ISSN:0975-9646



# Cloud-Based Multimedia Storage System with QoS Provision

Swati Vani, Bhagyashri Bhosale, Ganesh Shinde, Rajni Shinde, Prof.Manoj Pawar.

*Department of Information Technology*
*Marathwada Mitra Mandal's Institute of Technology, Lohgaon, Pune-411043, India*

***Abstract*—Recently there are new trends in the way we use computers and access networks due to advanced mobile devices and network technologies. One of trend is cloud computing where resources are stored and processed on network. The other is Mobile computing, where mobile devices such as smart phones and tablets combines network connectivity, mobility, and software functionality and working as personal computers. Cloud based multimedia services have high constraint in terms of bandwidth and jitter. Therefore different approaches required to manage resources more efficiently for better Quality of Service (QoS) and Quality of Experience (QoE) offered by the mobile media services. This paper introduces a novel concept of Mobile Multimedia Web Service using Cloud in which services will run on public cloud depending upon service demands and network status, the service will be populated on other public cloud in different geographical locations. If demand for particular service increases in a location it will be more reliable to populate that service[1] closer to the cloud in that location. This will prevent the high traffic loads on internet backbone due to streaming of multimedia data. It will offer service provider's management mechanism and an automated resource allocation for their services. This will help to reduce bandwidth and jitter on the cloud based multimedia services**.

***IndexTerms*—Computer Network Management, Communication System, Web Services, Mobile Computing.**

## I. INTRODUCTION

Cloud computing is one of the trends in IT which refers to application and services that run on distributed network using virtualized resources and accessing by common Internet protocols and networking standards. There are three categories of cloud services: Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). SaaS delivers software application over the internet. Google Apps (includes Google Mail, Docs, Sites, Calendar, etc) is the example of SaaS. PaaS delivers a host operating system and development tools which installed virtualized resources. The example of PaaS is Google App Engine which provides elastic platform for Java and Python applications. IaaS offers number of virtual machines or processors and storage space and leaves it up to the user to select how these resources are used. Amazon EC2[3] (example of IaaS) are probably most known and widely used. Amazon EC2 provides an instance of a virtual machine image that allows full control over the operating system. It is possible to select a suitable operating system, and platform (32 and 64 bit) from many available Amazon Machine Images (AMI) and several possible virtual machines, which differ in CPU power, memory and disk space. This functionality allows freely select suitable technologies for any particular task. In case of Amazon EC2 price for the service depends on machine size, its up time, and used bandwidth in and out of the cloud.

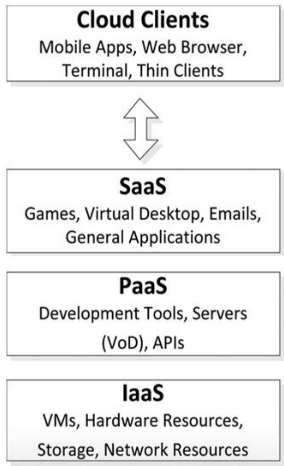


Fig. 1. CLOUD SERVICE LAYERS

## II. ART OF CLOUD COMPUTING

**Amazon Elastic Compute Cloud (EC2):** is a central part of Amazon.com's cloud computing platform, Amazon Web Services (AWS). EC2 allows users to rent virtual computers on which to run their own computer applications. EC2 allows scalable deployment of applications by providing a Web service through which a user can boot an Amazon Machine Image to create a virtual machine, which Amazon calls an 'INSTANCE', containing any software desired.

**iCloud[2]:** The service allows users to store data such as music and iOS applications on remote computer servers for download to multiple devices such as iOS-based devices running iOS. The service also allows users to wirelessly back up their iOS devices to iCloud instead of manually doing so using iTunes.

**Office 365:** It is a subscription-based online office and software plus services suite which offers access to various services and software built around the Microsoft Office platform.

