Enactment of Profitable Streaming of Video Games from Cloud with Run-Down Latency via Cloud Gaming system

Rudresh. M. S[#1], Dr. Ravikumar. G. K [*2], Yogendra Kumar Verma[#3]

[#]P.G.Scholar, Department of Computer Science, BGSIT
Visveswaraya Technological University, Karnataka, India
[1]rudresh999@gmail.com
[3]yogendragd@gmail.com

[*2]Professor, Department of Computer Science
B.G.S Institute of Technology, Karnataka, India
[2]gkravikumar@yahoo.com

***Abstract*—As a new exemplar, cloud gaming permit users to play high-end video games instantly without downloading or mount the inventive game software. In this phenomenon, we first conduct a series of well-designed active and passive measurements on a large-scale cloud gaming stage and identify that there exists significant multiplicity in the queuing delay and response delay among cloud users. We emphasize that the latency problem mostly results from user-specified request routing and inflexible server anticipation. To report latency problem of the cloud gaming platform, we further insinuate an online control algorithm called *iCloudAccess* to impersonate intelligent request send off and server anticipation. Our foremost objective is to rein back the provisioning outlay of cloud gaming service providers while still guarantee the user QoE requirements. We articulate the problem as a constrained stochastic optimization problem and apply the optimization theory to derive the online control algorithm with incontestable upper-bounds. We also bearing extensive trace-driven simulations to appraise the effectiveness of our algorithm and our ravages show that our proposed algorithm achieves substantial gain over other substitute approaches.**

Keywords*— *Cloud Gaming, Video Streaming, Queuing Delay, iCloudAccess, Server Anticipation

I. INTRODUCTION

Since the first video game was inaugurated on the market around 45 years ago, we have witnessed a progression of significant revolutions in the video game engineering. In recent years, the advent of cloud gaming provides a promising approach to make gaming (outstandingly high-end 3D video games) more affordable and reachable to game players. The basic plan of cloud gaming is to depict video games in the cloud and stream encoded game scenes to team via the broadband networks. Users can act together with the game application by delivering the control signals (e.g., key strokes, mouse clicks) to the cloud server. Users are calmed from downloading or installing the inventive game software. With such cloud-assisted gaming genre, users can with no trouble play high-end 3D video games on any device, such as PC, Set-Top Box (STB), iPad, smart phone, whenever and wherever conceivable. The probable of cloud gaming has already tempted a great amount of attention from many industrial practitioners, endeavors and researchers. It is expected that the size of global video game market revenue will grow to $78 billion in 2017, among which cloud gaming market is expected to expand the most [1]. However, it is very challenging to shape a cloud gaming platform that can endow users with high Quality-of- Experience (QoE). Surviving cloud gaming systems generally depend on a set of geographically distributed data centers to serve users in dissimilar regions. A game user request will be intent to a data center according to certain strategies (e.g., proximity). Due to the focused hardware requirements (e.g., GPUs), normally a specific cloud server, which can be either one a physical machine or a virtual machine, will be payable to a player exclusively upon receiving the request.

The cloud server is responsible for game rendering and streaming encoded game sights to the client. When the cloud platform cannot provision enough servers to meet user demand timely, user requests have to be queued for a period. As online game players are pretty impatient [2], if queueing delay is too long, it will result in the loss of user accesses. Especially, the increase of response delay is intolerable for real-time video games (e.g., First-Person Shooter (FPS) games). To better understand the problems and challenges therein, as the first step, we need to take a close look at the real cloud gaming systems. Starting from the first-hand observations, we can identify the underlying causes and address the problems in a right way for cloud gaming systems. In this paper, we focus on understanding and mitigating the latency problem of cloud gaming services from the perspective of cloud gaming service providers (CGSPs).

Especially, we are the first to study the queuing phenomena in a real-world cloud gaming system. To improve the QoE of game players in different regions, we further

propose an online control algorithm called iCloudAccess, which reduces queueing delay and response delay by smart request dispatching and server provisioning among data centers. Our proposed iCloudAccess can minimize the time-average server provisioning cost so as to be cost effective for CGSPs. In summary, our main contributions in this paper can be listed as follows:

We conducted an in-depth measurement study of CloudUnion, which is the first cloud gaming system in China and also proprietary. With both passive and active measurements, we are able to unveil the architecture and internal mechanisms of CloudUnion. We developed a customized crawler to query the status of user requests and obtained the queueing information of each data center. We observed that players have to wait in the queue for a rather long period due to improper request routing to data centers in hot regions. We also measured the response delay when accessing different data centers at different time points. We performed extensive trace-driven simulations to verify the effectiveness of our proposed iCloudAccess algorithm in the practical settings. Our simulation results show that, compared with other alternatives, iCloudAccess can save more than 30% of provisioning cost and reduce the queuing delay and response delay significantly.

The remainder of this paper is organized as follows: Section II introduces the concept and technologies of cloud gaming services. Section III presents our latency measurement results of CloudUnion and points out the underlying causes. In Section IV, we propose an online control algorithm called iCloudAccess to reduce user latency by smart request dispatching and server provisioning. In Section V, we evaluate the effectiveness of our proposed solution by simulation. Finally, Section VI concludes the paper and discusses the future work.

II. CLOUD GAMING: CONCEPT AND TECHNOLOGIES

Cloud gaming is also called "Gaming on Demand (GoD)" or "Gaming as a Service (GaaS)". Essentially, cloud gaming has much in common with a video-on-demand service, but more interactive. The player's computer receives streaming video (and audio) and sends the keyboard, mouse, and controller input actions to the cloud gaming server over the broadband networks. Fig. 1 briefly illustrates the concept of cloud gaming.

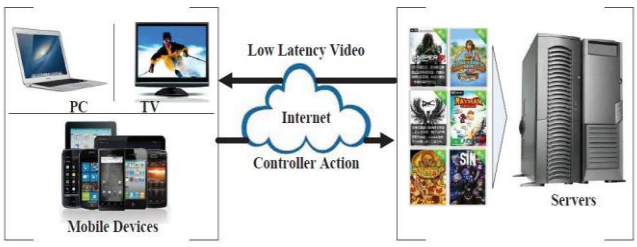

Fig. 1. The concept of cloud gaming

Unlike traditional PC games, cloud gaming offers many novel features: *Firstly*, with cloud gaming, players are relieved from expensive hardware investment and constant upgrades. A thin client (e.g., set-top box, laptop, mobile device) with a broadband Internet connection is enough to play any video games; *Secondly*, cloud gaming allows games to be platform independent and players don't need to worry about the compatibility issues when playing games. *Thirdly*, cloud gaming allows users to start playing games instantly, without the need to download and install the game images; General-purpose thin clients, such as VNC (Virtual Network Computing) [4], cannot satisfy the stringent requirements of cloud gaming on response time and frame rate [5].

Existing cloud gaming systems employ highly-optimized H.264/AVC codecs (e.g., x264 [6]) to perform real-time video encoding/ decoding on captured game frames. To reduce the processing latency, multi-threading is widely used to better leverage the computation power of multi-core CPUs and GPUs. The network communications between the client and the gaming server can be based on different real-time communication protocols, such as RTP [7], RTSP [8].

III. LATENCY MEASUREMENT OF A LARGE-SCALE CLOUD GAMING SYSTEM

To better understand the latency components of cloud gaming services, we first conduct extensive measurements on the cloud gaming platform of CloudUnion [11], which was the first to launch cloud gaming services in China and its subscribers have exceeded 300,000 as of July 2012.

A. *Cloud Union's Platform*

CloudUnion now offers more than 200 games via its cloud platform. Users can start playing games by either downloading the CloudUnion's client or directly using the web browser (e.g., IE, Firefox and Chrome). CloudUnion's platform supports game streaming in a variety of resolutions, ranging from 320x240 to 1024x768. The minimum bandwidth requirement for the client is 2 Mbps, but 6+Mbps bandwidth is recommended for high-quality game streaming. The measurement of CloudUnion is challenging because the CloudUnion's protocol is proprietary.

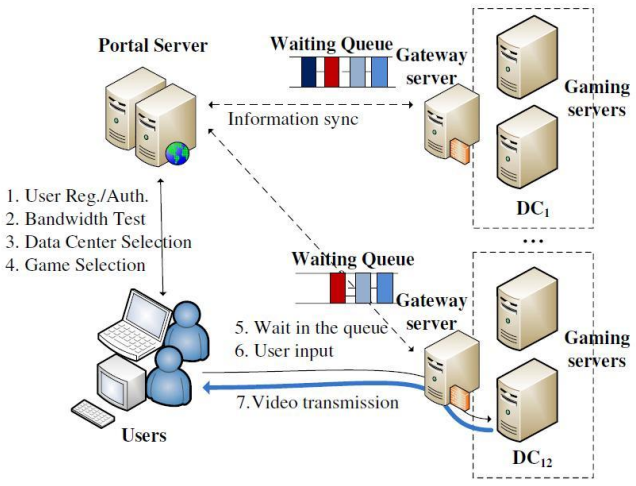

Fig. 2. The architecture of CloudUnion's cloud gaming platform

In order to understand the underlying protocol, we had to collect a large amount of Wireshark traces from multiple

gaming sessions and analyze the communications between the client and servers in the cloud platform. From the protocol analysis, we find that the CloudUnion's infrastructure can be illustrated in Fig. 2. To improve the Quality-of-Experience (QoE) of users in different regions, CloudUnion deploys its data centers in twelve geographically distributed locations.

A portal server is responsible for user registration, authentication and bandwidth test. After a user logs into the system, it should first manually choose a data center from a list and then select a preferred game. A user normally chooses a data center in a nearby region. The request will be routed to a gateway server of the selected data center. Upon receiving a user request, the cloud gaming platform will launch a dedicated server 1 to run the game specified in the request and stream the gaming video to the user client. When the capacity of a data center cannot meet the demand, user requests routed to that data center will be held in a waiting queue. In the whole gaming session, there are two major latency components for cloud game players, Queueing Delay, and Response Delay for measuring the delay variance in the service provided by the cloud game service.

B. Measurement of Queuing Delay

By performing multiple logins from the same location to a data center at different time slots of a day, we can calculate the median and variance of the queuing delay.

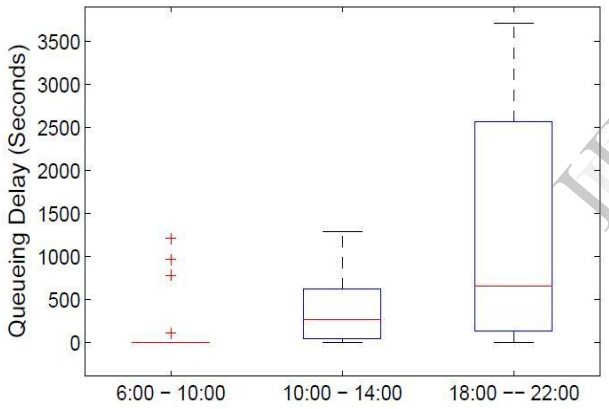

Fig. 4. Queueing delay experienced by a user at different time slots of a day

Fig. 4 shows the distribution of queuing delay at different time slots of a day. It is observed that there is almost no queuing delay for most requests during the morning time (6:00-10:00), while queuing delay becomes very serious in the night time (18:00-22:00), with over 50% requests being queued for more than 500 seconds.

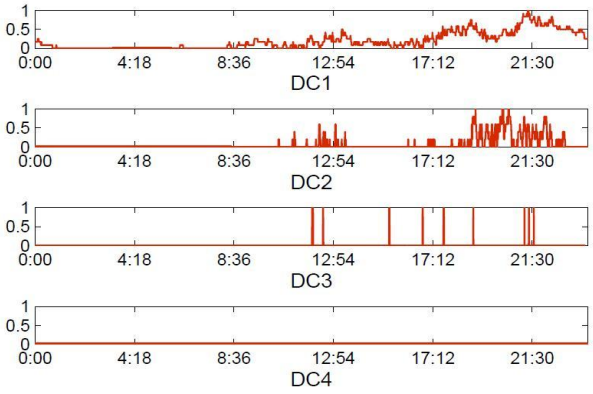

Fig. 5. The frequency of queueing phenomena in different data centers

Our measurement results point out that queuing delay exhibits both temporal and spatial diversity. Such diversity in queuing delay is largely caused by the sub-optimality of user-specified data center selection and the inefficiency of cloud resource provisioning. Actually, such kind of queuing problem widely exists in any online service system when the pace of resource provisioning cannot keep up with the increase of request arrivals.

Even with cloud computing, it still takes time (e.g., ranging from a few seconds to tens of minutes) to allocate and provision a new VM instance (see [13], [14], [15]), depending on the VM size. The queueing problem can be alleviated by faster VM provisioning, but cannot disappear due to the unpredictability of user demand. Therefore, we believe that cloud gaming systems outside of China (e.g., OnLive and GaiKai) will also have the same queueing problem.

C. Measurement of Response Delay

Queueing delay determines how long a user should wait before running a game, while response delay determines how interactive a cloud game is during a game session. The direct measurement of response delay is very difficult due to the proprietary nature of the CloudUnion system. Instead, we adopted a method similar to [12] to measure the response delay.

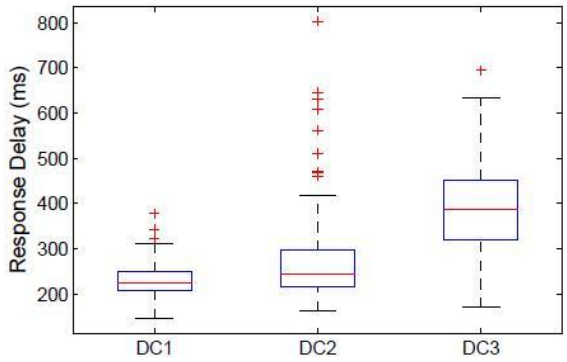

Fig. 6. The response delay when selecting different data centers

The basic idea is to calculate the time difference between the time a hot key is pressed and the time the updated screen is shown at the user side.

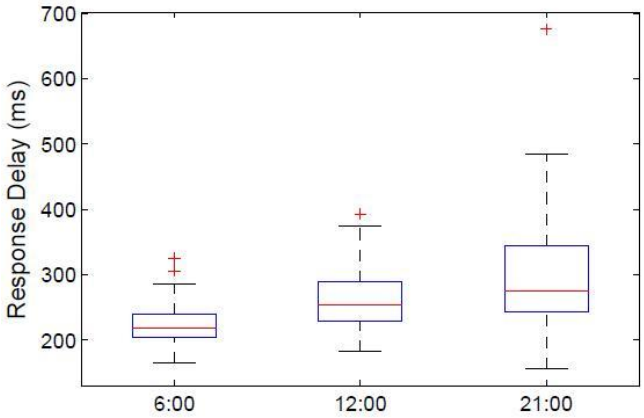

Fig. 7. The temporal diversity of response delay over a day

The pressing of a hot key can be simulated by utilizing the hooking mechanism in Windows, and the screen update can be detected by examining the color changes of a specific set of pixels. From the same location (i.e., Guangzhou), we initiated multiple game sessions simultaneously and measured the corresponding response delay to different data centers. Fig. 6 shows the difference of response delay when selecting different data centers. The response delay exhibits significant spatial diversity, with *DC*1 being the lowest and *DC*3 is the highest. For the same data center, we also initiated a series of game sessions at different time slots of a day and measured the dynamics of response delay. Fig. 7 clearly points out response delay also exhibits temporal diversity.

IV. ICLOUDACCESS: REDUCING LATENCY VIA ONLINE REQUEST DISPATCHING AND SERVER PROVISIONING

In this section, we design an online control algorithm called iCloudAccess to speed up the accesses of cloud game players. ICloudAccess provides a cost-effective approach to stream video games with low latency by smart request dispatching among data centers and dynamic cloud resource provisioning.

A. System Architecture

Fig. 8 describes the role of iCloudAccess in the cloud gaming platform. ICloudAccess contains two major components:

a) Request Dispatching Unit (RDU): This is responsible for dispatching requests intelligently.

b) Server Provisioning Unit (SPU): This is responsible for adjusting the number of game servers provisioned at each data center according to user demand.

The operations of RDU and SPU are performed at different time scales. For every incoming user request, RDU needs to dispatch the request timely and the dispatching operation needs to be completed within a few seconds;

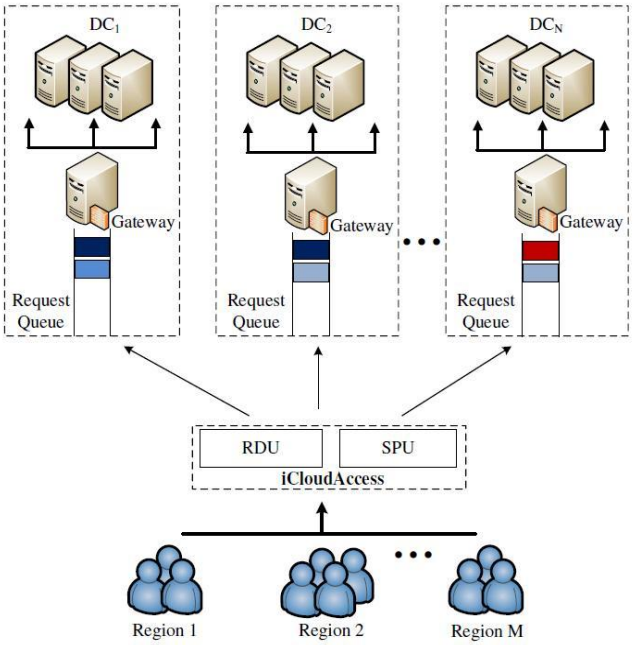

Fig. 8. System Architecture

However, the provisioning of cloud servers cannot be conducted in a real-time manner. Normally SPU adjusts the provisioning of cloud servers for each data center periodically, in the order of hours. By intelligently dispatching a request to a data center with a short waiting queue, queueing delay of a user request can be significantly reduced.

B. Online Control Algorithm for RDU and SPU

The optimization problem can be then solved by finding a strategy to minimize the "*relaxed*" upper bound Θ2. It can be easily verified that Θ2 is a convex function of $\vec{\boldsymbol{\lambda}}(t)$, $\vec{\boldsymbol{n}}(t)$. Thus, we can solve the minimization of Θ2 efficiently by exploiting standard convex optimization tools (e.g., cvx) and obtain the online decisions of RDU and SPU (i.e., $\vec{\boldsymbol{\lambda}}(t)$, $\vec{\boldsymbol{n}}(t)$). At the end of each time slot, all queues are updated accordingly.

The details of our online algorithm for requesting dispatching and server provisioning are given in Algorithm 1. The algorithm generates the decision $\vec{\boldsymbol{\lambda}}(t)$ on request dispatching every time slot, and the decision $\vec{\mathbf{n}}(t)$ on server provisioning every 'm' time slots.

Algorithm 1 *iCloudAccess*: Online control algorithm for RDU and SPU.

Input:
The values of *N,M, m, μ, ϵ, V* ;
Prices of on-demand cloud servers;
Number of incoming user requests $\lambda_{ij}(t)$;
Network delay between regions and data centers, $d_{ij}(t)$;
Output:
RDU and SPU decision $\vec{\boldsymbol{\lambda}}(t)$, $\vec{\boldsymbol{n}}(t)$.
1: Initialization step: Let $t = 0$, and set $Q_j(0) = 0$, $H_j(0) = 0$, for $j = 1, 2... N$.
2: **while** the cloud gaming service is running **do**

3: **if** (t mod m) == 0 **then**
4: Monitor the queue backlog $\boldsymbol{Q}(t)$, $\boldsymbol{H}(t)$ and the realtime information of $c_j(t)$ for each data center j.
5: Determine the SPU decision $\vec{\boldsymbol{n}}(t)$ by solving $\min_{\{\boldsymbol{n}(t)\}} \Theta 2$;
6: **end if**
7: Update information of network delay between a region i and a data center j, and the amount of user requests from a region i (i.e., $\lambda_i(t)$) for $i = 1, 2, ...,M, j = 1, 2, ...,N$.
8: Determine the RDU decision $\vec{\boldsymbol{\lambda}}(t)$ by solving $\min_{\{\lambda \vec{ij}(t)\}} \Theta 2$;
9: Update $\boldsymbol{Q}$ and $\boldsymbol{H}$ according to Eqn. (9) and Eqn. (11), respectively.
10: **end while**

In our algorithm, the time complexity lies in solving the optimization problem in each round. Efficient linear programming tools can be applied to resolve the time-slotted linear programming problem.

V. Cloud Gaming Experimental Evaluation

In this section, we develop a discrete-event simulator and conduct a set of experiments to evaluate the effectiveness of our proposed online algorithm.

A. Experimental Settings

In our simulation, we consider the following three typical request dispatching strategies:

a) *Proximity-aware Request Dispatching (PRD)*, in which requests are always dispatched to a data center with the lowest network delay (for example, a data center in a nearby region). It can maximize the reduction of the response delay for a user.
b) *Load-aware Request Dispatching (LRD)*, in which requests are always routed to a data center with the lowest workload level.
c) *Hybrid-weight Request Dispatching (HRD)*, in which we consider a weighted-sum of the normalized proximity factor and the normalized load factor as the metric.

Two server provisioning strategies are used for comparison in our experiments:
a). Stable Server Provisioning (SSP), in which a fixed number of cloud servers are provisioned at each data center. The server provisioning doesn't change over time. It is a simple yet widely adopted server provisioning strategy for CGSPs. In our experiment, we assume that the time average number of user requests for each data center is known beforehand, and the number of provisioned cloud servers equals to the time-average number of user requests
b). Queueing-aware Server Provisioning (QSP), in which the CGSP makes provisioning decision based on the observations on the backlogs of the waiting queue. At every decision time point, if there are k users in the waiting queue, the CGSP will provision k/2 additional cloud servers in the next period; Therefore, we have seven methods in comparison, namely, (1) PRD + SSP; (2) LRD + SSP; (3) PRD + QSP; (4) LRD + QSP; (5) HRD + SSP; (6) HRD + QSP; (7) our proposed iCloudAccess algorithm.

B. Comparison of Gaming Latency

From the results in Fig. 9(a), we can clearly observe that our proposed *iCloudAccess* can reduce the mean and variance of queueing delay significantly.

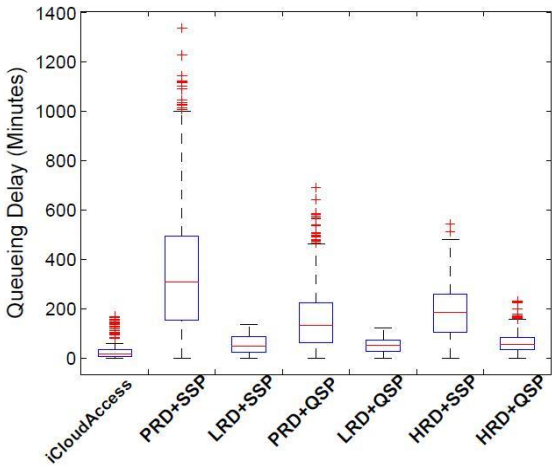

Fig. 9a. Queueing Delay

Among five methods, PRD+SSP perform the worst due to the serious queueing problem caused by proximity-aware request dispatching in the hot regions. Our iCloudAccess method achieves the lowest average queueing delay and the variance of queueing delay is also small. Most of the outliers occur in the early stage of the experiment. Fig. 9(b) plots the response delay incurred by different methods. Two methods (PRD+SSP, PRD+QSP) with proximity-aware request dispatching (PRD) achieve the lowest response delay as they always dispatch requests to the data center with the lowest network delay.

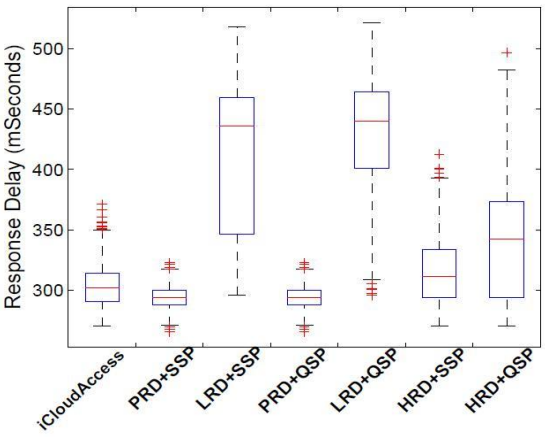

Fig. 9b.Response Delay

However, PRD may cause serious queueing delay in the hot regions if the paces of server provisioning cannot keep up with the increase of user demand (one example is PRD+SSP in Fig. 9(a)). The response delay of LRD+SSP and LRD+QSP is high because the selection of data centers is ignorant of proximity (or latency). Our proposed iCloudAccess approaches the performance of PRD+SSP and PRD+QSP with very small distance (normally with an increase of 10-20 milliseconds).

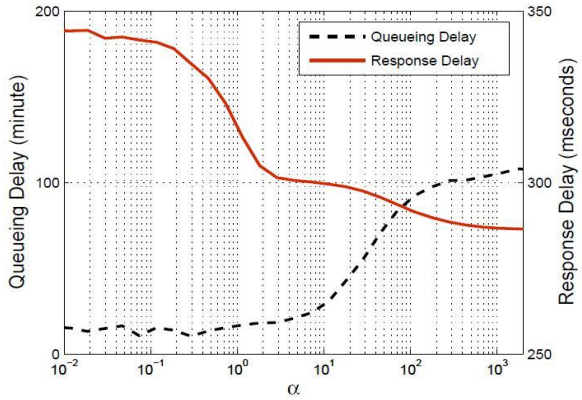

Fig. 10. The impacts of tunable parameter α on queueing delay and response delay

To examine the impacts of different values of α in the QoE function, we conducted additional experiments with various values of α. From the results in Fig. 10, we can observe that, with the increase of α, the response delay decreases significantly, while the queueing delay increases in the meanwhile. PRD and LRD can be thought as two special case of HRD by varying the weight.

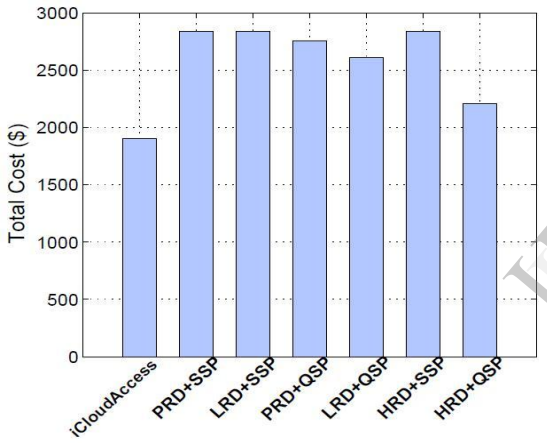

Fig. 11. Comparison of server provisioning cost

Fig. 11 depicts the total server provisioning cost incurred by different methods during the simulation period. Our simulation lasts for 3000 minutes. PRD+SSP and LRD+SSP incur the highest provisioning cost, and our proposed iCloudAccess has the lowest provisioning cost. More accurately, our method can reduce 33% of provisioning cost compared with PRD+SSP, LRD+SSP and HRD+SSP, 31% of provisioning cost compared to PRD+QSP, 27% of provisioning cost compared with LRD+QSP, and 16% of provisioning cost compared with HRD + QSP. Our algorithm will prefer to improve the user QoE, but it is at the cost of a higher server provisioning cost. On the contrary,

VI .CONCLUSION AND FUTURE WORKS

The on-demand feature of cloud gaming enables users to play gaming without the hardware and software constraints. In this paper, we study the latency problem of a multi-region multi-datacenter cloud gaming system. We first conduct a series of active and passive measurements on a large-scale cloud gaming service offering in China. From the measurement results, we observe that users suffer from high diversity in queueing delay and response delay. To optimize the user experience and minimize the operational cost of cloud gaming service providers simultaneously. Our proposed approach can significantly cut down the operational cost and reduce the latency at the same time. The work in this paper can provide useful guidelines for cloud gaming service providers to provision their services effectively. In the future, we plan to investigate the heterogeneity of user QoE requirements and study how to further optimize the cloud gaming infrastructure.

