Cloud Gaming: Architecture and Performance

[1]Prithwiraj Prakash, [2]Nakul S Kumar, [3]Dr. Neetha Thomas

[1]Student, [2]Student, [3]Assistant Professor
[1]Computer Science,
[1]Santhigiri College of Computer Science, Thodupuzha, India

Abstract: Recent developments in cloud computing have made the concept of "cloud gaming" a reality. In its most basic form, cloud gaming involves rendering an interactive game application remotely in the cloud and returning to the player via the Internet with scenes streamed as a video sequence. This is beneficial for less capable computing systems that would otherwise be unable to run high-quality games. Onlive and Gaikai, two industrial pioneers, have enjoyed market success and substantial user bases. In this article, we perform a methodical analysis of contemporary cloud gaming platforms and draw attention to the distinctiveness of their framework designs. We also measure their real world performance with different types of games, for both interaction latency and streaming quality, revealing critical challenges toward the widespread deployment of Cloud Gaming.

Index Terms: image processing, cloud gaming, interaction delay, image quality.

I. INTRODUCTION

Cloud computing has opened up numerous new possibilities for both new and existing applications thanks to the use of elastic resources and widely dispersed data centers. By utilizing cloud computing platforms, already-existing applications—from media streaming to file sharing and document synchronization—have seen significant improvements in their usability and system effectiveness. These technological developments are largely the result of exploiting the vast resources of the cloud through computational offloading and lowering user access latencies with carefully placed cloud data centers. The concept of cloud gaming has recently become a reality thanks to developments in cloud technology that now permit offloading of more complex tasks like high definition 3D rendering in addition to traditional computations.

In its most basic form, cloud gaming involves rendering an interactive game application remotely in the cloud and streaming the scenes as a video sequence back to the player over the Internet. The thin client, which is in charge of showing the video from the cloud rendering server as well as gathering the player's commands and sending the interactions back to the cloud, is how a cloud gaming user interacts with the application.. Onlive [1] , Gaikai [2] are two industrial pioneers of cloud gaming, both having seen great success with multimillion user bases. The recent 380 million dollar purchase of Gaikai by Sony [3], an industrial giant in digital entertainment and consumer electronics, shows that cloud gaming is beginning to move into the mainstream. From the perspective of industry, cloud gaming can bring immense benefits by expanding the user base to the vast number of less-powerful devices that support thin clients only, particularly smartphones and tablets.

As an example, the recommended system configuration for Need for Speed™ Heat, a highly popular racing game, is a quadcore CPU, 16 GB RAM, 50 GB storage space, and a graphics card with at least 4GB RAM (e.g., NVIDIA GEFORCE GTX 1060 or RADEON RX 480), which alone costs more than $500. The newest tablets (e.g., Apple’s iPad with Retina display and Google’s Nexus 10) cannot even meet the minimum system requirements that need a dual-core CPU over 3.40 GHz, 8 GB RAM, and a graphics card with 2 GB RAM, not to mention smartphones of which the hardware is limited by their smaller size and thermal control. Additionally, compared to PCs, mobile terminals have different hardware and software architecture, such as ARM rather than x86 for the CPU, lower memory frequency and bandwidth, power restrictions, and unique operating systems. As a result, the conventional console game model is impractical for such devices, which turn into targets for Gaikai and Onlive. Because the computational hardware is now entirely under the control of the cloud gaming provider, cloud gaming also lowers the cost of customer support while also providing better Digital Rights Management (DRM) due to the fact that the codes are not directly executed on a customer's local device.

Fig 1 : Cloud Gaming Overview

II. CLOUD GAMING : ISSUES AND CHALLENGE

Interaction Delay Tolerance

Different game types have different maximum tolerable delay [4] thresholds, according to studies on traditional gaming systems. Table I lists the longest delay that a typical player can stand before the quality of their experience (QoE) starts to suffer.

Delay Tolerance in Traditional Gaming [11]		
Example Game Type	***Perspective***	***Delay Threshold***
First Person Shooter (FPS)	First Person	100 ms
Role Playing Game (RPG)	Third Person	500 ms
Real Time Strategy (RTS)	Omnipresent	1000ms

First Person Shooter (FPS) games like Counter Strike, Valorant become noticeably less playable when actions are delayed by as little as 100 ms. This low tolerance for delays is a result of the action-heavy nature of these first-person games, which disadvantage players with longer delays [5]. An action-based First Person Shooter (FPS) game can be particularly sensitive to delays in the outcomes of decisive game-changing actions, such as who "pulled the trigger" first.