## III. SYSTEM IMPLEMENTATION AND MECHANISM

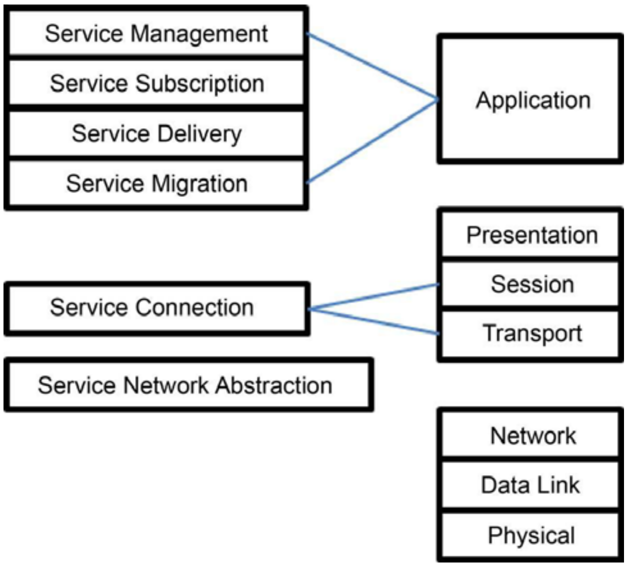


Fig. 2. SERVICE DELIVERY FRAMEWORK

Cloud based service Layered Framework: We relate the layers of the architecture with the OSI model. The proposed framework and the OSI model share the same level of abstraction in terms of network technologies and protocols and this makes it easy to use the OSI as a reference to our model as opposed to using the TCP/IP model.

The service architecture is not meant to map directly to some of the OSI layers. Some of the functions performed in the proposed layers can interact with OSI layers to perform network-level operations while other layers do not present any functions that directly interface with the OSI and are therefore considered extra layers.

**The Service Management Layer (SML):** Deals with how services are registered in a Cloud. This also includes the overall Service and Security Level Agreement (SSLA) between the Cloud providers and the service providers and the unique Service ID. The SML can be considered as part of the Application Layer in the OSI since it defines the applications themselves and how they use resources.

**The Service Subscription Layer (SSL):** Deals with the subscription of clients to the service and holds information that handles the subscriptions such as User IDs, the list of services subscribed to by individual client and the associated client SLAs between clients and services. This layer can give instructions to the Presentation Layer in the OSI in order to handle user specific service parameters such as encryption or CODECs in video streams. The SSL can be considered as part of the Application Layer in the OSI.

**The Service Delivery Layer (SDL):** Is responsible for the delivery of services to individual clients. The layers below receive instructions from this layer with regard to connecting to individual clients as well as populating Clouds.

**The Service Migration Layer (SMiL):** Is responsible for the Migration of services between Clouds. It deals with resource allocation across Clouds to facilitate service population. It also holds the mechanism that performs the handover of client connections between services. The SSL can be considered as part of the Application Layer in the OSI.

**The Service Connection Layer (SCL):** Monitors connections between clients and services. Some of this layer's functions map directly to the Session Layer in the OSI model.

**Service Network Abstraction Layer (SNAL):** Makes the network technology transparent to the upper layers in order to simplify and unify the process of migration. The function of this layer is to act as a common interface between the service delivery framework and the underlying network architecture such as IP overlay network [6] or new technologies which divide the Internet into a Core network surrounded by Peripheral wireless networks.

**Abstraction of service layer:** In SML when a service provider wishes to publish a service, they have to define security and QoS parameters [4]. In SDL, the logic that processes all the data regarding QoS characteristics and user mobility resides in this layer. It uses data from the overall SSLA and the client SLA and checks if the requirements are met by using network QoS data given by the layer below. Such data can be fed to this layer by the mobile devices themselves either in the form of a process running separately or through a QoS-aware protocol that can report latency and bandwidth between two end points. The Cloud that fulfills all the parameters in the SSLA list and can provide better QoS than the others can then proceed to the Migration process in the layer below. In SCL the SCL is also responsible for the network handover between clients and services after a service moves. This is done by gathering QoS data from the network and from client devices.

