Streaming of High Quality videos using Cloud

M.Nagesh, B.Naresh Achari
PG Student, Assistant Professor,
Department of Computer Science & Engineering,
Shri Shirdi Sai Institute of Science and Engineering, Anantapuramu.
Nageshrthd@gmail.com, bnachari@gmail.com.

ABSTRACT

Recently many mobile entertaining applications have been launched, but most popular app like Facebook, Twitter, YouTube have large demand among users. But these media applications are limited by the unstable wireless connectivity and limited battery lifetime of mobile devices. Due to these problems the quality of service encountered by the users of the Smartphone is not up to the mark. In order to overcome these problems cloud computing technology has been used. In this paper we review the design of the Streaming of high quality videos using Cloud platform.

Keywords-Battery lifetime, unstable connectivity, Mobile social TV.

1. INTRODUCTION

Nowadays laptops, Notepad and Smartphone are shipped with many microprocessor cores and GB’s of RAM’s, they have high computation power than Normal desktop computers of late 90’s. The wide development of 3G, 4G broadband cellular infrastructure has further increased the use of Smartphone by common people. Every Smartphone users need the fastest technologies like 3G, Wi-Fi for fast web access & chatting. These technologies focus more on the challenging scenarios such as real-time video streaming and online gaming, for social interacting, and exchanging emails. Recently many mobile entertaining or media applications have been launched, but most popular app like Facebook, Twitter, YouTube have larger demand among users. But these Media applications are limited by the unstable wireless connectivity and limited battery lifetime of mobile devices, due to these problems the quality of service encountered by the users of the Smartphone is not up to the mark. In order to overcome from these problems cloud computing technology has been used. Here, we propose the design of the Streaming of high quality videos using Cloud.

Cloud computing provides low-cost, agile scalable resource supply and power efficient communication between streaming devices. Cloud can reduce load of computation and other tasks which is involved in a mobile application. This significantly reduces battery consumption of the mobile devices. It effectively utilizes cloud computing standards to offer a co-viewing experience of video watching just as users are watching TV at their homes.

In the mobile social TV, the system effectively makes use of the cloud computing services like Infrastructure-as-a-Service(IaaS) and Platform-as-a-Service(PaaS) to provide good quality of video streaming, a mobile users can fetch a on-demand or live video to watch from video gallery and also you can invite your family and friends for watching the video. They can also chat with each other while watching the video. In traditional system each user uses dish TV, set boxes for digital broadcasting of channels.

Users are demanding uninterrupted delivery of increasingly higher quality videos over the Internet, in both wireless and wireline. Instead of tackling the video delivery problem head on, most current Internet media providers (like YouTube or Hulu) have taken the easy way out and changed the problem to that of a progressive download via a content distribution network. In such a framework, they are using a non-adaptive codec, but ultimately, the delivery variabilities is handled by freezing, which significantly degrades the user experience. To enable good video quality we study the development of SVC video proxy. Two major functions of proxy are (1) video transcoding from original formats to SVC, and (2) video streaming to different users under Internet dynamics. Because of codec incompatibilities, a video proxy will have to decode an original video into an intermediate format and re-encode it to SVC. While the video decoding overhead is negligible, the encoding process is highly complex that the transcoding speed is relatively slow even on a modern multicore processor. This results in a long duration before a user can access the transcoded video(called video access time), and possible video freezes during its playback because of the unavailability of transcoded video data.
To enable real-time transcoding and allow scalable support for multiple concurrent videos, video proxy employs a cluster of computer or a cloud for its operation. Specifically, our proxy solution partitions a video into clips and maps them to different computer nodes (instances) configured with one or multiple CPUs in order to achieve encoding parallelization. In general, video clips with the same duration can demand different computation time because of the video content heterogeneity. The CloudStream utilizes agile resource support to the functionalities which are Platform-as-a-Service (PaaS) and Infrastructure-as-a-Service (IaaS) to achieve the goals.

1.1. Encoding Flexibility

There are various mobile devices which have large screen size and small displays also they have various screen resolutions. The mobile phones support for customized media playback hardware, video playback and also support for various codec. This system unloads the transcoding streams of different devices at real time in an IaaS Cloud.

A surrogate is employed for each user, which is a virtual machine(VM) in the IaaS cloud. The surrogate downloads the video and transcodes it into the proper formats, while considering particular configurations of the mobile device as well as the current connectivity quality.

1.2. Battery Efficiency

Analysis indicates that the display and network modules both 3G and Wi-Fi absorb maximum power in video streaming device to save energy consumption from the network of video streaming devices cloud computing technologies are used, Cloud can reduce load of computation and transcoding tasks which is involved in a video streaming application.

1.3. Spontaneous Social Interactivity

Multiple mechanisms are involved in the design of Streaming of high quality videos for concurrent viewing and social chatting with each other. First factor is efficient synchronization mechanisms in which friends joining in a video program may watch the same portion and share their views and comments about video with each other.