Role-playing games (RPGs) and many other massively multiplayer online games, like World of Warcraft, frequently have a higher delay tolerance of up to 500 ms. This is due to the fact that in these games, player commands like "use item," "cast spell," or "heal character" are typically carried out by the player's avatar. In this type of games, player may become upset if the interaction delay results in a bad result, such as when they heal before an enemy attack but still perish because their commands were not registered by the game in time. This is why the actions must still be registered promptly.

Simulation games like The Sims and real-time strategy (RTS) games like Star Craft. These types of games can tolerate delays up to 1000 milliseconds because the player frequently commands numerous entities and executes numerous individual commands, many of which take several seconds or even minutes to complete. A build unit action that takes more than a minute may have a delay of up to 1000 ms in a typical RTS game, but the player won't likely notice it.

Video Streaming and Encoding Now let’s examine video streaming and encoding needs in a cloud gaming system. Cloud gaming’s video streaming requirements are quite similar to another classical application, namely, live media streaming. Both live media streaming and cloud gaming require quick encoding/compression of incoming video and distribution to end users. The fact that we are only interested in a small subset of the most recent video frames and do not have access to future frames before they are created means that encoding must be performed with respect to a very small subset of frames in both cases.

Cloud gaming has essentially no client-side buffering capacity when compared to live media streaming. This is due to the fact that any commands given by players to the local thin client must first travel across the Internet to the cloud, where they must then pass through the game logic, be rendered by the processing unit, be compressed by the video encoder, and then be streamed back to the player. There does not appear to be much room for a buffer given that everything must be completed in less than 100–200 ms. Major Cloud gaming providers encode using H.264/MPEG-4 AVC encoder. For eg, two major cloud gaming platform are Gaikai and OnLive. Gaikai uses a software based approach for encoding where as Onlive is using specialized hardware to compress its cloud gaming video streams.

III. FRAMEWORK

The template is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin in this template measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

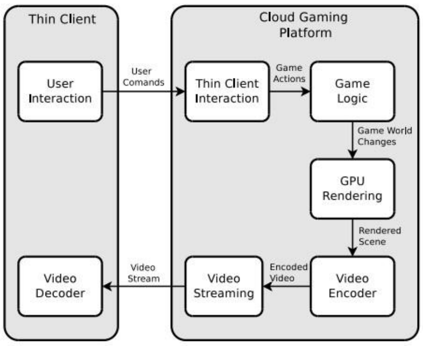


Fig 2 : Framework of Cloud Gaming Platform [11].

IV. REAL WORLD PERFORMANCE

Real world performance is evaluated the using Batman Arkham Asylum on Onlive and compare its performance to a copy of the game running locally. In our analysis, we look at two important metrics, namely, the interaction delay (response time) and image quality. Our hardware remains consistent for all experiments. We run Batman through an Onlive thin client as well as locally on our local test system. The test system contains an AMD 7750 dual core processor, 4 GB of ram, a 1-terabyte 7200 RPM hard drive, and an AMD Radeon 3850 GPU. The network access is provided through a wired connection to a residential cable modem with a maximum connection speed of 25 Mb/s for download and 3 Mb/s for upload.

Measuring Interaction Delay

Define A crucial metric to track is the reduction of interaction delay, which is a fundamental design challenge for cloud gaming developers. We employ the following method to precisely measure interaction delay for Onlive and our local game:

First, we install and configure our test system with a video card tuning software, MSI afterburner. It allows users to control many aspects of the system’s GPU, even the fan speed. It allows users to control many aspects of the system’s GPU, even the fan speed. Second, we configure our screen capture software to begin recording at 100 frames per second when we press the ―Z‖ key on the keyboard. The Z key also corresponds to the ―Zoom Vision‖ action in our test game. We start the game and use the zoom vision action. By looking at the resulting video file, we can determine the interaction delay from the first frame that our action becomes evident. Since we are recording at 100 frames per second, we have a 10 millisecond granularity in our measurements. To calculate the interaction delay in milliseconds, we take the frame number and multiply by 10ms. Since recording at 100 frames per second can be expensive in terms of CPU and hard disk overhead we apply two optimizations to minimize the influence that recording has on our games performance.

