P. Hui, R. Mortier and X. Zhang, "Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading," in *Proc. IEEE INFOCOM*, (2012).
- [3] Z. Huang, C. Mei, L. E. Li and T. Woo, "Cloudstream: Delivering high-quality streaming videos through a cloud-based SVC proxy," in *Proc. INFOCOM'11*, (2011), pp. 201–205.
- [4] T. Coppens, L. Trappeniniers and M. Godon, "AmigoTV: Towards a social TV experience," in *Proc. Euro ITV*, (2004).
- [5] N. Ducheneaut, R. J. Moore, L. Oehlberg, J. D. Thornton and E. Nickell, "Social TV: Designing for distributed, sociable television viewing," *Int. J. Human-Comput. Interaction*, vol. 24, no. 2, (2008), pp. 136–154.
- [6] A. Carroll and G. Heiser, "An analysis of power consumption in a smartphone," in *Proc. USENIXATC*, (2010).
- [7] "What is 100% Pure Java", [Online] Available: <http://www.javacoffeebreak.com/faq/faq0006.html>.
- [8] J. Santos, D. Gomes, S. Sargento, R. L. Aguiar, N. Baker, M. Zafar and A. Ikram, "Multicast/broadcast network convergence in next generation mobile networks," *Comput. Netw.*, vol. 52, (2008) January, pp. 228–247.
- [9] K. Chorianopoulos and G. Lekakos, "Introduction to social TV: Enhancing the shared experience with interactive TV," *Int. J. Human-Comput. Interaction*, vol. 24, no. 2, (2008), pp. 113–120.
- [10] M. Chuah, "Reality instant messaging: Injecting a dose of reality into online chat," in *CHI '03 Extended Abstracts on Human Factors in Computing Syst.*, ser. CHI EA '03, (2003), pp. 926–927.
- [11] R. Schatz, S. Wagner, S. Egger and N. Jordan, "Mobile TV becomes social – Integrating content with communications," in *Proc. ITI*, (2007).
- [12] R. Schatz and S. Egger, "Social interaction features for mobile TV services," in *Proc. 2008 IEEE Int. Symp. Broadband Multimedia Syst. And Broadcasting*, (2008).
- [13] "HTTP Live Streaming", [Online] Available: <http://tools.ietf.org/html/draft-pantos-http-live-streaming-01>.
- [14] "Kernel Based Virtual Machine", [Online]. Available: <http://www.linuxkvm.org/>.
- [15] "Google App Engine", [Online]. Available: <http://appengine.google.com/>.