CAREER POINT UNIVERSITY HAMIRPUR

CPUH-Research Journal: **2015, *1(2)*,** 19-22
ISSN (Online): 2455-6076
http://www.cpuh.in/academics/academic_journals.php

Cloud Gaming: Architecture and Quality of Service

Anmol Gupta[1]* and Kamlesh Dutta[1]

[1]*Department of Computer Science & Engineering, NIT, Hamirpur (H.P.), INDIA*

* *Correspondance Email: bratanmol@gmail.com, kdnith@gmail.com*

ABSTRACT: Cloud Gaming can be defined as the offloading of game logic to the cloud server and rendering the game scenes as a video stream transfer to the client. The thin client sends the user commands to the remote server which executes the game logic and sends back the changed game scene to the client in the form of a video, which is decoded at the client side. The scheme helps people to play games on their computer system irrespective of the specification of their systems as the requirements for playing the game are fulfilled with the help of a good enough server residing on the cloud. Cloud gaming also helps to overcome the traditional gaming issues such as incompatibility and portability. Cloud gaming differs from online gaming in executing the main game logic, which is done at the server side in former and at the client side in latter. In this paper, the different aspects of the Cloud Gaming System (CGS) are discussed to illustrate the various advantages and power that this concept possesses. The cloud gaming concept, architecture of a cloud system, and Quality of Service (QOS) parameters are discussed in detail. A cloud gaming system should aim at providing the best response time and low bandwidth consumption. The future of cloud gaming seems to be bright as some cloud gaming systems like Gaikai and On Live are gaining popularity at a commendable pace.

Keywords: Cloud Gaming; cloud server and Architecture.

INTRODUCTION: Cloud Gaming is defined as the use of remote resources in order to play games on the local system. Generally, the UI input events are transmitted with the help of a *thin client*, and then the game logic executes on the remote server which in turn, is reflected back to the client in the form of a video stream which gets played with the help of *thin client.*

Computer games are graphics intensive and hence there is a demand for new hardware which can meet these high-end demands[1]. The limitations affect the individual game player who has to change his hardware anytime a new incompatible version of a game is released. Also, there is hardware/software incompatibility and the additional time required in setting up the games.

In cloud gaming, the user can access the hardware of the cloud systems thus eliminating the need of reconfiguring his/her own hardware, installing the game or worrying about software/hardware incompatibility. Another significant advantage is that the game can be played on any system such as PC, mobile device and even on TVs with Set-top boxes provided you have a good enough internet connection.

It is noteworthy that Cloud Computing is different from online gaming. “Traditional” online games also take advantage of remote resources on the cloud but the execution of game logic is done at client side only, while the servers are there for handling the consistency between the states of various games being played on it. While in cloud gaming architecture the execution of program logic is done at server side[2].

Why Cloud Gaming is a good idea? Consider the case of a First Person Shooter (FPS) game “Battlefield 3”. The minimum requirements for playing this game is – Quad Core Processor, 4 GB RAM, 20 GB storage space, and a graphics card with at least 1GB RAM (e.g., NVIDIA GEFORCE GTX 560 or ATI RADEON 6950), which alone costs more than $500. Needless to say that the recent mobile systems, tablets or even some of the PCs don't meet this requirement. Also mobile terminals don't have the same architecture of a PC, have power limitations, distinct operating system, and many more limitations. All these are overcome by using Cloud Gaming. Cloud Gaming also offers an added advantage of better Digital Rights Management (DRM) as the hardware being used is controlled by the owner of the cloud system rather than the user [3]. Cloud Gaming being such a computationally intensive process requires many technologies to work in unison so that the user can't fathom the remote execution of the main game logic. In this paper, a detailed analysis of the various technologies used in providing a pleasant cloud gaming experience to the end user, is presented.

The paper is divided into following sections. After the introduction in section I, section II provides the general architecture of a cloud gaming system. Section III comprises of the various Quality of Service (QOS) parameters used for measuring the End User Experience. Finally section IV concludes the paper with a look ahead at the future of cloud gaming.

General Architecture of a Cloud Gaming System (CGS): Generally there can be three different types of remote rendering Real time Systems. (i) 3D Graphics Streaming[4 & 7], (ii) video streaming[5 & 6], and (iii) video streaming with post-rendering operations[14].

In 3D Graphics system the cloud server sends the graphics related commands to the client which then interprets it and renders the scene accordingly while in Video Streaming the server is responsible for rendering the 3D commands, converting them to 2D and then sending the video stream to the client. The third system comes in between the first two where the heavy work of rendering the 3D graphics is done on server while some low processor intensive work is completed on the client side via Thin Client.