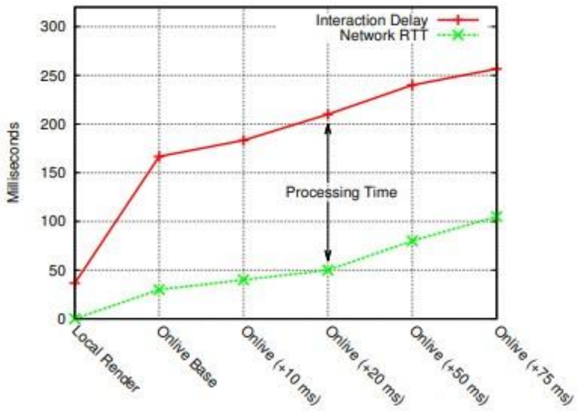


Fig 3 : Interaction Delay in Onlive [11].

Installing a Linux software router between our test system and Internet connection will cause network delays. We installed the Linux network emulator Netem on our router, which enables us to regulate network conditions like network delay. While the same game action is registered in our locally rendered copy with an average interaction delay of about 37 ms, it takes our Onlive baseline about four times as long—167 ms—to do so. The interaction delay increases as expected when we simulate longer network latencies. In a surprising number of our tests, the Onlive system maintains an interaction delay under 200 ms. This shows that Onlive could offer tolerable interaction delays for many game genres. But when the network latency is greater than 50 milliseconds, the interaction delays the interaction delays may begin to hinder the users’ experience.

In Fig., the processing time is defined as the interaction delay brought on by the game logic, GPU rendering, video encoding, etc.; in other words, it is the interaction delay components that are not brought on by network latency. For instance, the processing time for the game's locally rendered version, which has no network latency, is just 37 ms. On the other hand, our Onlive-base case experiences a communication delay of about 30 ms as a result of network latency, resulting in a processing time of about 137 ms.

PROCESSING TIME AND CLOUD OVERHEAD		
Measurement Processing Time (ms) Cloud Overhead (ms)	***Measurement Processing Time (ms) Cloud Overhead (ms)***	***Measurement Processing Time (ms) Cloud Overhead (ms)***
Local Render Onlive base Onlive (+10 ms) Onlive (+20 ms)	36.7 136.7 143.3 160.0	NA 100.0 100.7 123.3
PROCESSING TIME AND CLOUD OVERHEAD		
Measurement Processing Time (ms) Cloud Overhead (ms)	***Measurement Processing Time (ms) Cloud Overhead (ms)***	***Measurement Processing Time (ms) Cloud Overhead (ms)***
Onlive (+50 ms) Onlive (+75ms)	160.0 151.7	123.3 115.0

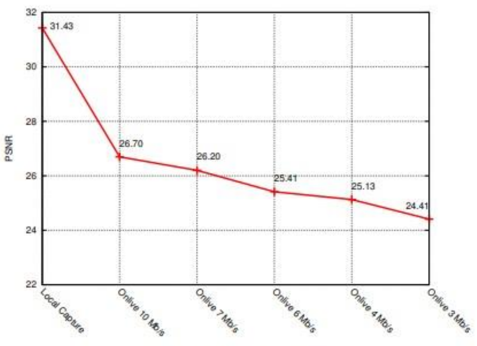
The interaction processing and cloud overhead that we measured in our experiments are shown in above table. As can be seen, the cloud processing increases the Onlive system's interaction delay by about 100–120 ms. According to this finding, the processing overhead in the cloud alone is over 100 ms, indicating that any attempt to meet the ideal interaction delay threshold will call for more effective designs for video encoders and streaming software.

