A New Approach to Scalable ROIA in Cloud

LIU Dong
Department of Computer Science,
JiNan University
Guangzhou 510632, China
Wze2k@163.com

ZHAO Yue-long*
School of Computer Science and Engineering,
South China University of Technology
Guangzhou 510641, China
ZhaoLab@126.com

***Abstract*—Real-time online interactive application (ROIA) is an emerging distributed application in recent years. Scalability is a hot and key issue in ROIA. We are motivated by the facts that the number of concurrent users around the world is widely and periodically fluctuant and requirements for network latency in various ROIA are different. So we propose the Multi-ROIA Cloud Platform (MRCP), a new approach to achieve high scalability in ROIA under Cloud environment. In this paper, we firstly make a classification of the traditional approach to scalability in ROIA and compare their merits and disadvantages. And then we propose MRCP, explain the motivation and describe the architecture and balance strategy of MRCP.**

Keywords-ROIA; MMOG; Scalability; MRCP

I. INTRODUCTION

Real-time Online Interactive Application (ROIA) is an emerging large-scale distributed application. Compared with other traditional distributed applications, ROIA has its own challenging features[1], such as highly intensive user interaction, concurrency at a single instance. The most typical application of ROIA is Massively Multiplayer Online Game (MMOG). MMOG, the main type of today's online games, has a huge user base and has become an important emerging industry. According to the statistic report, the scale of Chinese online game users has reached about 336 million as of December 2012[2]. But the number of ROIA users is very different between during the peak period and during off-peak period. If there is no appropriate scalable technology, the huge number of users in peak time will make the QoS (Quality of Service) reduction or even downtime. The effective scalable technology not only be able to comfortably cope with the pressure of peak time, can also improve resource utilization. So the scalability of ROIA is an important, difficult and hot issue.

In this paper, taking MMOG as a typical example of ROIA, we discuss the scalability of ROIA and propose the MRCP (Multi-ROIA Cloud Platform), a new approach to achieve high scalability in ROIA under Cloud environment.

The rest of paper is structured as follows. In Section II, we make a classification of the traditional approach to scalability in ROIA and compare the merit and shortcoming. In Section III, we propose a new approach to achieve high scalability in ROIA under Cloud environment. Paper explains the motivation and ideas of the new approach, describes its architecture and balance strategy. At last, in Section IV we give a conclusion of this paper.

II. TRADITIONAL APPROACH FOR SCALABILITY IN ROIA

The traditional approach for scalability in ROIA can be divided into the following three approaches:

A. C/MS Approach

Although Client/Server (C/S) model is an ancient architecture, it still is the mainstream deployment of ROIA[3]. For supporting more users concurrently, the server-side is more likely clustered servers rather than a single server. So some literatures call it Client/Multi-Server (C/MS)[4] .

1) Architecture of C/MS Approach

In this approach, the server-side consists of a number of servers. Figure 1 shows a typical C/MS architecture in MMOG.

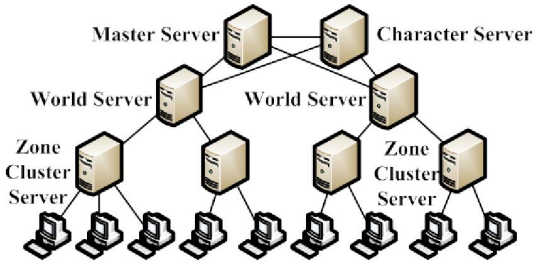

Figure 1. Client/Multi-Server Architecture

2) Scalable Technology of C/MS Approach

At present, C/MS approach is the main approach to achieve scalability in ROIA. Three expansion techniques are mainly used in C/MS approach: Zoning, Replication and Instancing[5].

a) Zoning

Zoning[6,7,8] based on the principle of data locality partitions the game world into several zones. The shape and size of each zone does not require to being the same, and it can be partitioned according to the requirements of server performance and load balancing.

For example, the game world is partitioned into four different sizes zones (i.e. A, B, C zone) in Figure 2, each zone supports by the different zone server. The avatars in the

* Corresponding author.

978-0-7695-5044-2/13 $26.00 © 2013 IEEE
DOI 10.1109/EIDWT.2013.13

same zone perceive each other, and the avatars in different zones cannot perceive each other directly. The transition between zones can only happen through certain portals (e.g., special doors) and requires an important amount of time. In zoning, the static fixed-size partitioning approach is simple but may occur with the flocking[31] (i.e., the surge in the number of avatars due to some emergencies in a zone). The dynamically changing the size of the zone according to load (e.g., the approach of [8]) can solve these problems. However, the dynamic partition needs to consider the problem of data migration and the migration processing of user actions which greatly increase the complexity.