There have been many thin client architectures proposed for distributed gaming systems [4-7]. They can be divided into two categories – instruction based systems and image based systems. The main difference between the two is that, in instruction based systems, only the instructions for generating the graphics corresponding to a control event are sent over the network while in image-based systems all the computationally intensive rendering of the game scene is done on the server side and is sent over the network in the form of a video stream. All the CGSs use image based thin client architectures only because they don't require client to have the computing resources which is the whole selling point of Cloud Gaming.

Here we discuss the general image based architecture of a CGS.

Thin Client: It consists of a User Interaction module and a video decoder. The User Interaction module is responsible for capturing all the control movements performed by the end user with the help of mouse, keyboard or any other input device.

The video decoder plays the video being streamed by the server in response to the moves made by the player.

Cloud Gaming Platform: It consists of four modules namely Thin Client Interaction, Game Logic, GPU Renderer and Video Encoder. Thin Client Interaction module handles all the client commands and it converts the messages sent over the network into appropriate game actions. These are interpreted by the game logic module in game world and then the GPU will actually render the scene which then gets compressed by the encoder and then sent by the Video Streaming service to the thin client which then decodes the stream and play the video frames to the client [3].

There are several design alternatives which can be considered to provide different implementations of a CGS.

1) The way the existing game software is modified and run on the server
2) The way the game screen is encoded (on the server) and decoded (on the client)
3) The way the encoded game screen is streamed to the client.
4) The way short-term network instability is handled to maintain the game's responsiveness and graphic quality.

Various QOS Parameters: The effectiveness of a CGS can be measured from many point of views, from the service provider's view, the allocation of resource is important while from the end user's perspective the metrics which affect gaming experience are important. We can also quantify the QOS from the time scale, it can be larger time-scale across multiple gaming sessions and smaller across individual game session. Here we will focus only upon small time-scale systems because most CGSs uses only single Virtual Machine without any offloading to serve each client.

The main metrics that are focused upon:

1. *Traffic Characteristics:* The amount of bandwidth utilized in a single game session. It also includes payload size and packet rate (both uplink and downlink).
2. *Latency:* It's probably the most important criterion for measuring the performance of CGSs. It can be defined as the response time of a system and it comprises all the individual latencies incurred by different components.
3. *Graphics Quality:* The quality of the images/videos streamed over the network is very important for the Quality of Experience (QOE) for the user. Measurements also include the changes in quality over varying network conditions. Streaming quality is generally measured with the metric Frame rate (FPS). Graphic Quality can be measured by Peak Signal to Noise Ratio (PSNR) or Structural Similarity Index Method (SSIM) metric[9].

Latency quantifies the responsiveness of game playing and is a measure of Response Delay which is the amount of time between a user sends his/her command and the time the corresponding game frame is displayed to the user.

Response Delay (RD) comprises of four individual delays namely Network Delay (ND), Processing Delay (PD), Game Delay (GD) and Playout Delay (OD)[2].

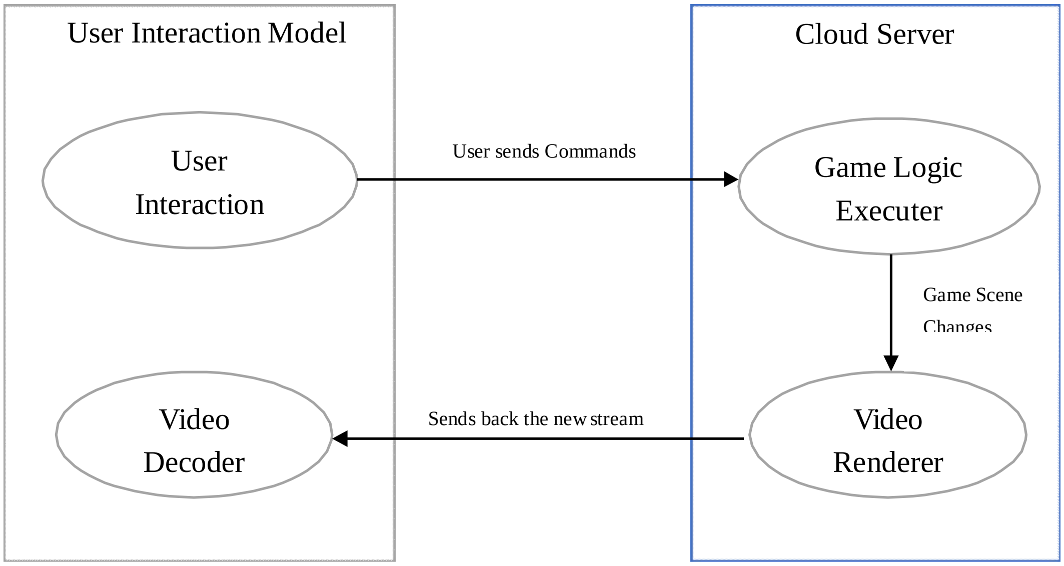

Figure 1: General Framework of a CGS

Network Delay (ND): The round trip time of the network, it actually measures the time taken by a client's command to reach the server and the time taken by the game screen to come to the client.