Second factor is an efficient message communication mechanism which is designed for social interactions among friends. PaaS cloud provides these mechanisms through data storage of BigTable. PaaS is proven model for running applications without hassle of maintaining hardware and software infrastructure at the company salesforce.com for the simplicity, scalability and reliability. PaaS cloud can be used for social interaction support due to its provision of robust underlying platforms. IaaS is the foundation of cloud computing. Some space can be taken on rent in data centers from IaaS provider to maintain and deploy computer servers, cloud networks and storage.

1.4. Portability

A prototype Cloud MoV is implemented following the philosophy of “Write Once, Run anywhere” (WORA): both the front-end mobile modules and the back-end mobile modules are implemented in Java, Android and MySql database is used for storing user data. The client module can run on any mobile devices supporting HTML5, including Android phones. In Peer to Peer sharing of Cloud Based Mobile Social TV the videos are shared between two users using peer to peer technology. Real Time Streaming Protocol (RTSP) is used for online streaming of videos in the mobile phones and open Fire Char Server for chatting with each other.

II. ARCHITECTURE OF CLOUDSTREAM

In this paper, we describe the design of a Streaming of High Quality videos using Cloud. CloudStream can effectively utilize the cloud computing paradigm to offer a living-room experience of video watching to disparate mobile users with spontaneous social interactions. Fig. 1 gives an overview of the architecture of CloudStream.

The modules used in the architecture of CloudStream are:-

A VM Proxy Server: It is a proxy server which acts between online video streaming sites like YouTube sites and mobile devices which provides transcoding services to the user. The proxy server handles social messages between users in proper efficient way. In CloudStream we have gateway server which tracks participating users and their VM surrogates. CloudStream has following major functional modules which are as follows:

Video converter: Online video converter which is transcoder is part of surrogate i.e. proxy server. The transcoder transforms video from video streaming sites like YouTube into proper format which can be used on mobile devices.

Reshaper: It receives the encoded stream which divides it into segments and sends each converted stream into mobile devices.

Google Social PaaS Cloud: Google Social cloud stores all the social data in the system, including the online
statues of all users, records of the existing sessions, user login details and messages.

**Fig. 1: Architecture of CloudStream**

**Messenger:** It is the client side component of mobile user which can be used for chatting and exchanges of messages between users. The user can share his views and opinions, photos and videos with his friends and family using this messenger.

**Syncer:** It is component of surrogate which can be used to retrieve user viewing status within certain time limit.

**Mobile Client:** It is user which can access Messenger, and can watch videos on to his Mobile using HTML5 compatible browser which are Google Chrome.

**Gateway:** It verifies the user’s login details and stores user’s login details into MySQL database. It can also store pool of videos into database.

**Burst Transmissions**

1) **3G Power States:** Different from Wi-Fi which is more similar to the LANed Internet access, 3G cellular services suffer from the limited radio resources, and therefore each user equipment (UE) needs to be regulated by a Radio Resource Control (RRC) state machine [19]. Different 3G carriers may customize and deploy complex states in their respective cellular networks. A 3G carrier may commonly transfer a UE from a high-power state to a low power state (state demotion), for releasing radio channels allocated to this UE to other users. For example, if a UE working at a high-power state does not incur any data traffic for a pre-configured period of time (measured by a critical inactivity timer), the state of the UE will be transferred to a low-power one; when the volume of data traffic rises, the UE “wakes up” from a low-power state and moves to a high-power one. Timeout of the critical inactivity timers for state transitions are properly set by the carrier to guarantee performance in both delay and energy consumption, since extra delay and energy consumption are potentially incurred for acquiring new radio channels when the UE transits from a low-power state to a high-power one later (state promotion).

2) **Transmission Mechanism:** Using the HTTP live streaming protocol [18], the mobile device sends out requests for the next segment of the video stream from time to time. The surrogate divides the video into segments, and sends each segment in a burst transmission to the mobile device, upon such a request.

3) **Burst Size:** To decide the burst size, i.e., the size of the segment transmitted in one burst, we need to take into consideration characteristics of mobile streaming and energy consumption during state transitions. For video streaming using a fixed device without power concerns, it is desirable to download as much of the connection bandwidth allows; however, for streaming over a cellular network, we should a video as what avoid downloading more than what is being watched for one main reason: users may switch among channels from time to time and those prefetched contents are probably never watched, leading to a waste of the battery power and the cellular data fee due to their download. Hence, the bursty size should be kept small, to minimize battery consumption and traffic charges and this in return helps saving the battery lifetime.

**III. PROTOTYPE IMPLEMENTATION**

Following the design guidelines in Section III, since our implementation is done on Java platform, we can deploy our system in Google App Engine (GAE) [As a matter of choice] and Rackspace (freely available cloud service) which are most commonly used PaaS and IaaS platforms respectively. GAE, as a PaaS cloud, provides rich services on top of Google’s data centers and enables rapid deployment of Java-based and Python-based applications. Hence, GAE is an ideal platform for implementing our social cloud, which dynamically handles large volumes of messages. On the other hand, GAE imposes many constraints on application deployment, example, lack of support for multi-threading, file storage, etc. Rackspace is a representative IaaS cloud, offering raw hardware resources including CPU, storage, and networks to users. Rackspace has two main service-level segments: Managed and Intensive. Both service levels receive support via e-mail, telephone, live chat, and ticket systems, but they are designed to fit the needs of
different businesses. The Managed support level consists of "on-demand" support where proactive services are provided, but the customer can contact Rackspace when they need additional assistance. The Intensive support level consists of "proactive" support where many proactive services are provided, and customers receive additional consultations about their server configuration. Highly customized implementations generally fall under this level of support.