Measuring Image Quality

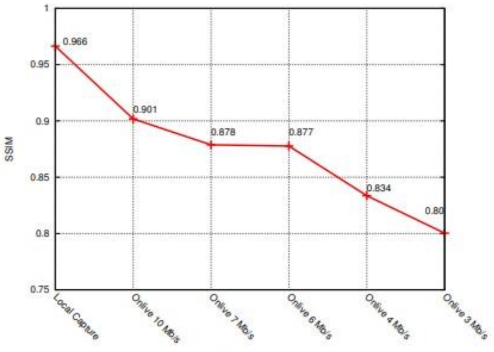
Image quality is just as important to a cloud game player as low interaction latency.. Onlive employs a hardware H.264 encoder with a real-time encoding profile, indicating that some degree of image quality loss will result from compression. There are a number of technical difficulties in developing a methodology to impartially assess the image quality of a commercial cloud gaming system like Onlive.

First, we need to be able to record a deterministic sequence of frames from Onlive and compare it to our local platform in order to get an accurate sample for the video quality analysis. Although the stream is known to be encoded by H.264, it is difficult to directly capture and analyse the stream packets because it appears that Onlive is using a proprietary RTP variant (RTP). Onlive's rendering preferences are not accessible to the general public either. For instance, it is still unknown if Onlive has turned on antialiasing or what the draw distance is for any particular game. In order to evaluate the Onlive image quality, we came up with the following methodology.

Again selected our test game Batman Arkham Asylum, and we use the same test platform OnLive. To mitigate the effect that different rendering settings have on the image quality, we choose the pre-rendered intro movie of the game to record. To improve the accuracy of our analysis, we unpack the intro video's master file from the game files of our local copy of Batman Arkham Asylum. The extracted movie file has a resolution of 1280 x 720 pixels (720p), which perfectly matches the video streamed by Onlive. And also configured our local copy of Batman to run at a resolution of 1280 x 720 pixels. Configured display driver to force a frame rate of 30 FPS to match the rate of target video. Next, configure MSI afterburner to record the video uncompressed with a resolution of 1280 x 720 pixels at 30 FPS. The lack of video compression is very important as we do not want to taint the samples by applying lossy compression. We then capture the intro sequence of our locally running game and Onlive running with different bandwidth limits.



(a) PSNR experiments



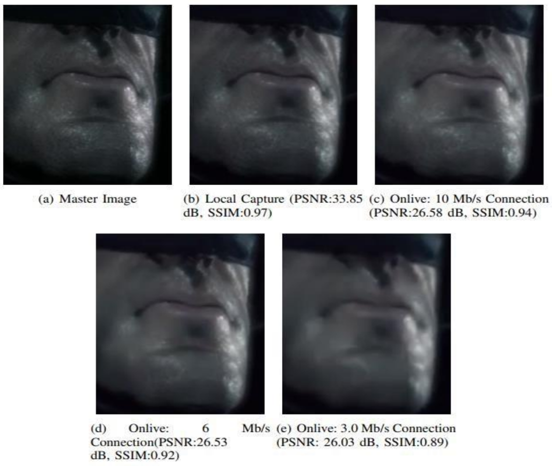
(b) SSIM experiments

To control the bandwidth, again use our Linux software router and perform traffic shaping to hit our targets. Test Onlive running from its optimal bandwidth setting of 10 Mb/s gradually down to 3.0 Mb/s. It covers a broad spectrum of bandwidths commonly available to residential Internet subscribers. Before each run, ensure our bandwidth settings are correct by a probing test.

After capturing all the required video sequences, we select the same 40 second (1200 frame) section from each video on which to perform an image quality analysis. Analyze the video using two classical metrics, namely Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Method (SSIM). The results for PSNR are given in Figure 4(a) and SSIM are given in Figure 4(b), respectively.

The PSNR method measures how much error (noise) was added to the reconstructed video during compression.. The SSIM method calculates the structural similarity between the two video frames. As can be seen, our local capture scored a high PSNR and SSIM; however it is not perfect, indicating some difference in the recorded video and the master file. Much of this difference is likely due to slightly different brightness and colour settings used by the internal video player in the Batman game engine. When the local capture is compared to Onlive running at any connection rate, we can see a large drop in terms of both PSNR and SSIM.

Since PSNR and SSIM are not on a linear scale, the drops actually indicate a considerable degradation in image quality


(a) Master Image

(b) Local Capture (PSNR:33.85 dB, SSIM:0.97)

(c) Onlive: 10 Mb/s Connection (PSNR:26.58 dB, SSIM:0.94)

(d) Onlive: 6 Mb/s Connection(PSNR:26.53 dB, SSIM:0.92)

(e) Onlive: 3.0 Mb/s Connection (PSNR: 26.03 dB, SSIM:0.89)

Fig 5: Image Quality Comparison

Generally a PSNR of 30 dB and above is considered good quality, however 25 and above is considered acceptable for mobile video streaming. Not surprisingly, as we drop our test systems connection bandwidth the image quality begins to suffer considerable degradation as well. With the exception of the 3.0 Mb/s test, all samples stay above a PSNR of 25 dB; so although there is room for improvement, the image quality is still acceptable. Fig 5 illustrates the effect of Onlive's compression taken from a single frame of the opening sequence. As can be seen the effect of compression is quite noticeable especially as the amount of available bandwidth decreases.

V. CONCLUSION

The framework design of modern cloud gaming platforms has been thoroughly examined in this article. One of the most representative and profitable cloud gaming platforms to date, Onlive, has also had its performance evaluated. The results, in particular on interaction latency and streaming quality under various game, computer, and network configurations, have shown both the potentials of cloud gaming and the significant obstacles to its widespread adoption. We would like to look more closely at how other network issues like packet loss and jitter affect the end user's cloud gaming experience in a subsequent study. An exciting new technology that is quickly developing is cloud gaming. One that is frequently mentioned is bringing cutting-edge 3D content to comparably less powerful devices like smartphones and tablets. The fact that Gaikai and Onlive are both actively developing Android apps for these mobile platforms makes this observation even more pertinent. However, recent extensive research shows that cellular network connections with network latencies greater than 200 ms are common [7], which alone may already cause the interaction delay to become too high for many games.

The switching to LTE may help solve the issue, which is expected to be seamless integration between cellular data connection and the slower WiFi connection. Other possible developments include distributed game execution across various specialized virtual machines or intelligent thin clients that can handle some of the game logic and rendering locally to hide some of the problems related to interaction delay [8]. Games tailored for cloud platforms will probably need to be developed in order to achieve this.

In addition to software and service providers, hardware manufacturers have also expressed a strong interest in cloud gaming. Some have started developing specialized hardware solutions to address the key problems with cloud gaming. The GeForce grid graphical processor, developed by NVIDIA specifically for cloud gaming systems, was just unveiled [9]. It functions as an all-in-one graphic processor and encoding solution. According to the published specification, each of these processors is capable of rendering and encoding four games at once. According to internal tests conducted by NVIDIA, the latency introduced by current cloud gaming systems can be significantly reduced [10]. It is widely believed that this type of specialized hardware will usher in a new era of cloud gaming.

REFERENCES

1. Onlive. http://www.onlive.com/.
2. Gaikai. http://www.gaikai.com/.
3. Engadget. Sony buys Gaikai cloud gaming service for 380 million. http://www.engadget.com/2012/07/02/sony-buys-gaikai/.
4. M. Claypool and K. Claypool. Latency and player actions in online games. Communications of the ACM, 49(11):40–45, 2006.
5. M. Claypool and K. Claypool. Latency can kill: precision and deadline in online games. In Proceedings of the first annual ACM SIGMM conference on Multimedia systems, pages 215–222. ACM, 2010.
6. M. Jarschel, D. Schlosser, S. Scheuring, and T. Hoßfeld. An evaluation of qoe in cloud gaming based on subjective tests. In Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2011 Fifth International Conference on, pages 330–335. IEEE, 2011.
7. J. Sommers and P. Barford. Cell vs. wifi: on the performance of metro area mobile connections. In Proceedings of the 2012 ACM conference on Internet measurement conference, pages 301–314. ACM, 2012.
8. Z. Zhao, K. Hwang, and J. Villeta. Game cloud design with virtualized cpu/gpu servers and initial performance results. In Proceedings of the 3rd workshop on Scientific Cloud Computing Date, pages 23–30. ACM, 2012 [9] Geforce grid. http://www.nvidia.ca/object/grid-processors-cloudgames.html.
9. James Wang. Nvidia geforce grida glimpse at the future of gaming. http://www.geforce.com/whats-new/articles/geforce-grid.
10. https://www.sfu.ca/~rws1/papers/Cloud-Gaming-Architecture-and-Performance.pdf