Processing Delay (PD): Time taken by the server to receive and process the user's command. It also includes time to encode and packetize the current frame for the client.

Game Delay (GD): Time taken by the game software to process the user's command and generate the corresponding game frame. This is usually considered to remain unchanged in the cloud environment and hence GD in standalone gaming equals GD in Cloud Gaming.

Playout Delay (OD): Time for client to receive, decode and play the current frame.

$$\mathbf{RD = ND + PD + GD + OD}$$

ND can be measured with the help of ICMP pings or Wireshark or any other such network tools while GD is game dependent. Measuring PD and OD is not straightforward because they occur internally at the server and the client respectively.

As mentioned that RD is an important factor in deciding the quality of the CGS, various studies have been performed on the delay tolerance for different kind of games. The results can be summarized as [10]:

Table 1: Delay Tolerance in Traditional Gaming.

Example Game Type	Perspective	Delay Thresh-
First Person Shooter	First Person	100 ms
Role Playing Game	Third-	500 ms
Real Time Strategy	Omnipres-	1000 ms

CONCLUSION: This paper compiles the various aspects of cloud computing and discusses some of it's aspects at length. The platform is gaining much popularity and is particularly powerful in bringing the graphics and processing heavy games to the mobile platforms, which in comparison to a general computer system, has less resources. Some of the shortcomings of cloud gaming are high network latency and the effectiveness of thin client. The future of cloud computing seems bright as network latencies are becoming shorter with the advent of technologies such as 3G/4G making their mark. Also, the techniques used in cloud gaming can serve for other industries too, such as video streaming and graphics rendering.

On Live and Gaikai are the two mainstream commercial platforms which are successful in the market while open source platforms are emerging slowly. One such open source platform is Gaming Anywhere.

As there are many design alternatives available, it is not yet decided which design system offers the best quality of service and an empirical study is needed to help the decision.

REFERENCES:

1. Claypool, M. 2009. Motion and scene complexity for streaming video games," *in Proceedings of the 4th International Conference on Foundations of Digital Games. ACM*. 34–41.

2. Chen, K. T. 2013. On the quality of service of cloud gaming systems. *IEEE transactions on multimedia*.

3. Shea, R. Cloud Gaming: Architecture and Performance.

4. Jurgelionis, A., Fechteler, P., Eisert, P., Bellotti, F., David, H., Laulajainen, J. P., Carmichael, R., Poulopoulos, V., Laikari, A., Per¨al¨a, P., Gloria, A. De and Bouras, C. 2009. *Platform for distributed 3D gaming," Int. J. Comput. Games Technol.*, 1:1-15.

5. Winter, D. De, Simoens, P., Deboosere, L., Turck, F. De, Moreau, J., Dhoedt, B. and Demeester, P. 2006. A hybrid thin-client protocol for multimedia streaming and interactive gaming applications," *in Proceedings of ACM* NOSSDAV 2006. ACM, 15:1–6.

6. Holthe, I., Mogstad, O. and Rønningen, L. A. 2009. Geelix livegames: Remote playing of video games," *in Proceedings of IEEE CCNC* 2009. IEEE Press. 758–759.

7. Eisert, P. and Fechteler, P. 2008. Low delay streaming of computer graphics," *in Proceedings of IEEE ICI*P 2008. 2704–2707.

8. Nave, I., David, H., Shani, A., Tzruya, Y., Laikari, A., Eisert, P. and Fechteler, P. 2008. Games@Large graphics streaming architecture," *in IEEE International Symposium on Consumer Electronics 2008*.

9. Wang, Z., Bovik, A., Sheikh, H. and Simoncelli, E. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* 13: 600–612.

10. Claypool, M. and Claypool, K. 2006. Latency and player actions in online games. *Communications of the ACM*, 49: 40– 45.

11. Shi, S., Hsu, C. Nahrstedt, K. and Campbell, R. 2011. Using graphics rendering contexts to enhance the real-time video coding for mobile cloud gaming. *In Proc. of ACM Multimedia'11*. 103–112.

12. Giesen, F., Schnabel, R. and Klein, R. 2008. Augmented compression for server-side rendering. *In Proc. of International Fall Workshop on Vision, Modeling, and Visualization* (VMV'08).