**Implementation mechanism:** In order to gather QoS data and know the network conditions in a specific area, we are using another mechanism that we call the QoS Monitor. It is considered to be part of the SCL and acquires such data by querying the clients for network conditions. The mechanism that we are assuming here that can resolve human-friendly service names to unique Service IDs. In the SDL we need mechanisms that will connect service subscribers to the correct instance of a service for service delivery purposes. A record of Service IDs and in which Clouds their instances are running and also uses input by the QoS Tracking are maintained by the Service Tracking and Resolution or STAR. STAR will make a decision on which Cloud is better suited to service a client request based on the location of the client, using this information.

STAR achieve this functionality is by look up routing tables in order to identify which Cloud is closer to a user. Service delivery mechanism using STAR is shown in fig.2 Service to reject the new client and forward them to another Cloud if possible. This gives control to service providers and also becomes a contingency mechanism in case STAR makes a wrong decision. The STAR server can be scaled similarly to the DNS[5] system since it is essentially the same type of service albeit with some extra parameters. Once a Cloud ID is found, then the ID is resolved into the IP addresses of the Cloud controllers that the client can contact to access the service. The process is shown in the Fig. 3. It should be noted that alternatively the Cloud ID can be returned to the client, at which point, the client will have a choice of which DNS to use to find the IP addresses.

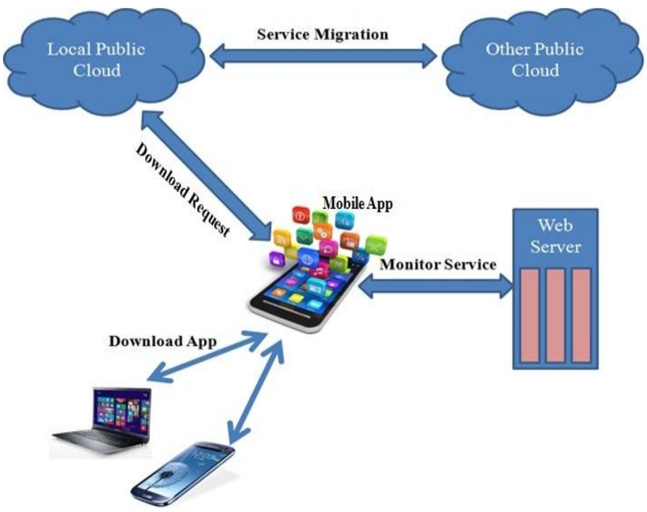


Fig. 3. SYSTEM ARCHITECTURE

## IV. MATHEMATICAL MODEL

We start by defining the time to prefetch blocks of data, which is given by:

$T_{Prefetch}=L+C*p$

In this equation, L is the network latency and C is the per block time of copying data between the in-cache memory and network buffers. Ideally should be at least equal to the number of blocks required to display a video frame of data. On a lightly loaded wired network we can consider these values constant for each link. However, in a mobile environment, changes as the client moves and the number of network links increase. We can express L as follows:

$L=F_{n,s,\theta}+F_{cloud}+F_{Protocol}$

Where, ($F_{n,s,\theta}$) is the latency incurred by the number of links(n) between client and service, the network bandwidth on each link ($S_i$) and the network load on each link ($\theta_i$), $F_{cloud}$ is the Cloud latency caused by the network topology and hierarchy within the Cloud $F_{protocol}$ is the latency caused by the transport protocol.

If the time to prefetch blocks is larger than the time it takes for the device to consume them, then we have jitter. This can be expressed as:

$T_{prefetch}(p)\geq T_{cpu}*p$

Where ($T_{cpu}$) the time it takes for a device to consume a number of blocks by playing them as audio and video frames. ($T_{cpu}$) is therefore dependent on the type of video being displayed and the hardware capabilities of the mobile device. We now substitute for Tprefetch in (3) with the expressions in (1) and (2). Rearranging, we get:

$F_{n,s,\theta}+F_{cloud}+F_{Protocol}\geq(T_{cpu}-C)*p$

Exploring network latency in detail, for each link we have transmission delay and queuing delay. Therefore, the total network latency will be the sum of the latencies for each link between client and service. Hence, we can express as:

$F_{n,s,\theta}=\sum(D_{ti}+Q_i)$

If we denote the transport block size as b, then the time to transmit p blocks over a link is equal to the number of blocks multiplied by the block size and divided by the bandwidth of the link.

