High Quality Video Sharing In Cloud

[1]Dr.S.Thangam, [2]P.K.Mounekaa, [3]C.Jenny priya

[1]Associate Professor, CACET [2, 3] Student. CACET

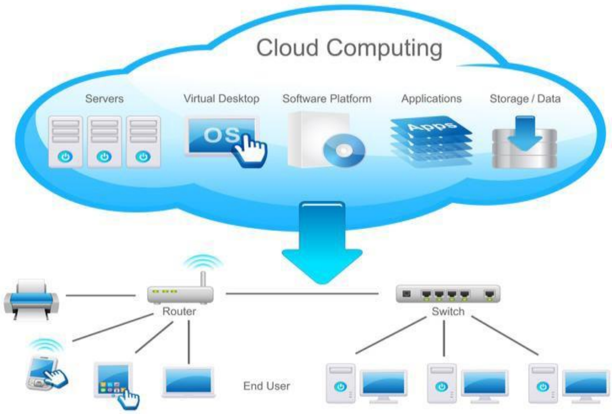
Abstract— *The multiplications of advanced mobile phones and tablets have fuelled another influx of interest for portable administrations. Today's clients expect universal system associations, as well as look for a rich media encounter wherever they go and at whatever point they require. It is difficult to manage acceptable nature of experience over intrinsically questionable remote connections. Extensive scale, synchronous conveyance of media substance tends to overwhelm existing portable system framework. In addition, a principal hole exists between the asset requests of computationally costly interactive media applications and the abilities managed by battery-worked cell phones. Late advances in distributed computing help to scaffold this hole. By turning to the cloud as a repository for extra calculation and capacity, portable media administrations can scale better with client request in a flexible and element way, at generally low expenses. Cloud asset provisioning and booking, media content administration, nature of-administration (QoS), client portability, security, and so forth. The framework viably uses both PaaS (Platform-as-a-Service) and IaaS (Infrastructure- as-a-Service) cloud administrations to offer the parlor experience of video viewing to a gathering of different versatile clients who can collaborate socially while sharing the video.*

I. Introduction

Cloud Computing

Cloud computing is a technology that uses the internet and central remote servers to maintain data and applications. Cloud computing allows consumers and businesses to use applications without installation and access their personal files at any computer with internet access. This technology allows for much more efficient computing by centralizing storage, memory, processing and bandwidth. Cloud computing is a comprehensive solution that delivers IT as a service.

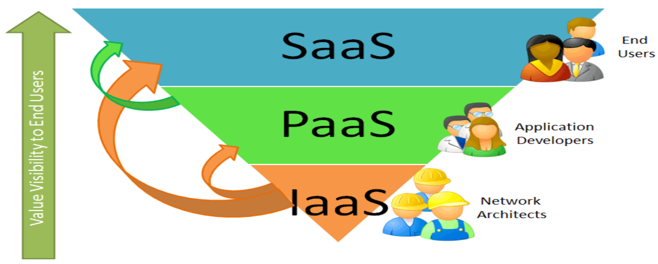
The flexibility of cloud computing is a function of the allocation of resources on demand. Before cloud computing, websites and server-based applications were executed on a specific system. Cloud computing is broken down into three segments application, storage and connectivity.



Cloud Computing Models

Service model:

There are three fundamental Service models in Cloud computing. Three service models are explained below.



Software as a Service:

Cloud Applications or Software as a Service (SaaS) refers to software delivered over a browser. SaaS eliminates the need to install and run applications on the customer's own computers/servers and simplifies maintenance, upgrades and support. Examples of SaaS are Face book, Sales Force, etc.

Platform as a Service:

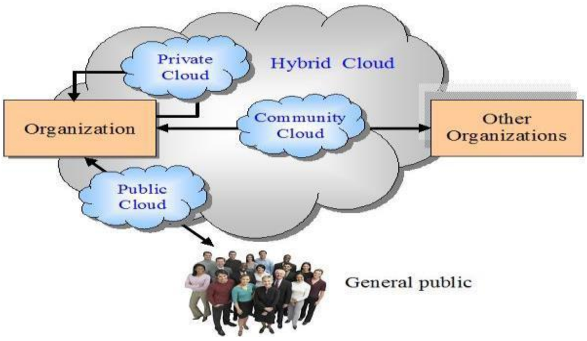
Cloud platform services or Platform as a Service (PaaS) refers to an environment for software development, storage and hosting delivered as-a-service over the Internet.

Examples of PaaS are Google App Engine, Force.com, Microsoft Azure, WOLF, etc.

Infrastructure as a Service:

Cloud infrastructure services or Infrastructure as a Service (IaaS) delivers a computing infrastructure, typically a virtualization environment, as-a-service. Examples of IaaS are virtual servers leased by Amazon, Rack space, Go Grid, etc.

Deployment model:



Each company chooses a deployment model for a cloud computing solution based on their specific business, operational, and technical requirements. Four primary cloud deployment models are private cloud, community cloud, public cloud, and hybrid cloud.

II. VIDEO SHARING

We describe the design of a novel social TV system, CloudMoV, 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 CloudMoV, 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, 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, it remains a challenge to achieve mobile social TV, where the concurrently viewing experience with friends is enabled on mobile devices. We design CloudMoV to seamlessly utilize agile 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, 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 video platforms, if not only to the current hottest models. CloudMoV customizes the streams for different devices at real time, by offloading the transcoding tasks to an IaaS cloud. In particular, we novelty 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.

III. PROBLEM STATEMENT

Various portable TV frameworks have sprung up lately, determined by both equipment and programming progresses in cell phones. Some early frameworks convey the lounge room understanding to little screens moving. In any case, they concentrate more on obstruction leeway to understand the union of the broadcasting company and the portable system, than investigating the request of "social" collaborations among versatile clients. Albeit numerous versatile social or media applications have developed, genuinely executioner ones increasing mass acknowledgment are as yet blocked by the constraints of the present portable and remote advances, among which battery lifetime and precarious association transmission capacity are the most troublesome ones.

A. *Energy-Efficient Community Cloud for Real-Time Stream Mining*

Propose to employ a cloud-based stream mining system in which the mobile devices send via wireless links unclassified media streams to the cloud for classification. We focus on minimizing the classification-energy cost, defined as an affine combination of classification cost and energy consumption at the cloud, subject to an average stream mining delay constraint (which is important in real-time applications). To address the challenge of time-varying wireless channel conditions without a priori information about the channel statistics, we develop an online algorithm in which the cloud operator can adjust its resource provisioning on the fly and the mobile devices can adapt their transmission rates to the instantaneous channel conditions. It is proved that, at the expense of increasing the average stream mining delay, the online algorithm achieves a classification-energy cost that can be pushed arbitrarily close to the minimum cost achieved by the optimal offline algorithm. Extensive simulations are conducted to validate the analysis.

Merits: Satisfying the average power constraint for each user.

Demerits: Cloud-based mobile stream mining systems operating in random environments with unknown distributions.

B. *Towards an Elastic Application Model for Augmenting Computing Capabilities of Mobile Platforms*

We propose a new elastic application model that enables the seamless and transparent use of cloud resources to augment the capability of resource constrained mobile devices. The salient features of this model include the partition of a single application into multiple components called web lets, and a dynamic adaptation of web let execution configuration. While a web let can be platform independent (e.g., Java or .Net byte code or Python script) or platform dependent (native code), its execution location is transparent – it can be run on a mobile device or migrated to the cloud, i.e., run on one or more nodes offered by an IaaS provider. Thus, an elastic application can augment the capabilities of a mobile device including computation power, storage, and network bandwidth, with the light of dynamic execution configuration according to device's status including CPU load, memory, battery level, network connection quality, and user preferences. This paper presents the motivations, concepts, typical elasticity patterns, and cost consideration of elastic applications. We validate the augmentation capabilities with an implemented reference architecture and example applications.

Merits: It can be run on a mobile device or migrated to the cloud, i.e., run on one or more nodes.

Demerits: Management of heterogeneous computing environments, data management, Communication dependencies between web lets.

C. ThinkAir: Dynamic resource allocation and parallel execution in cloud for mobile code offloading

we propose ThinkAir, a framework that makes it simple for developers to migrate their smartphone applications to the cloud. ThinkAir exploits the concept of smartphone virtualization in the cloud and provides method-level computation offloading. Advancing on previous work, it focuses on the elasticity and scalability of the cloud and enhances the power of mobile cloud computing by parallelizing method execution using multiple virtual machine (VM) images. We implement ThinkAir and evaluate it with a range of benchmarks starting from simple micro-benchmarks to more complex applications. First, we show that the execution time and energy consumption decrease two orders of magnitude for a N-queens puzzle application and one order of magnitude for a face detection and a virus scan application. We then show that a parallelizable application can invoke multiple VMs to execute in the cloud in a seamless and on-demand manner such as to achieve greater reduction on execution time and energy consumption. We finally use a memory hungry image combiner tool to demonstrate that applications can dynamically request VMs with more computational power in order to meet their computational requirements.