A. Client Use of Cloud Mobile TV

Android is used for programming for the client mobile devices. Our Cloud based Mobile social TV is installed with HTML5 compatible browsers can use Cloud based mobile TV services, as long as the HTTP Live Streaming (HLS) [24] protocol is supported, for achieving this have used Used the Http servelet objects for the interface between the data owner and cloud system. The user first connects to the login page of application, after the user successfully log in through the gateway (Third Party Auditor), User is assigned a VM surrogate from the VM pool (Multi-threading) user is automatically redirected to the assigned VM surrogate, and welcomed by a portal page. The user can enter the filename of the video which downloads the stream on the user’s behalf, converted video and sends properly encoded segments to the user. From the surrogate to the mobile device, the video stream delivered using HLS is always divided into multiple segments, with a playlist file giving the indices. The client starts to play the video as soon as the first segment is received. When watching a video, the user can check for their friends’ messages and invite them to join in watching the video. Users in the same session can exchange opinions and comments on the “Chat” tab where new chat messages can be entered and the chat history of the session is shown.

B. VM Surrogates

All the VM surrogates are provisioned from Rackspace web services and tracked by the gateway. We have also installed a Tomcat web server (version 6.5) to serve as a Servlet container and a file server on each Surrogate and process the video stream by video converting and segmentation. For example, in our experiments, since we are working in better speed of internet we have excluded the different streaming part dynamically, but we have the proposed system to implement high-quality stream to have “480 x 272” resolution with 24 frames per second, while the low-quality one has a “240 x 136” resolution with 10 frames per second. The transcoded stream is further exported to an MPEG-2 transporting stream (.ts), which is segmented for burst transmission to the user.

![Fig.](image)

Fig. 2. Social message exchange via Google App Engine

C. Data Models in the Social Cloud

Google App Engine is mainly used as the back-end data store keep online presence status, social messages (invitation and chat messages) in all the sessions With Jetty as the underlying Servlet container, most Java-based applications can be easily migrated to GAE, under limited usage constraints, where no platform-specific APIs are enforced for the deployment.

GAE provides both can be easily migrated to other PaaS clouds as well. If the user wishes to synchronize his playback progress with that of the session host, his VM surrogate synchronizes with the session host to maintain the playback “current time” value (HTML5 property).

The social cloud maintains a “Logs” entry for each existing session in “Streaming of High Quality videos using Cloud” with the session ID as the primary key and an array list as the value, which corresponds to individual messages in this session.

When a user in a session posts a comment, this message is first sent to his VM surrogate, which further injects the message into the social cloud via another Servlet listener. The message is stored as a “Message” entry in the social cloud, with the message content as the value, and an auto-generated integer as the key. This message can then be viewed by the client, the user can also reply to the messages that has been received, hence this leads to a chat or and interaction which is socially among the users using the cloud mobile TV.

IV. REMARKS AND FUTURE WORK

Our work has concerned primarily in suggesting a framework integrating the social interactive live streaming with the Streaming of High Quality videos using Cloud. The enhancement in the CloudStream of the user obviously shows great scope in the further feature enhancement with the streamed video contents, like as recording the live video contents to the device, noise removal and so on. On the other side the interactive aspect of the framework has several areas to
be more deeply addressed like applying memcache support and more efficient transcoding mechanism.

V. CONCLUSION

We conclude by proposing a framework for enriching the quality of experience to the users in an interactive live streaming, with the cloud computing as its backbone. And mobile users can import a live or on-demand video to watch from any video streaming site, invite their friends to watch the video concurrently, and chat with their friends while enjoying the video.

REFERENCES


M.Nagesh received his B.E degree from the department of Computer Science and Engineering from Er.Perumal Manimekalai College of Engineering, Hosur, Tamil Nadu (Affiliated to Anna University, Coimbatore). He is pursuing M.Tech from the department of Computer Science and Engineering in Shri Shirdi Sai Institute of Science and Engineering (Affiliated to JNTUA). His current research interests includes “Streaming of High Quality videos using Cloud”.

Mr.B.Naresh Achari is currently working as an assistant professor in CSE Department at Shri Shirdi Sai Institute of Science and Engineering (Affiliated to JNTUA). He received his B.Tech Degree from SKD Engg.College, Gooty (Affiliated to JNTUA). He received his M.Tech Degree from G.Pulla Reddy Engineering College, Kurnool (Autonomous). His research interests includes “Reversible Data Hiding in Encrypted image and Separating the image and Data”.

www.ijrcct.org