ISSN-2320-7884 (Online)
ISSN-2321-0257 (Print)

A Novel Mobile Social TV System based on Cloud Environment

Syeeda Meherunisa[1], G. Preethi Jyosthna[2]

[1]M.Tech (CSE), Nimra College of Engineering and Technology, A.P., India.

[2]Asst. Professor, Dept. of Computer Science & Engineering, Nimra College of Engineering and Technology, A.P., India.

Abstract—Personal Mobile devices (smartphones, tablets, etc.) enhanced and enriched in such a way that it is providing much richer contents and social interactions to users on the move. This trend however is curbed 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. Tough challenges arise on how to effectively exploit cloud resources to facilitate mobile services, especially those with stringent interaction delay requirements. In this paper, we propose the design of a Novel Mobile Social TV System based on Cloud Environment. The system effectively utilizes both PaaS (Platform-as-a-Service) and IaaS (Infrastructure-asa- 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. To guarantee good streaming quality as experienced by the mobile users with time-varying wireless connectivity, we employ a surrogate for each user in the IaaS cloud for video downloading and social exchanges on behalf of the user. The surrogate performs efficient stream transcoding that matches the current connectivity quality of the mobile user. Given the battery life as a key performance bottleneck, we advocate 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. Social interactions among the users, in terms of spontaneous textual exchanges, are effectively achieved by efficient designs of data storage with BigTable and dynamic handling of large volumes of concurrent messages in a typical PaaS cloud. These various designs for flexible transcoding capabilities, battery efficiency of mobile devices and spontaneous social interactivity together provide an ideal platform for mobile social TV services.

Keywords — Cloud Computing, Mobile devices, Social TV, Surrogate, Paas, Iaas

I. INTRODUCTION

Now-a-days smartphones are shipped with multiple microprocessor cores and gigabyte RAMs, thanks to the revolutionary “reinventing the phone” campaigns initiated by Apple Inc. in 2007, they possess more computation power than personal computers of a few years ago. On the other hand, the wide deployment of 3G broadband cellular infrastructures further fuels the trend. Apart from common productivity tasks like emails and web surfing, smartphones are flexing their strengths in more challenging scenarios such as real-time video streaming and online gaming, as well as serving as a main tool for social exchanges.

Despite the fact that many mobile social or media applications have emerged, truly killer ones gaining mass acceptance are still inhibited by the limitations of the current mobile and wireless technologies, among which battery lifetime and unstable connection bandwidth are the most difficult ones. It is natural to resort to cloud computing, the newly-emerged computing paradigm for low-cost, and agile, scalable resource supply, to support power-efficient mobile data communication. With virtually infinite hardware and software resources, the cloud can offload the computation and other tasks involved in a mobile application and may significantly reduce battery consumption at the mobile devices, if a proper design is in place. The big challenge in front of us is how to effectively exploit cloud services to facilitate mobile applications. There have been a few studies on designing mobile cloud computing systems [1][2][3], but none of them deal in particular with stringent delay requirements for spontaneous social interactivity among mobile users.

In this paper, we describe the design of a novel mobile social TV system, which can effectively utilize the cloud computing paradigm to offer a living-room experience of video watching to disparate mobile users with spontaneous social interactions. In our system, 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. It therefore blends viewing experience and social awareness among friends on the go. As opposed to traditional TV watching, mobile social TV is well suited to today's life style, where family and friends may be separated geographically but hope to share a co-viewing experience. While social TV enabled by set-top boxes over the traditional TV systems is already available [4] [5], it remains a challenge to achieve mobile social TV, where the concurrently viewing experience with friends is enabled on mobile devices.

We design our system to seamlessly utilize swift resource support and rich functionalities offered by both an IaaS (Infrastructure-as-a-Service) cloud and a PaaS (Platform-asa- Service) cloud. Our design achieves the following goals.