Merits: Applications can dynamically request VMs with more computational power in order to meet their computational requirements.

Demerits: parallelization reduces the execution time and energy consumption of these applications with significant margins when compared to non-parallel executions of the same.

D. Energy-Efficient Soft Real-Time CPU Scheduling for Mobile Multimedia Systems

This paper presents GRACE-OS, an energy-efficient soft real-time CPU scheduler for mobile devices that primarily run multimedia applications. The major goal of GRACE-OS is to support application quality of service and save energy. To achieve this goal, GRACE-OS integrates dynamic voltage scaling into soft real-time scheduling and decides how fast to execute applications in addition to when and how long to execute them. GRACE-OS makes such scheduling decisions based on the probability distribution of application cycle demands, and obtains the demand distribution via online profiling and estimation. We have implemented GRACE-OS in the Linux kernel and evaluated it on an HP laptop with a variable-speed CPU and multimedia codec's. Our experimental results show that (1) the demand distribution of the studied codec's is stable or changes smoothly. This stability implies that it is feasible to perform stochastic scheduling and voltage scaling with low overhead; (2) GRACE-OS delivers soft performance guarantees by bounding the deadline miss ratio under application-specific requirements; and (3) GRACE-OS reduces CPU idle time and spends more busy time in lower-power speeds. Our measurement indicates that compared to deterministic scheduling and voltage scaling, GRACE-OS saves energy by 7% to 72% while delivering statistical performance guarantees.

Merits: Reduces GRACE-OS idle time and spends more busy time in lower-power speeds.

Demerits: Limitation of experiments due to lack of diversity of application classes.

E. Energy-aware adaptation for mobile applications

In this paper, we demonstrate that a collaborative relationship between the operating system and applications can be used to meet user-specified goals for battery duration. We first show how applications can dynamically modify their behavior to conserve energy. We then show how the Linux operating system can guide such adaptation to yield a batterylife of desired duration. By monitoring energy supply and demand, it is able to select the correct tradeoff between energy conservation and application quality. Our evaluation shows that this approach can meet goals that extend battery life by as much as 30%.

Merits: Show applications can dynamically modify their behavior to conserve energy.

Demerits: Performance degradation.

IV. PROPOSED SOLUTION

We propose the outline of a Cloud-based, novel Mobile social TV framework. The framework adequately uses both PaaS (Platform-as-a-Service) and IaaS (Infrastructure-as-a-Service) cloud administrations to offer the front room understanding of video viewing to a gathering of different portable clients who can cooperate socially while sharing the video. To ensure great gushing quality as experienced by the versatile clients with time differing remote network, we utilize a surrogate for every client in the IaaS cloud for video downloading and social trades for the benefit of the client.

Encoding flexibility:

Different mobile devices have differently sized displays, customized playback hardware's, and various codec's. 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. CloudMoV customizes the streams for different devices at real time, by offloading the transcoding tasks to an IaaS cloud.

Battery efficiency:

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:

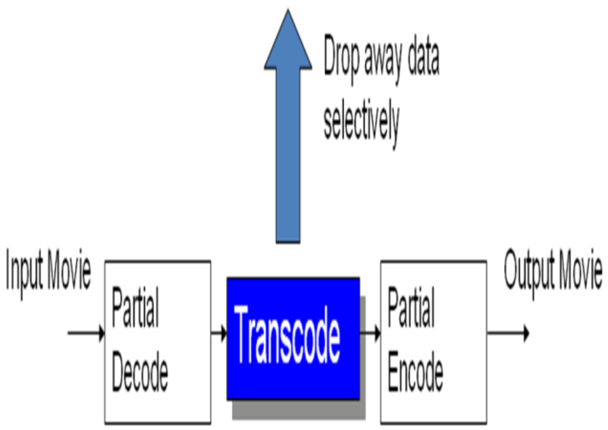
Multiple mechanisms are included in the design of CloudMoV to enable spontaneous social, co-viewing experience.

Portability:

A prototype CloudMoV system is implemented following the philosophy of "Write Once, Run anywhere" (WORA): both the front-end mobile modules and the backend server modules are platforms implemented in "100% Pure Java" Our prototype can be readily migrated to various cloud and mobile with little effort.