$F_{n,s,\theta}=\sum((p*b)/S_i+Q_i)$

So, we have,

$F_{cloud}+F_{Protocol}+\sum((p*b)/S_i+Q_i)\geq (T_{cpu}-C)*p$

On a lightly loaded system, we consider $F_{protocol}$, $F_{cloud}$ and $Q_i$ to be negligible.

$\sum(b/S_i) )\geq (T_{cpu}-C)$

Let be the soft limit that we are aiming for in order to prevent jitter and $S_L$ is the migration time.

$H_L-S_L=a_tM_t$

Where $a_l$ is the rate of network latency increase as the number of network links increases. We can calculate $a_l$ at the mobile device and we can also find $M_t$ between two Clouds. $H_L$ is given by the mobile device, so we can calculate to $S_L$ find where to set out QoS trigger for service migration.

We can visualized how the increasing number of links between a user and a service can bring the connection near the QoS limit and how we can use a soft limit to trigger service migration in order to prevent this. We can also see that for a given migration time, we need to adjust $S_L$ so that during the migration the QoS will not reach the $H_L$.

## V. APPLICATIONS

From a computational perspective, Cloud providers can share their resources with other providers. This gives them the flexibility to request additional resource when their Cloud needs them or rent some of their resources to other providers that need them.

By taking into account multimedia creation services such as rendering, we can see how such a scenario is applicable and how it can benefit clients and providers alike. Furthermore, if we combine the above scenario with mobile devices, we can see how in the future we may find ourselves in a position where rendering is done on the Cloud and the mobile devices only display the content.

This can occur in applications such as games. In these situations, the proposed framework will not only balance the rendering load on Clouds but will also relieve networks from the high traffic generated by streaming video and audio. The distance reduction between clients and services caused by migrations will also decrease the latency and give users a more interactive feel to their multimedia application, thus improving the QoE.

## VI. CONCLUSION AND FUTURE SCOPE

In this paper, we discuss the challenges which are faced by the mobile user in future networks. The service delivery models which are used currently are not that much sufficient and not consider the needs of mobile user in future.

A cloud storage system was proposed in order to provide robust, scalable, highly available and load-balanced services. In the meantime, the system also needs to provide quality of service provision for multimedia applications and services.

The proposed system achieves the three functions of a multimedia-aware cloud: 1) QoS supporting and provisioning, 2) Parallel processing in distributed environment, 3) QoS adaptation. These functions make the proposed system especially suitable to the video on demand service. it often provides different service quality to users with various types of devices and network bandwidth.

We believe that our implementation will provide the better quality of service (QoS) as well as better quality of experience (QoE) to the user.

## REFERENCES

[1] Fragkiskos Sardis, Glenford Mapp, Jonathan Loo, Mahdi Aiash, Member, IEEE, and Alexey Vinel, Senior Member, IEEE On the Investigation of Cloud-Based Mobile Media Environments With Service-Populating and QoS-Aware Mechanisms IEEE TRANSACTIONS ON MULTIME-DIA, VOL. 15, NO. 4, JUNE 2013

[2] Apple, 2012. iCloud Feb. 15, 2012. [Online]. Available: http://www.apple.com/icloud

[3] Amazon, 2012, EC2, Feb. 28, 2012. [Online]. Available: http://aws.amazon.com/ec2/.

[4] D. Gupta, S. Lee, M. Vrable, S. Savage, A. C. Snoeren, G. Varghese, G. M. Voelker, and A. Vahdat, Difference engine: Harnessing memory redundancy in virtual machines,? in Proc. OSDI, 2008.

[5] W. Zhu, C. Luo, J. Wang, and S. Li, Multimedia cloud computing, IEEE Signal Process. Mag., vol. 28, no. 3, pp. 5969, May 2011.

[6] T. Brisko, RFC 1794, DNS Support for Load Balancing, IETF, 1995.

[7] Y-Comm Research, Middlesex Univ., Mar. 2, 2012. [Online]. Available: http://www.mdx.ac.uk/research/areas/software/ycomm,e search:aspx: