CROSS PLATFORM ONLINE GAMING AS A SERVICE: A CLOUD COMPUTING PERSPECTIVE

Bello Ibrahim Kangiwa
Department of computer Science, Federal College of Education (Technical) Bichi, Kano
kangiwabello4u@yahoo.com

Abstract

Cloud computing is a buzz word in recent years, which provides a paradigm shift from computation as a product to computation as a service. Instead of purchasing costly hardware and software which require installation, maintenance and configuration, cloud computing provides the usage of cloud application and infrastructures based on pay per usage scheme. In the emerging markets with large population, entrepreneurs, government and universities do not have mass computation power to run their applications and to meet the changing requirements. Using cloud computing helps to alleviate these problems. In computer game playing, variation in Computing platform limits the player’s interest in it because they might be interested in playing a game which is only available in different platforms meaning that they have no other option other than to purchase a new device. With cross platform online games deployed to the cloud, one can play with multiple people using any available device and sharing the same game license (user no longer has to pay for license). However, one limitation of the service includes blurry images due to inefficient conversion of graphics from one platform to another. This paper presents a design of cross platform games as a cloud computing service.

Keywords: cloud computing, cross platform, online game.

Introduction

Cloud computing is a networking solution in which everything from computing power to computing infrastructure, applications, business processes to personal collaboration — can be delivered to you as a service wherever and whenever you need them. The cloud refers to the set of hardware, software, networks, storage, services, and interfaces that combine to deliver aspects of computing as a service. Cloud computing makes it possible to use an application, store information, communicate, and access that information from any Internet-connected point and device, using an interface that is operated remotely. No software or hardware on the individual’s computer is needed to operate the software. Instead, a network of computers and/or database storage systems, or some other infrastructure combination both outside of and
fully connected to the Web, operates the software and stores user information. The only thing the user needs is Internet access and a Web browser, and the cloud’s network handles the rest. Cloud computing applications offer a number of advantages to the modern business: streamlined functionality across all departments, the facilitation of interdepartmental communication, and real-time visibility of metrics. Additionally, since all applications that are available in the cloud are fully customizable, they can be tailored to a company’s specifications, thereby further improving productivity and performance (Federico E., 2011). One of the most attractive benefits to cloud-based applications is their mobility. Without the need for servers, hardware and software installation, users can access data from any location that boasts of a Web browser and an Internet connection – whether that location is the office, the home, or on the road. As technology migrates from the traditional on-premise model to the new cloud model, service offerings evolve almost daily. The new cloud model has made it possible to deliver such new capabilities to new markets via web browsers. Web service offerings often have a number of common characteristics, such as a low barrier to entry, where services are offered specifically for consumers and small business entities (John W.R., 2010). The cloud computing service models mostly used include;

• **IAAS (Infrastructure As A Service):** An organization or consumer does not need to own servers, hardware, or network capacity to operate the necessary service. A client, such as an educational institution or an individual consumer, would access the infrastructure and pay on a per-access basis. Examples are Amazon’s Elastic Cloud Computing (EC2) and go grid.

• **PAAS (Platform As A Service):** The organization or individual does not need to own the operating system or necessary supporting software to use the application. Rather, operating systems and associated services are delivered over the Internet without downloads or installation. Examples are Google App Engine and Heroku.

• **SAAS (Software As A Service):** In this software distribution model, applications are provided by the cloud service host and made accessible via the Internet. An example is when a higher education institution would be providing access to Gmail, instead of having the institutions’ email system for students.

The deployment of these services is targeted to both the public, private, community or hybrid based cloud users. The economies of scale, no capital expenditures, and on-demand services are the basic reasons why the world is now moving towards Cloud computing. Moreover, as technology grew over the years, game playing became a very popular activity especially to people of ages 12 to 50 who are involved in the growing activity.

1. **Cross Platform Games**
   The gaming industry has come a long way since its humble beginning more than thirty years ago. From a time when people were thrilled to see a square white block and two rectangular
paddles on the screen to today, where gamers explore realistic three-dimensional worlds in high visual resolution with sound, the experience of being a gamer has changed radically. Zaheer H. et al (2009), explained that Games are objects which consist of components and rules and have certain criteria: rules, a goal, always changing course; chance; competition; common experience; equality; freedom; activity; diving into the world of the game; and no impact on reality. Game playing went from video games to online gaming which is now the most popular type of game playing.

A platform is a combination of hardware and software used to run software applications. A platform can be described simply as an operating system or computer architecture, or it could be the combination of both. Video games vary across platforms. Examples of these platforms include personal computers and video game consoles. Nowadays these consoles are made to operate like computers, providing them with storage devices (hard disk), Memory, input and output slots, ports for networking and more. This technology has allowed various game users to connect to a network and play. In computing, cross-platform, or multi-platform, is an attribute conferred to computer software or computing methods and concepts that are implemented and inter-operate on multiple computer platforms. This follows that a cross platform game is an online game played over some form of computer network on different computing platforms. Some people see cross platform gaming as playing a game which presents different vertical slices of the same environment on different platforms, all interconnected into a wider gaming experience. With this, it means that different parts of the game can be played on different devices. The expansion of online gaming has reflected the overall expansion of computer networks from small local networks to the Internet and the growth of Internet access itself. Online games can range from simple text based games to games incorporating complex graphics and virtual worlds populated by many players simultaneously. One service that is bringing cross platform games to the general public is OnLive. Although this is a cloud gaming platform that delivers a game from a server real-time through the Internet, the end result is the same as what cross platform gaming is trying to achieve. You can now theoretically play the games that are available through OnLive on every Internet connected device with video, including phones and TVs.

2. Cross Platform Game Service Architecture

This is the physical design of an individual service that encompasses all the resources used by a service. Most Online Games are deployed using a client–server system architecture. They interact using what is known as client software to gain access to the entire playing world. As more people venture into gaming experience a service provider will have to scale up to meet the demands of the users, such situations have been observed in massively multiplayer online role playing game (Sergio C., 2002). Also, due to platform restriction, players are restricted to using one server/ one game/ one-time client software. This restriction actually has led to a decline in the number of users of such services especially when the game itself is less entertaining.
Considering these limitations and in a bid to make cross platform game possible a model is proposed which will provide the following improvements:

a. Using a distributed architecture (cloud) at the user level, the game can be accessed as simple client-server architecture.

b. The Game Service Providers (servers) provide services through the cloud which is accessed by customer through the standard platform provided by the cloud.

c. Cross platform software allows user to access games which are played by multiple players from different platforms. This software is said to be a middleware which will allow interoperability between platforms and it is to be paid for by the client.

d. Connections are better, irrespective of the platform one is using because game service providers do not have to worry about scaling up since they are well provided with large enough data centers to accommodate their users via cloud.

e. Users can now access the services directly from a cloud provider which will present multiple options (of games) to users, since the cloud has succeeded in enabling collaboration and acquiring different game service providers to satisfy its customers.

3. Methodology of Service Implementation

To implement the cross platform gaming as a service, the methodology used is similar to the software development life cycle. It is used as a basis for the service implementation. The following are considered in the service implementation methodology:

i. Requirement Phase
ii. Analysis Phase
iii. Deployment Phase

i. Requirement Phase

As Cloud providers are business organizations, the focus on the requirements are business oriented aimed at ensuring that the Cross platform online gaming experience could be a
source of income to these providers. Since the targeted users (people within the age range of 12 – 50) cover a large population of users or subscribers which is good for business, the provider will require very large data centers and servers which will service the teeming population of users and a fast downlink connection to customers.

Every user requires a computer system which has all the functionality required. Functionalities such as fast processing power, good response time, high bandwidth (for Uplink) etc are needed when dealing with the cloud services. Some Games require more processing power, as such part of the requirement is that the system should be at its best in portraying the client-server functionalities.

ii. Analysis Phase

Here, an in-depth look into the network architecture was done where focus was on distributed systems accessed over the Internet which are normally organized as client-server systems. In a client-server system, the user interacts with the cloud using the client software (e.g. a web browser or phone-based application (apps) or other console software). This interacts with another program (the cross platform software) running on various servers. The servers provide various platform games which the user enjoys access to irrespective of the platform used.

Owing to the fact that processing graphic software (game) is tedious to a processor, a three tier architecture system is employed to reduce the workload to be carried out by various servers since it will have to process request from many different users. By moving part of the processing to user, service provider will only have to manage the data and load balancing whenever their servers are being utilised. This will also improve network performance as well as reduce scaling up and down of services from time to time.

![Client-Server Connections](image)

**Fig 2: Client-Server Connections**

Above is a highlight of the architecture used in the client process.
Client Process
The pattern used in the architecture of the client side application layer is that of a model view-
controller (MVC). The model-view-controller pattern is well known in applications that rely
on graphical user interfaces (GUI), user input, and logic. However, this architecture does not
follow the traditional MVC model. Here, the link between the traditional model and view are
eliminated to provide a persistent interface and increased speed of the system. By persistent
interface we mean that if a player leaves a game at some point in the play, the interface state
is preserved so that re-entry into play takes the player back to where the game was ended. In
this architecture, all module communication is routed through the Controller (Mayor). All
information, messages, and data will travel through central logic, thereby creating a more
maintainable system. The details of this modified MVC model are discussed below.

Mayor: Mayor is primarily responsible for local decision-making; The Mayors are the
knowledge sources for the blackboard on the server. The Mayor is responsible for passing
messages and requesting information between all the client-side modules. It is also
responsible for keeping track of the client side timer. The Mayor has the following
components, which are the Library, AI, Graphics, and Command.

Library Component: The Library component contains the Library, which holds the states of
all the objects in the client’s worldview. It is analogous to the World module on the server
side. The Library module should only have to be loaded once, at user initialization, and the
Mayor updates the module as many times as necessary when the client receives messages
from the server to keep the client’s worldview up-to-date.