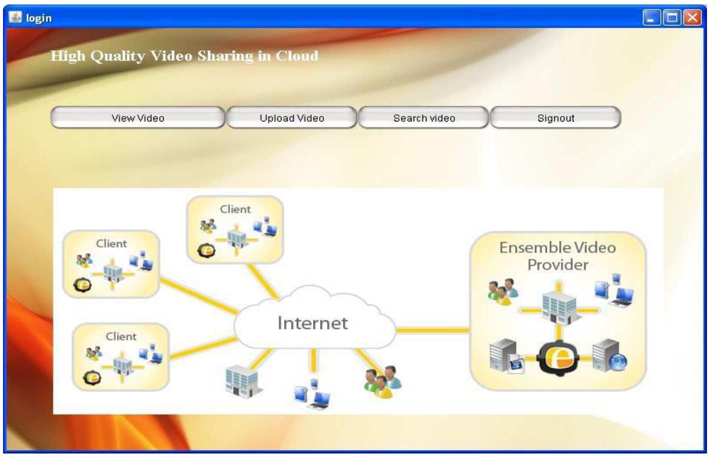
Transcoding mechanism

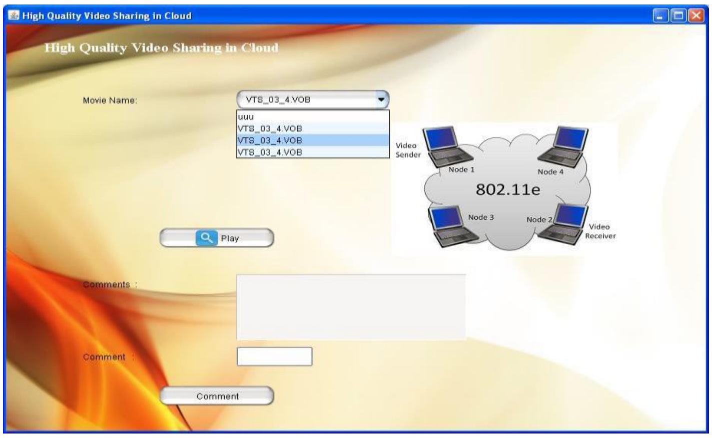
It resides in each surrogate, and is responsible for dynamically deciding how to encode the video stream from the video source in the appropriate format, dimension, and bit rate. Before delivery to the user, the video stream is further encapsulated into a proper transport stream. Each video is exported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

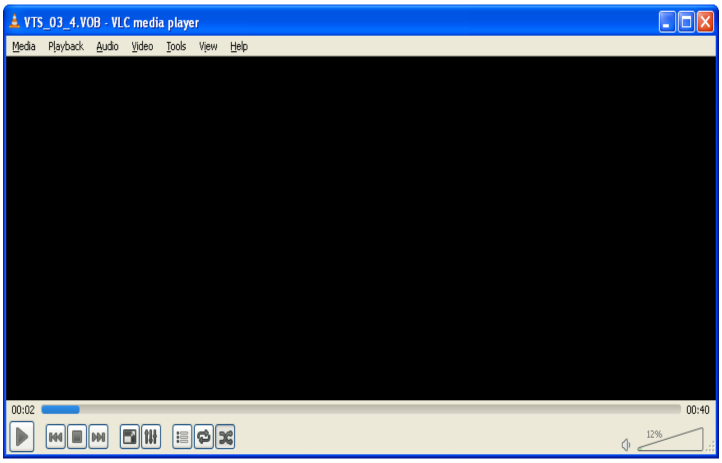


- Only one high quality compressed video is stored
- No/Much less computations on motion estimation
- Can produce comparable video quality with direct encoding

V. OUTPUT RESULT







CONCLUSION

We conclude that the results prove the superior performance of CloudMoV, in terms of transcoding efficiency, timely social interaction, and scalability. In CloudMoV, 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

[1] Anastasi. G, M. Conti, E. Gregori, and A. Passarella, "Saving energy in wi-fi hotspots through 802.11 psm: an analytical model," in Proceedings of the Workshop on Linguistic Theory and Grammar Implementation, ESSLLI-2000, 2004, pp. 24–26.

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

[3] Chorianopoulos. K and G. Lekakos, "Introduction to social tv: Enhancingthe shared experience with interactive tv," International Journal of Human- Computer Interaction, vol. 24, no. 2, pp. 113–120, 2008.

[4] Ducheneaut.N, 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.

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

[6] Flinn.J 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.

[7] Jason 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.