Encoding flexibility. Different mobile devices have differently sized displays, customized playback hardwares, and various codecs. Traditional solutions would adopt a few encoding formats ahead of the release of a video program. But even the most generous content providers would not be able to attend to all possible mobile platforms, if not only to the current hottest models. Our system customizes the streams for different devices at real time, by offloading the transcoding tasks to an IaaS cloud. In particular, we newly employ a surrogate for each user, which is a virtual machine (VM) in the IaaS cloud. The surrogate downloads the video on behalf of the user and transcodes it into the desired formats, while catering to the specific configurations of the mobile device as well as the current connectivity quality.

Battery efficiency. A breakdown analysis conducted by Carroll et al. [6] indicates that the network modules (both Wi- Fi and 3G) and the display contribute to a significant portion of the overall power consumption in a mobile device, dwarfing usages from other hardware modules including CPU, memory, etc. We target at energy saving coming from the network module of smartphones through an efficient data transmission mechanism design. We focus on 3G wireless networking as it is getting more widely used and challenging in our design than Wi-Fi based transmissions. Based on cellular network traces from real-world 3G carriers, we investigate the key 3G configuration parameters such as the power states and the inactivity timers, and design a novel burst transmission mechanism for streaming from the surrogates to the mobile devices. The burst transmission mechanism makes careful decisions on burst sizes and opportunistic transitions among high/low power consumption modes at the devices, in order to effectively increase the battery lifetime.

Spontaneous social interactivity. Multiple mechanisms are included in the design of our system enable spontaneous social, co-viewing experience. First, efficient synchronization mechanisms are proposed to guarantee that friends joining in a video program may watch the same portion (if they choose to), and share immediate reactions and comments. Although synchronized playback is inherently a feature of traditional TV, the current Internet video services (e.g., Web 2.0 TV) rarely offer such a service. Second, efficient message communication mechanisms are designed for social interactions among friends, and different types of messages are prioritized in their retrieval frequencies to avoid unnecessary interruptions of the viewing progress. For example, online friend lists can be retrieved at longer intervals at each user, while invitation and chat messages should be delivered more timely. We adopt textual chat messages rather than voice in our current design, believing that text chats are less distractive to viewers and easier to read/write and manage by any user.

Portability. A prototype of our system is implemented following the philosophy of "Write Once, Run Anywhere"
(WORA): both the front-end mobile modules and the backend server modules are implemented in "100% Pure Java" [7], with well-designed generic data models suitable for any BigTable-like data store; the only exception is the transcoding module, which is implemented using ANSI C for performance reasons and uses no platform-dependent or proprietary APIs. The client module can run on any mobile devices supporting HTML5, including Android phones, iOS systems, etc. To showcase its performance, we deploy the system on Amazon EC2 and Google App Engine, and conduct thorough tests on iOS platforms. Our prototype can be readily migrated to various cloud and mobile platforms with little effort.

II. RELATED WORK

A number of mobile TV systems have emerged in recent years, driven by both hardware and software advances in mobile devices. Some early systems [8][9] bring the "livingroom" experience to small screens on the move. But they focus more on barrier clearance in order to realize the convergence of the television network and the mobile network, than exploring the demand of "social" interactions among mobile users. There is another trend in which efforts are dedicated to extending social elements to television systems [4] [5][10].

In [4] Coppens et al. try to add rich social interactions to TV but their design is limited to traditional broadcast program channels.

In [5] Oehllberg et al. conduct a series of experiments on human social activities while watching different kinds of programs. Though inspiring, these designs are not that suitable for being applied directly in a mobile environment.

Chuah et al. [11] extend the social experiences of viewing traditional broadcast programs to mobile devices, but have yet to deliver a well integrated framework.

Schatz et al. [12][13] have designed a mobile social TV system, which is customized for DVBH networks and Symbian devices as opposed to a wider audience. Compared to these prior work and systems, we target at a design for a generic, portable mobile social TV framework, featuring co-viewing experiences among friends over geographical separations through mobile devices. Our framework is open to all Internet-based video programs, either live or on-demand, and supports a wide range of devices with HTML5 compatible browsers installed, without any other

ISSN-2320-7884 (Online)
ISSN-2321-0257 (Print)

mandatory component on the devices. Spotnet therefore uses both wired and wireless links and corresponding protocols.

For any application targeted at mobile devices, reducing power consumption is perennially one of the major concerns and challenges. Flinn et al. [14] exploit collaborations between the mobile OS and the mobile applications to balance the energy conservation and application performance. Yuan et al. [15] investigate mobile multimedia streaming, similar to most of the other work, by adjusting the CPU power for energy saving; however, according to the recent measurement work of Carroll et al. [6], the display and the wireless network card (including the cellular module) and not the CPU consume more than half of the overall power consumption in smart phones nowadays. Our work is able to achieve a significant (about 30%) power saving, by opportunistically switching the device between high-power and low-power transmission modes during streaming. Some existing work (e.g., Anastasi et al. [16]) have provided valuable guidelines for energy saving over WiFi transmissions; our work focuses on 3G cellular transmissions which have significantly different power models; 3G is a more practical wireless connection technology for mobile TVs on the go at the present time.

A recent work by Zhang et al. [16] investigates the media caching management problem under HTTP adaptive bit rate streaming over a wireless network environment, which can complement our work when video streams are required to be transcoded into multiple bit rates.

III. PROPOSED WORK

As a novel Mobile social tv system provides two major functionalities to participating mobile users: (1) **Universal streaming.** A user can stream a live or on-demand video from any video sources he chooses, such as a TV program provider or an Internet video streaming site, with tailored encoding formats and rates for the device each time. (2) **Co-viewing with social exchanges**. A user can invite multiple friends to watch the same video, and exchange text messages while watching. The group of friends watching the same video is referred to as a session. The mobile user who initiates a session is the host of the session. We present the architecture of our system and the detailed designs of the different modules in the following.

A. Key Modules

Fig. 1 gives an overview of the architecture of our system.

A surrogate (i.e., a virtual machine (VM) instance), or a VM surrogate equivalently, is created for each online mobile user in an IaaS cloud infrastructure. The surrogate acts as a proxy between the mobile device and the video sources, providing transcoding services as well as segmenting the streaming traffic for burst transmission to the user. Besides, they are also responsible for handling frequently exchanged social

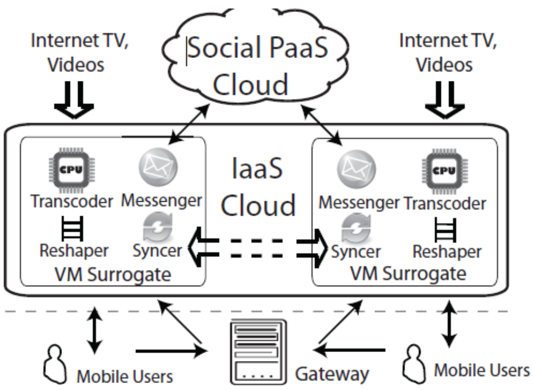

Figure 1 Architecture of our system

messages among their corresponding users in a timely and efficient manner, shielding mobile devices from unnecessary traffic and enabling battery efficient, spontaneous social interactions. The surrogates exchange social messages via a back-end PaaS cloud, which adds scalability and robustness to the system. There is a gateway server in the system that keeps track of participating users and their VM surrogates, which can be implemented by a standalone server or VMs in the IaaS cloud. The design of our system can be divided into the following major functional modules.

Transcoder. It resides in each surrogate, and is responsible for dynamically deciding how to encode the video stream from the video source in the appropriate format, dimension, and bit rate. Before delivery to the user, the video stream is further encapsulated into a proper transport stream. In our implementation, each video is exported as MPEG-2 transport streams, which is the de facto standard nowadays to deliver digital video and audio streams over lossy medium.

Reshaper. The reshaper in each surrogate receives the encoded transport stream from the transcoder, chops it into segments, and then sends each segment in a burst to the mobile device upon its request (i.e., a burst transmission mechanism), to achieve the best power efficiency of the device. The burst size, i.e., the amount of data in each burst, is carefully decided according to the 3G technologies implemented by the corresponding carrier.

Social Cloud. The social cloud is built on top of any general PaaS cloud services with BigTable-like data store to yield better economies of scale without being locked down to any specific proprietary platforms. Despite its implementation on Google App Engine (GAE) as a proof of concept, our prototype can be readily ported to other platforms. It stores all the social data in the system, including the online statuses of all users, records of the existing sessions, and messages (invitations and chat histories) in each session. The social data are categorized into different kinds and split into different entities (in analogy to tables and rows in traditional relational database, respectively). The social cloud is queried from time to time by the VM surrogates.

Messenger. It is the client side of the social cloud, residing in each surrogate in the IaaS cloud. The Messenger periodically queries the social cloud for the social data on behalf of the mobile user and pre-processes the data into a light-weighted format (plain text files), at a much lower frequency. The plain text files (in XML formats) are asynchronously delivered from the surrogate to the user in a traffic-friendly manner, i.e., little traffic is incurred. In the reverse direction, the messenger disseminates this user's messages (invitations and chat messages) to other users via the data store of the social cloud.

Syncer. The syncer on a surrogate guarantees that viewing progress of this user is within a time window of other users in the same session (if the user chooses to synchronize with others). To achieve this, the syncer periodically retrieves the current playback progress of the session host and instructs its mobile user to adjust its playback position. In this way, friends can enjoy the "sitting together" viewing experience. Different from the design of communication among messagers, syncers on different VM surrogates communicate directly with each other as only limited amounts of traffic are involved.

Mobile Client. The mobile client is not required to install any specific client software in order to use CloudMoV, as long as it has an HTML5 compatible browser (e.g., Mobile Safari, Chrome, etc.) and supports the HTTP Live Streaming protocol. Both are widely supported on most state-of-the-art smartphones.

Gateway. The gateway provides authentication services for users to log in to the CloudMoV system, and stores users' credentials in a permanent table of a MySQL database it has installed. It also stores information of the pool of currently available VMs in the IaaS cloud in another in-memory table. After a user successfully logs in to the system, a VM surrogate will be assigned from the pool to the user. The in-memory table is used to guarantee small query latencies, since the VM pool is updated frequently as the gateway reserves and destroys VM instances according to the current workload. In addition, the gateway also stores each user's friend list in a plain text file (in XML formats), which is immediately uploaded to the surrogate after it is assigned to the user.

B. Loosely Coupled Interfaces

For social message exchanges among friends, our system employs asynchronous communication. All the exchanged messages are routed via the surrogates to the social cloud, which efficiently organizes and stores the large volumes of data in a BigTable-like data store. The VM surrogates query the social cloud frequently and processes the retrieved data into XML files, for later retrieval by users in an asynchronous fashion. Such a design effectively separates the mobile users from the social cloud to significantly simplify the architecture, while the extra delay introduced at the VM surrogates is ignorable.

C. Pipelined Video Processing

Both live streaming of real-time contents and on-demand streaming of stored contents are supported in our system. Video processing in each surrogate is designed to work on the fly, i.e., the transcoder conducts real-time encoding from the video source, the encoded video is fed immediately into the reshaper for segmentation and transmission, and a mobile user can start viewing the video as soon as the first segment is received. To support dynamic bit rate switch, the transcoder launches multiple threads to transcode the video into multiple bit rates once the connection speed between the surrogate and the mobile user changes. The IaaS cloud where the surrogates are deployed, represents an ideal platform for implementing such computation intensive jobs.

D. Burst Transmissions

A 3G carrier may commonly transfer a UE from a highpower 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. Timeouts 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).

IV. CONCLUSION

This paper presents our view of what might become a trend for mobile TV, i.e., mobile social TV based on agile resource supports and rich functionalities of cloud computing services. We introduce a generic and portable mobile social TV framework, that makes use of both an IaaS cloud and a PaaS cloud. The framework provides efficient transcoding services for most platforms under various network conditions and supports for co-viewing experiences through timely chat exchanges among the viewing users. By employing one surrogate VM for each mobile user, we achieve ultimate scalability of the system. Through an in-depth investigation of the power states in commercial 3G cellular networks, we then propose an energy-efficient burst transmission mechanism that can effectively increase the battery lifetime of user devices. We have implemented a realistic prototype of our system, deployed on Amazon EC2 and Google App Engine, where EC2 instances serve as the mobile users' surrogates and GAE as the social cloud to handle the large volumes of social message exchanges. We conducted carefully designed experiments on iPhone 4S platforms. The experimental results prove the superior performance of Social TV, in terms of transcoding efficiency, power saving, timely social interaction, and scalability. The experiments also highlight the drawbacks of the current HTTP Live Streaming protocol implementation on mobile devices as compared to our proposed burst transmission mechanism which achieves a 29.1 % increase of battery lifetime. Much more, however, can be done to enhance the system to have product-level performance. In the current prototype, we do not enable sharing of encoded streams (in the same format/bit rate) among surrogates of different users. In our future work, such sharing can be enabled and carried out in a peer-to-peer fashion, e.g., the surrogate of a newly joined user may fetch the transcoded streams directly from other surrogates, if they are encoded in the format/bit rate that the new user wants.

For implementing social interactions, most BigTable-like data stores (including GAE) support memcache [34] which is a highly efficient secondary storage on the data stores. We seek to integrate memcache support into CloudMoV, by possibly memcaching the data (e.g., chat histories) of sessions where friends chat actively, so as to further improve the query performance. To sustain the portability of the system, we will stick to standard API interfaces, i.e., JCache (JSR 107), in our system.

REFERENCES

[1] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for vm-based cloudlets in mobile computing," IEEE Pervasive Computing, vol. 8, pp. 14–23, 2009.

[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. of 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 INFOCOM'11, 2011, pp. 201–205.

[4] T. Coppens, L. Trappeniners, and M. Godon, "AmigoTV: towards a social TV experience," in Proc. of EuroITV, 2004.

[5] N. Ducheneaut, R. J. Moore, L. Oehlberg, J. D. Thornton, and E. Nickell, "Social TV: Designing for Distributed, Sociable Television Viewing," International Journal of Human-Computer Interaction, vol. 24, no. 2, pp. 136–154, 2008.

[6] A. Carroll and G. Heiser, "An analysis of power consumption in as smartphone," in Proc. of USENIXATC, 2010.

[7] What is 100% Pure Java, 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, pp. 228–247, January 2008.

[9] DVB-H, http://www.dvb-h.org/. [10] K. Chorianopoulos and G. Lekakos, "Introduction to social tv: Enhancing the shared experience with interactive tv," International Journal of Human-

Computer Interaction, vol. 24, no. 2, pp. 113–120, 2008.

[11] M. Chuah, "Reality instant messaging: injecting a dose of reality into online chat," in CHI '03 extended abstracts on Human factors in computing systems, ser. CHI EA '03, 2003, pp. 926–927.

[12] R. Schatz, S. Wagner, S. Egger, and N. Jordan, "Mobile TV becomes Social - Integrating Content with Communications," in Proc. of ITI, 2007.

[13] R. Schatz and S. Egger, "Social Interaction Features for Mobile TV Services," in Proc. of 2008 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting, 2008.

[14] J. Flinn and M. Satyanarayanan, "Energy-aware adaptation for mobile applications," in Proceedings of the seventeenth ACM symposium on Operating systems principles, ser. SOSP '99, 1999, pp. 48–63.

[15] W. Yuan and K. Nahrstedt, "Energy-efficient soft real-time cpu scheduling for mobile multimedia systems," in Proceedings of the nineteenth ACM symposium on Operating systems principles, ser. SOSP '03, 2003, pp. 149–163.

[16] W. Zhang, Y. Wen, Z. Chen, and A. Khisti, "Qoe-driven cache management for http adaptive bit rate (abr) streaming over wireless networks," in Proc. of IEEE Globecom, 2012.