ACKNOWLEDGEMENT

We would like to thank the editors and anonymous reviewers for their valuable comments and helpful suggestions.

REFERENCES

[1] Distribution and monetization strategies to increase revenues from cloud gaming, July 2012.http://www.cgconfusa.com/report/documents/Content-5minCloudGamingReportHighlights.pdf.
[2] C. Chambers, W.-C. Feng, S. Sahu, D. Saha, and D. Brandt, "Characterizing online games," IEEE/ACM Transactions on Networking (TON), vol. 18, no. 3, pp. 899–910, 2010.
[3] C.-Y. Huang, C.-H. Hsu, Y.-C. Chang, and K.-T. Chen, "GamingAnywhere:An open cloud gaming system," in Proceedings of ACM Multimedia Systems 2013, Feb 2013.
[4] N. Tolia, D. Andersen, and M. Satyanarayanan, "Quantifying interactive user experience on thin clients," IEEE Computer, vol. 39, no. 3, pp. 46–52, 2006.
[5] Y.-C. Chang, P.-H. Tseng, K.-T. Chen, and C.-L. Lei, "Understanding the performance of thin-client gaming," in Communications Quality and Reliability (CQR), 2011 IEEE International Workshop Technical Committee on. IEEE, 2011, pp. 1–6.
[6] X264 web page, July 2013. http://www.videolan.org/developers/x264.html.
[7] R. Frederick and V. Jacobson, "Rtp: A transport protocol for real-time applications," IETF RFC3550, 2003.
[8] H. Schulzrinne, "Real time streaming protocol (rtsp)," 1998.
[9] Onlive. Homepage. http://www.onlive.com/.
[10] Gaikai. Homepage. http://www.gaikai.com/.
[11] CloudUnion. Homepage. http://www.yxyun.com/.
[12] K. Chen, Y. Chang, P. Tseng, C. Huang, and C. Lei, "Measuring the latency of cloud gaming systems," in Proceedings of the 19th ACM international conference on Multimedia. ACM, 2011, pp. 1269–1272.
[13] H. Lagar-Cavilla, J. Whitney, A. Scannell, P. Patchin, S. Rumble, E. D.Lara, M. Brudno, and M. Satyanarayanan, "Snowflock: rapid virtual machine cloning for cloud computing," in Proceedings of the 4th ACM European conference on Computer systems. ACM, 2009, pp. 1–12.
[14] L. Shi, M. Banikazemi, and Q. Wang, "Iceberg: An image streamerfor space and time efficient provisioning of virtual machines," in Proceedings of the International Conference on Parallel Processing-Workshops, 2008, pp. 31–38.
[15] J. Zhu, Z. Jiang, and Z. Xiao, "Twinkle: A fast resource provisioning mechanism for internet services," in Proceedings of IEEE INFOCOM.IEEE, 2011, pp. 802–810.
[16] Z. Xue, D. Wu, and J. He, "A measurement study of a largescale commercial cloud gaming system," Department of Computer Science, Sun Yat-sen University, Tech. Rep., 2013. [Online]. Available:http://sist.sysu.edu.cn/_dwu/cloudunion-tr.pdf
[17] M. J. Neely, Stochastic network optimization with application to communication and queueing systems. Morgan & Claypool Publishers, 2010.
[18] Y. Yao, L. Huang, A. Sharma, L. Golubchik, and M. Neely, "Data centers power reduction: A two time scale approach for delay tolerant workloads," in INFOCOM, 2012 Proceedings IEEE. IEEE, 2012, pp. 1431–1439.

[19] M. Kwok and G. Yeung, "Characterization of user behavior in a multiplayer online game," in Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology. ACM, 2005, pp. 69–74.

[20] Amazon EC2 pricing. http://aws.amazon.com/ec2/pricing/.