AI Component: The Artificial Intelligence (AI) component contains the AI module that acts
as a player control for the pseudo client process on the server. The AI module receives data
from the Mayor as would the View, and responds to that data, based on some logic, the same
way the View would respond as if the user had sent the module the input.

Graphics component: The Graphics component contains the View, Graphics and Input
modules. The view is called by the Mayor to start setting up graphics calls for the Graphics
module. The Graphics module renders the model to the screen. The module’s purpose is to
contain all platform or Application Program Interface (API) dependent graphics calls for
abstraction reasons. The View also has the responsibility of translating the user input into
messages that the Mayor can understand. This module abstracts all graphics, and input data
from the Mayor. The Input module takes input from the user and passes it to the View, which
determines what the input means and relays the appropriate message to the Mayor. The
purpose of this module is to abstract the type of input from the user, be it keyboard, mouse,
joystick, etc.
**Command component:** The Command component contains the Command and Chat modules. The Command module abstracts certain user commands from the Mayor because they do not have any direct bearing on the game play. These commands could consist of calls to check the time (on both the client and server side), or ask how many hits the user has made, etc. The Chat module is responsible for handling specific types of commands. The module is to parse the chat command, form the message, and pass it to the Mayor, which will send it to the server for distribution.

**iii. Deployment Phase**

The purpose of the deployment phase is to ensure that the Service is properly deployed. Generally, implementing a Cloud service aimed at replacing an on-premise major business application may seem at times a simple straightforward implementation but it is fraught with pitfalls which may undermine the true value of the investment and in fact put enterprises in worst situations than before. Some of the things that were considered here are:

a) The Virtualization of the Game service providers (servers) have to be done during the deployment process, creating the first layer for interoperability between the servers and the different platforms many different users are working with.

b) Application virtualization describes software technologies that improve portability, manageability, and compatibility of applications by encapsulating them from the underlying operating system on which they are executed. After that, the gaming software which is uploaded into the servers provides users with multiple choices for their enjoyment.

The cross platform software exists and is made active whenever a user from one platform is to connect to another user on a different platform over the network. Session Initiation Protocol (SIP) enables flexible game interaction functionality. The Protocol can be used for handling online game sessions, presence and messaging in text or even data stream. Every player can be identified with a unique SIP address. The players can, for instance, invite other players who are not logged in the game to play with them.

It is pertinent to note here that issues like configuration, multitenant efficiency, scalability and more flexible billing system models are taken into consideration when implementing or porting the service. The player who connects to another platform is charged on a monthly service for cross platform software. Also, the game charges will be based on the method through which they subscribed. For example, if a subscription is made as a group of friends, charges are split to all friends involved in the activity.

**3. Service Limitations**

Elina, K. (2010) opined that issues in cross platform games include the graphics power (mostly based on GPU). The graphics power of mobile phones has increased a lot. The high end mobile phones already have accelerated graphics and are capable of showing equivalent
graphics as the Playstation2 gaming console but that is when they are dealing with standalone games. Cross platform online game software finds it difficult converting graphics based on one technology to another. Images become blurry when viewed on a platform which is not fully compatible with others. Also, when it comes to mobile phones, their biggest limitations include connectivity, small screen size and keypad, network latency, traffic pricing and platform fragmentation. However, due to large number of service subscribers (gaming), it was observed that “depending on the number of players and the system architecture, a massive multiplayer online role playing games might actually be run on multiple separate servers, each representing an independent world, where players from one server cannot interact with those from another; World of Warcraft is a prominent example, with each separate server housing several thousand players (Wikipedia, June 2010)

One can imagine how many players can be accommodated when there is interoperability between various platforms. The cloud provider will have to face the increasing demand for Ping Power and Pipe, which will cause more investment on the provider’s side because one has to scale up from time to time if the provider is to sustain the business or service.

4. Conclusion.
Cross-platform is an attribute conferred to computer software or computing methods and concepts that are implemented and inter-operate on multiple computer platforms. As technology migrates from the traditional on-premise model to the new cloud model, service offerings evolve almost daily. The new cloud model has made it possible to deliver such new capabilities to new markets via web browsers. With cross platform online gaming, the issues of purchasing the software for a console could be remedied since all what a targeted user has to do is to subscribe to the service. Once cross platform gaming takes off, it will give another big boost to casual gaming, unite the world in play, on all kinds of devices and all kinds of online and social game platforms. With cross platform gaming we are almost there.

5. Recommendations
Variation in computing platform limits the player’s interest in game playing, however, with cross platform online games deployed to the cloud the barrier is no more, one can play with multiple people using the available device and sharing the same game license. Thus cloud service providers should employ the offerings in cross platform online gaming service to fully entertain its teeming users despite their platform differences and which will boost income to these providers. On the issue of service limitations, with improvements in mobile and computer communication technology, the problem can be resolved however, where there is poor network access; the user stands to be disappointed.
References


