ISSN: 2394-6881

International Journal of Engineering Technology and Management (IJETM)
Available Online at www.ijetm.org
Volume 2 Issue 2 Page No. 42-45

SOCIAL VIDEO STREAMING ON LINUX POWERED SMARTPHONES USING CLOUD SERVICES

Sachin Kumar P. Agwan, Akash V. Jadhav, Amit M. Bhoyar, Abhimanyu Dawane, Rahul Pawar

Computer Engineering Department, Dr. D Y Patil College of Engineering, Pune University, Maharashtra, India

sachinagwan19@gmail.com, akkijadhav002@gmail.com, amitbhoyar143@gmail.com, abicool77@gmail.com, rahulpawar8087@gmail.com

ABSTRACT

The cloud computing is increasing day by day as its advantages overcome the disadvantage of various early computing techniques. Cloud provides online data storage where data is stored. If in case, file deletion or if the cloud gets destroyed due to any reason the data stored at cloud gets lost. Hence to overcome this problem various data recovery techniques have been developed in cloud computing. In this literature paper, we explore some existing techniques that are previously implemented to tackle this problem. The objective of this paper is to give the brief explanation of the existing techniques.

In 21st century there are large android users and these numbers are increasing exponentially. Large numbers of applications are available especially related to social interactions, entertainment field, sport fields etc. But there are lack of combination of video services and chat simultaneously. Users can not chat while watching Video in same application, they can only comment on Video. In this paper we are propagating model of application which can provide social interactions with video with some advanced features. For this purpose we are using Cloud Technology. Tough challenges arise on how to effectively exploit cloud resources to facilitate mobile services. The proposed application provides living room experience at the time of watching video to mobile users that users haven't experienced before. As Cloud Technology is involved the System will effectively use IAAS (Infrastructure as a Service), PAAS (Platform as a Service) and SAAS (Software as a Service) Cloud Services. To guarantee good Streaming Quality here we are using Surrogate for each user. The surrogate performs efficient stream transcending that matches the current connectivity quality of the mobile user. This Application will be helpful to the Educational Institutions.

Keywords: IaaS (Infrastructure as a Service), PaaS (Platform as a Service), SaaS (Software as a Service), VM (Virtual Machine)

INTRODUCTION:

Smartphones nowadays are shipped with multiple microprocessor cores and gigabyte RAMs; they possess more computation power than personal computers of a few years ago. On the other side, 4G broadband cellular infrastructures wide deployment further fuels the trend. In addition to common productivity tasks like emails and web browsing, smartphones are improving their strengths in more challenging scenarios such as realtime streaming of videos and online games, as well as providing as a main tool for social exchanges. It is natural to resort to cloud computing, the newly-emerged computing technologies for low-cost, adoptable, scalable resource supply, to provide power efficient mobile data communication. By using 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 made. The big challenge in front of us is how to effectively exploit cloud services to facilitate mobile applications.

In this paper, we describe the design of a novel mobile social system Social Video Streaming on Linux Powered Smartphones using Cloud Services. 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 Social Video Streaming on Linux Powered Smartphones using Cloud Services, mobile users can import a live or on-demand video to watch from any video site, can also invite their friends to watch the video concurrently, and chat with their friends while watching the video. Thus it provides better viewing experience and social awareness among friends on the go.

We design Social Video Streaming on Linux Powered Smartphones using Cloud Services to seamlessly utilize agile resource support and rich functionalities offered by both an IaaS (Infrastructure-as a-Service) cloud , PaaS

(Platform-as a- Service) cloud and SaaS (Software-as a-Service).

1. PORTABILITY:-

A prototype Social Video Streaming on Linux Powered Smartphones using Cloud Services 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" and uses no platform-dependent or proprietary APIs.

2. SPONTANEOUS SOCAIL INTERACTIVITY:

Multiple mechanisms are included in the design of Social Video Streaming on Linux Powered Smartphones using Cloud.

II. FEATURES:

A. VIDEO RECOMMENDER:

System will notify user via email according to his/her interest. Let us suppose user search for particular video and it is not available at that time. If after sometime it becomes available on Server then System will send notification to the user about that video. Also User will receive notifications about videos according to his/her likes, daily visits, etc.

B. VIDEO INTEGRITY:

Presently there are some web sites that contain same videos but with different name. The System will implement Video Integrity so that there won't be same video with different name. Video Integrity is implemented by frame matching. Threshold Ratio of videos can be measured i.e how much percentage of video is same is measured and decision can be taken whether to upload or not upload the video.

C. VIDEO ACCESSING:

Server Administrator can set video accessibility It includes options like public and private. If video accessibility is public then anyone can see the video while private option is totally opposite to that of public that is only selected users can watch the video.

D. VIDEO SHARING:

Online user can suggest or recommend particular video to his/her friends who are online at that particular instant. Thus, Social Video Streaming on Linux Powered Smartphones using Cloud Services provides social interaction with the video to the user.

III. ARCHITECTURE:

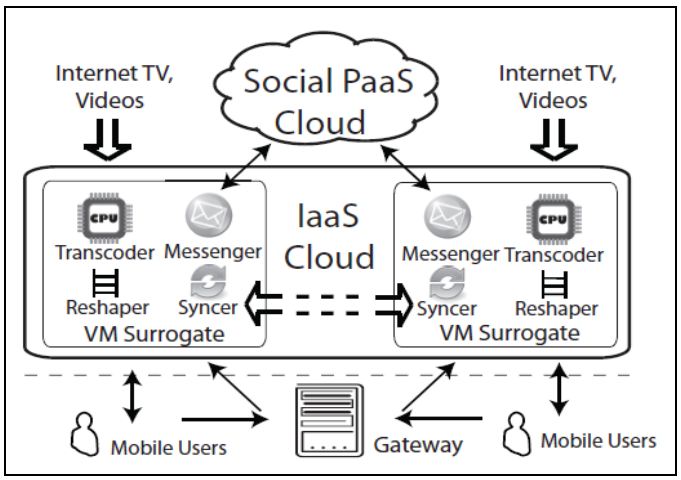

Diagram gives overview of social video streaming on linux powered smartphones using cloud services. As you can see there will be Surrogate also known as Virtual Machine (VM) for each Online User. The Surrogate acts an intermediate between user and video source. There is a gateway server in Social Video Streaming on Linux Powered Smartphones using Cloud Services keeps track of participating users and their VM surrogates, which implementation can be done by standalone server or VMs in the IaaS cloud**.**

A. TRANCSCODER :

There will be Transcoder for each Surrogate. It takes decision about encoding the video stream from source in correct format, bit rate and dimension. Before delivery to the user, the video stream is further compress into a proper transport stream.

B. RESHAPER :

The input to the Reshaper is the encoded transport stream from transcoder. It performs some task on that data like chops it into segments, and then sends each segment in a burst to the mobile device upon its request. By performing this best power efficiency of the device can be achieved. The burst size, i.e., the amount of data in each burst, is carefully decided according to the network speed and capability of device.

C. SOCIAL CLOUD :

D. It is useful to store all the social data in the system. These stored social data include the online statuses of all users, information of the current sessions, and messages in every session. The social cloud is queried from time to time by the VM surrogates.

E. MESSENGER:

It resides in each surrogate in the IaaS Cloud. It is the Client side Component of the Social 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.

F. MOBILE CLIENT:

The Android Device should be compatible with HTML5 and supports the HTTP Live Streaming Protocol. These requirements are mostly satisfied by almost all Android Mobile phones.

IV ALGORITHMS:

1. RC6:

In Cryptography,RC6 stands for Rivest Cipher 6 and it is a symmetric block Cipher. It is preceed by RC 5. The RC 6 was invented by Ron Rivest,Matt Robshaw , Ray Sidney, and Yiqun Lisa Yin. It is designed to meet the requirements of of the Advanced Ecryption Standard (AES) Competition.. Basically RC 6 is an Encryption Algorithm.

RC 6 in proper format has a block size 128 bits and support key sizes 128,192,256 bits.But as it is modified version of RC 5 it can be parameterised to to support a wide variety of word-lengths, key sizes, and number of rounds.RC6 is similar to RC5 using data-dependent rotations, modularaddition, and XOR operations. In RC6 does use an extra multiplication operation not present in RC5 in order to make the rotation dependent on every bit in a word, and not just the least significant few bits.

```
/ Encryption/Decryption with RC6-w/r/b
//
// Input:   Plaintext stored in four w-bit input registers A, B, C & D
//          r is the number of rounds
//          w-bit round keys S[0, ... , 2r + 3]
//
// Output: Ciphertext stored in A, B, C, D
//
// '''Encryption Procedure:'''

B = B + S[0]
D = D + S[1]
for i =1 to r do
{t =(B*(2B +1))<<< lg w
u =(D*(2D +1))<<< lg w
A =((A ⊕ t)<<< u)+ S[2i]
C =((C ⊕ u)<<< t)+ S[2i +1]
(A, B, C, D)=(B, C, D, A)}
A = A + S[2r +2]
C = C + S[2r +3]
// '''Decryption Procedure:'''
C = C - S[2r +3]
A = A - S[2r +2]
for i = r downto 1do
{(A, B, C, D)=(D, A, B, C)
u =(D*(2D +1))<<< lg w
t =(B*(2B +1))<<< lg w
C =((C - S[2i +1])>>> t)⊕ u
A =((A - S[2i])>>> u)⊕ t}
D = D - S[1]
B = B - S[0]
```

2. SPEKE :

The simple password exponential key exchange (SPEKE) has two stages. The first stage uses a DH exchange to establish a shared key K, but instead of the commonly used fixed primitive base g, a function *f* converts the password S into a base for exponentiation. The rest of the first stage is pure Diffie-Hellman, where Alice and Bob start out by choosing two random number RA and RB:

S1. Alice computes: QA = *f*(S)RA *mod* p, A®B: QA.
S2. Bob computes: QB = *f*(S)RB *mod* p, B®A: QB.
S3. Alice computes: K = *h*(QBRA *mod* p)
S4. Bob computes: K = *h*(QARB *mod* p)

In the second stage of SPEKE, both Alice and Bob confirm each other's knowledge of K before proceeding to use it as a session key. One way is:

S5. Alice chooses random CA, A®B: *E*K(CA).
S6. Bob chooses random CB, B®A: *E*K(CB, CA).
S7. Alice verifies that CA is correct, A®B: *E*K(CB).
S8. Bob verifies that CB is correct.

To prevent discrete log computations, which can result in the attacks the value of p-1 must have a large prime factor q.

The function *f* is chosen in SPEKE to create a base of large prime order. This is different than the commonly used primitive base for DH. The use of a prime-order group may also be of theoretical importance.

Other variations of the verification stage are possible. QB is used instead of the random numbers C*x* in an minimal three-message refinement apply to SPEKE.

More generally, verification of K can use any classical method, since K is cryptographically large. This example repeatedly uses a one-way hash function:

S5. Alice sends proof of K: A®B: $h(h(K))$

S6. Bob verifies $h(h(K))$ is correct, B®A: $h(K)$

S7. Alice verifies $h(K))$ is correct.

This approach uses K in place of explicit random nonces, which is possible since K was built with random information from both sides.

Table 1: Technical feasibility

Programming Language	JavaScript, PHP
Hardware	Android Smartphone
Front End Programming	JAVA, .NET
Technology	Wireless Network, Sales Force.

V. CONCLUSION

The Project will go to implement highly social interactions. Users can interact with video as well as simultaneously they can chat with other user's who are online at that instant.

Special Features like video integrity, video recomendeter etc are also provided in the system introduced in Project. It is potentially provide an ideal platform to support the desired mobile services.

In the following diagram login page is shown as below:

In this way the Project effectively uses Cloud platform to provide required services i.e. chatting along with the watch watching feature. Project results are useful at College level, industry level etc.

VI. REFERENCES:

1. CloudMoV: Cloud-based Mobile Social TV ,2013,Yu Wu , Zhizhong Zhang, Chuan Wu, Zongpeng Li, Francis C.M. Lau Department of Computer Science,The University of Hong Kong.
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. http://en.wikipedia.org/wiki/A*_search_algorithm
7. http://en.wikipedia.org/wiki/XOR_cipher