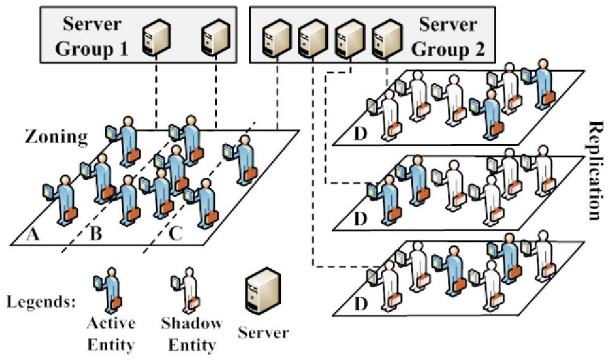

Figure 2. Zoning and Replication

In the zoning approach, the state consistency is relatively easier to maintain. Zone server maintains consistency within its own zone in update-based consistency model. When user wants to move in another zone, it needs master server and character server to assist the zone server to transmit the avatar's state information.

b) Replication

Replication[11,12] is a approach to supporting large number of avatars in one zone, as the D zone in Figure 2. The avatar density in D zone is very large and it is not suitable for re-partition into smaller zones, so the entire zone can be replicated in different servers. Each server computes and updates the state consistencies of part of the entities (known as Active Entity). And the state consistencies of other entities (known as Shadow Entity) are maintained through the inter-server synchronization.

In Figure 2, D zone is replicated to three servers, each server supports two active entities and four shadow entities. Through evenly distributing the works of state computing and maintenance to the three servers, it makes D zone can support larger number of avatars concurrently.

c) Instancing

Instancing[13] is a simplification of replication, which distributes the session load by starting multiple parallel instances of the highly populated zones. The instances are completely independent of each other, which means that two avatars form different instances will not see each other.

3) Merits and Disadvantages of C/MS Approach

The merits of C/MS approach are as follows:

a) Authority: A ROIA application needs to maintain the consistency of users interaction. For example, it needs to maintain the consistency of game world in MMOG. In C/MS, a centrailzed approach to management and storage of the global game state, and game provider can completely control their game with authority[14].

b) Security: The important processing logics are deployed on the server side in C/MS approach, so it easier to ensure security. And Server can be authoritative verify the effectiveness of the actions, it easier to prevent users from cheating in C/MS approach[15].

c) Simple and Mature Technology: As we know that the design and implementation of MMOG are difficult tasks, but C/MS approach at least relative to P2P approach easier to implement and deploy[16].

These merits make some developers even think that the C/MS approach is the only approach to MMOG[17]. However, in fact, the C/MS approach ROIA also has some disadvantages:

a) Single point of failure and Performance bottleneck: If the server failure or overload in C/MS approach, it will likely affect part or even all of clients.

b) Poor scalability: Due to the server is the bottleneck of system, so the processing capability of server-side determines the max number of concurrent users.

c) Infrastructure investment is large but utilization is low: Although the power of server-side can be expanded by the technigues of cluster servers, but such an extension lacks flexibilities. The game provider must reserve enough capacity of computing and storage to cope with the peak of the load. However, the peak period is relatively short, so the utilization of infrastructure is poor.

B. P2P Approach

It is because the C/MS approach ROIA has these shortcomings, researchers have proposed a variety structures and mechanisms of P2P approach ROIA[18,19,20].

1) Architecture of P2P Approach

According to the degree of centralization, we can classify P2P approach in three[21]: Purely decentralized, Partially centralized, Hybrid decentralized.

In purely decentralized P2P, peer must be connected to all other peers. It means that the network with n peers will have $\frac{n(n-1)}{2}$ connections. When n is increasing, the number of connections and the amount of transfer data will be increased exponentially. Moreover, it's difficult to maintain the consistency of the game world, so purely decentralized P2P is rarely used in actual applications.

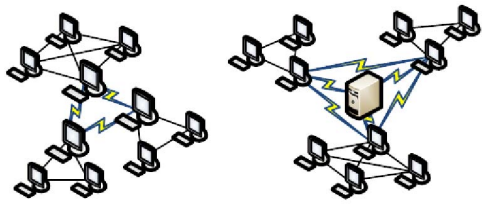

Figure 3. Architecture of P2P Approach

Most P2P approach use partially centralized or hybrid decentralized. Their architectures are shown in figure 3.

In partially centralized architecture, there are some of peers, called superpeers, act as local central servers maintaining some local information. In hybrid decentralized architecture, there exists a central server which maintains the metadata of users. The central server processes the log in requests, and helps user to found established peer group or create a new peer group. After user joins a peer group to start game, the game information will been transmitted between peers directly[19,22-25].

2) Merits and Disadvantages of P2P Approach

The merits of P2P approach are as follows:

a) Decrease of cost: In P2P approach, the user's compter takes part in computing and storage as a peer. For the MMOG provider, the overall cost can effectively reduce, which includes hardware costs, maintenance costs, network bandwidth costs and other costs.

b) Well scalability: With the increase of the number of users, the computing and storage capacity of the entire system is increased too.

*c) Good fault-tolerant capability:*Each peer can decide to join or quit the system freely. System structure can be self-organizing and even the load can be automatically balanced. So it has a good fault-tolerant capability.

However, to implementing the P2P model successfully still needs to address some challenges. For example, literature [26] run simulations using traces from World of Warcraft and found that pure P2P solutions would result in average latency more than 10x greater than C/S solutions. In addition, the absence of authoritative centralized computing and storage makes the state consistency, security and many other aspects to be more challenges. These areas still need much further research.

C. Cloud Approach

In recent years, cloud computing has been successfully adopted in many applications. And some researcher have begin to adopt in ROIA to address scalability problem and other issues.

Now the main cloud approach is a technology called Cloud Gaming. The typical representatives of Cloud Gaming are OnLive[27], Gaikai[28].

In this approach, all the game logics and graphics are rendered in the cloud rather than by the client. The cloud transfers the results by the way of video streaming. The user just need play the streaming like YouTube and need a very small thin client to send user's game command to the cloud.

A major attraction of this approach is that it frees players from the need to frequently upgrade their computers as they can now play games that host on remote servers. And the scalability and consistency is easy to achieve as all the game logics rendered in the cloud.

But because which is transferred to client is video streaming rather than update information in this approach, the network bandwidth becomes more important. When a player makes a move, he cannot see the results until the cloud receives the command, processes the move, renders a new screen, and delivers the screen to the client. Moreover, the traditional compensation techniques[29], such as dead reckoning[30], cannot be used in cloud gaming, because those techniques require game state information which are not available in cloud gaming clients.

Therefore, this approach is more susceptible to the response latency than other approach. Yeng-Ting Lee[31] also obtained the conclusion that not all games are suitable for cloud gaming by electromyographic experiments. The same degree of latency may have very different impact on a game's QoE (Quality of Experience).

III. NEW APPROACH FOR SCALABILITY IN ROIA

Through the analysis of the previous section of this paper, we can see that server side has to reserve sufficient resources to cope with the rush hour load, while these resources may be idle during off-peak times in the C/MS approach.

The P2P approach theoretically has well scalability, but has a number of challenges in safety, cheating, consistency control. Cloud Gaming is a good idea of adopting cloud computing in ROIA, but this approach is not suitable for all ROIA, especially for latency require higher ROIA.

So we propose a new approach to scalable ROIA in Cloud, we call it MRCP (Multi-ROIA Cloud Platform).

A. Motivation of MRCP

The MRCP is proposed mainly based on the following two points to consider.

1) Fluctuations in the Number of Concurrent ROIA Users around the World at Different Times

The number of concurrent users is not fixed, and it will fluctuate with the time every day. We can use this point, an overload data center can migrate some load to the low-load data center located in other time zone. Thereby this approach can obtain well scalability and high resource utilization.

2） Differences in Requirements for Network Latency in Different Type ROIA

ROIA can be divided into different types of ROIA. And the different type ROIA has different requirement for network latency in actual operation.

For example, MMOG can be divided into different types, such as First Person Shooting (FPS), Role Playing Game (RPG) and Real-time Strategy Game (RTS). These different MMOG have different requirements for network latency in order to ensure good user experience. According to the results of literature [32], the latency tolerance threshold is about 100ms in FPS, the threshold is about 500ms in RPG, and the threshold for RTS is about 1000ms.

So based on this character, we can deploy a variety type of ROIA on one data center, so that the load balancing has more reasonable choice. Because if only one type of ROIA, the following situation may be happen. The data center A is overloading, and the data center B located in other place (maybe on the other side of the earth) has very low load. However, due to B is so far from A that the latency is hard to meet the requirements of users on A. So it cannot migrate some load of A to B. In this case, if there are various types of ROIA on A, we can migrate the low latency requirement

ROIA to B. This is not only reduces the load of A, and also does not affect the good user experience.

B. Architeture and Balance Strategy of MRCP

Based on the above two considerations, we propose a Multi-ROIA Cloud Platform (MRCP). MRCP will have several data centers around the world, and various types of ROIA will be deployed on these data centers. According to the load of data centers, we can make load balancing between the various data centers, so as to achieve high scalability and high resource utilization.

1) MRCP's Architeture

Figure 4 depicts the architecture of MRCP.

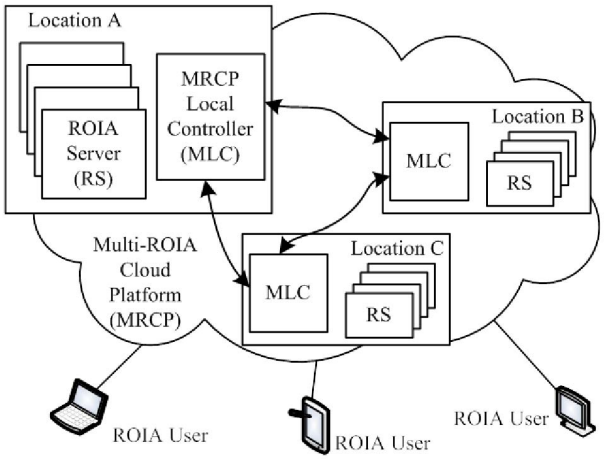

Figure 4. The Architecture of MRCP

MRCP consists of multiple data centers distributed around the world. Each data center has many ROIA Servers (RS) and one MRCP Local Controller (MLC).

MLC is responsible for user's login, storing user information, and the internal load balancing of data center. Moreover, MLC communication with other MLCs periodically in order to get the load information of other data centers. If necessary, MLC will execute load balancing between data centers.

RS is responsible for the functions of ROIA. Various types of ROIA are deployed on different RS in one data center. On RS, the traditional expansion technology, such as zoning, replication and instancing, can also be used.

2) MRCP's Balance Strategy

The balance strategy of MRCP is divided into the internal balance strategy which focuses on load balancing of RSs in one data center and the external balance strategy which focuses on load balancing between data centers. With regard to internal balance strategy, we can learn from C/MS approach, and previous scholars have many research results. So here we focus on the external balance strategy between data centers.

Firstly, we divide ROIA into three types according to their requirements for network latency.

a) Hight Demand Class: In the paper we call it H-Class for short. H-Class ROIA has higher requirement for network latency which should generally be less than 100ms, such as FPS.

b) Medium Demand Class: It is called M-Class for short. The latency requirement of M-Class ROIA should be less than 500ms generally. RPG is a example.

c) Low Demand Class: We call it L-Class for short. Such ROIA has lowest demand for network latency. Usually just less than 1000ms, it does not affect the good user experience. For example, RTS is a L-Class ROIA.

In MRCP, MLC monitors the load information of RSs in this data center, and estimates the overall load of this data center. There are two thresholds for overall load. $Threshold_{Low}$ represents that the overall load of this data center is low and this data center can receive more load or can shut down some servers to improve resource utilization. $Threshold_{Over}$ represents that this data center is over load and the internal load balancing cannot meet the performance requirements. It needs the load balancing between data centers.

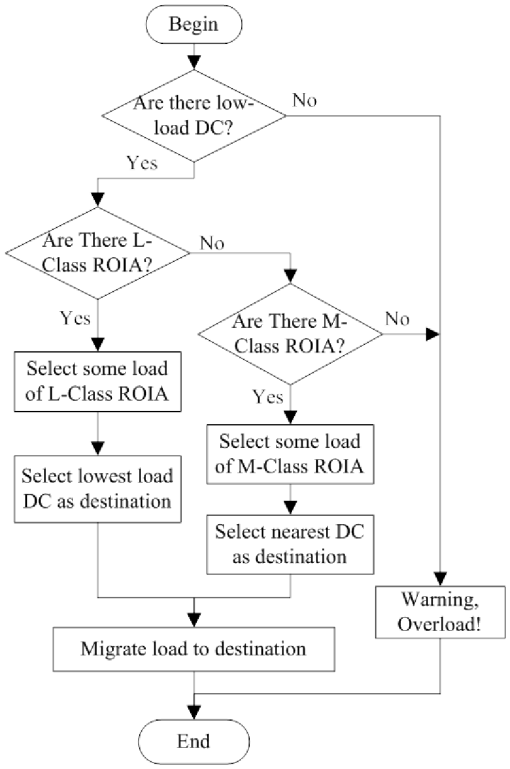

Figure 5. Balance Strategy between Data Centers

Figure 5 describes the balance strategy between data centers (DC). Load balancing between data centers begins when the local data center overall load is higher than $Threshold_{Over}$.

Firstly, MLC determines whether there is a low load DC according to the periodically received advertisement information. Due to the fluctuation in the number of concurrent users around the world, there will be low load DC generally.

Secondly, select the appropriate load from the L-Class ROIA and migrate it to lightest load DC. If there is no L-Class ROIA, then only select the appropriate load from M-Class ROIA. In this case, the load cannot be simply migrate

to lighter load DC. Because M-Class has higher demand for latency than L-Class, latency will be taken into account.

If DC does not have L-Class and M-Class more, then DC may overload. MRCP does not intend to migrate H-Class ROIA. It's because that H-Class has high demand for latency, and user usually will select the fastest server to log in. After migration, the latency often cannot meet the requirement. However, no L-Class and M-Class is an extreme case which can prevent by adjusting the ratio of L-Class and M-Class.

IV. CONCLUSCION

The previous researchers mostly focus on the extension techniques in one ROIA. And many research focus on determining how many servers are needed to handle player load. Few researchers take advantage of the mixed deployment of variety ROIA. So we propose a new solution idea which taken into account the fluctuations in the number of concurrent users and difference in requirements for network latency in various ROIA.

This paper present MRCP, a new approach to achieve scalability in ROIA under cloud computing environment. The infrastructure of MRCP is distributed across the world and various types of ROIA are deployed on it. We describe the architecture and balance strategy of MRCP. The balance strategy of MRCP is divided into the internal balance strategy and the external balance strategy. We focus on the external balance strategy in this paper. This paper sheds a little light to the relevant further research on ROIA.

ACKNOWLEDGMENT

This work is supported by Guangzhou Science and Technology Project (2010Y0-C681); Guangdong Science and Technology Project (2010B060100056); Natural Science Foundation of Guangdong (S2012010008831).

REFERENCES

[1] Frank Glinka, Allaithy Raed, Sergei Gorlatch, *et al*. "A Service-Oriented Interface for Highly Interactive Distributed Application," Proc. Euro-Par 2009 Workshops, LNCS 6043, 2010:266-277.

[2] CNNIC, "The 31th China Internet Development Statistics Report," http://www.cnnic.net.cn/hlwfzyj/hlwxzbg/hlwtjbg/201301/P020130122600399530412.pdf, 2013.1

[3] T. Alexander, "Massively Multiplayer Game Development," Charles River Media, ISBN: 1584503904, 2nd edition, 2005.

[4] Ginhung Wang, Kuochen Wang, "An Efficient Hybrid P2P MMOG Cloud Architecture for Dynamic Load Management," 2012

[5] Vlad Nae, Alexandru Iosup, *et al*. "Dynamic Resource Provisioning in Massively Multiplayer Online Games," IEEE Transaction on Parallel and Distributed System, VOL. 22, No. 3, 2011:380-395.

[6] W. Cai, P. Xavier, S.J. Turner, *et al*. "A Scalable Architecture for Supporting Interactive Games on the Internet," Proc. of 16th Workshop Parallel and Distributed Simulation, 2002:60-67.

[7] M. Assiotis, V. Tzanov, "A Distributed Architecture for MMORPG," Proc. of 5th ACM SIGCOMM Workshop on Network and System Support for Games, 2006:4.

[8] R. Chertov, S. Fahmy, "Optimistic Load Balancing in a Distributed Virtual Environment," Proc. of NOSSDAV'06, 2006:1-6.

[9] J. Chen, B. Wu, *et al*. "Locality Aware Dynamic Load Management for the Massively Multiplayer Games," Proc. of PPoPP'05, 2005:289-300

[10] R. Chertov, S. Fahmy, "Optimistic Load Balancing in a Distributed Virtual Environment," Proc. of NOSSDAV'06, 2006:1-6.

[11] J. Muller-Iden, S. Gorlatch, "Rokkatan: Scaling an RTS Game Design to the Massively Multiplayer Realm," Computers in Entertainment, Vol. 4, No. 3,2006:11.

[12] E. Cronin, B. Filstrup, A. R. Kurc, *et al*. "An Efficient Synchronization Mechanism for Mirrored Game Architectures," Proc. of 1st ACM SIGCOMM Workshop on Network and System Support for Games, 2002:67-73.

[13] C. GauthierDickey, D. Zappala, *et al*. "Low Latency and Cheat-proof Event Ordering for Peer-to-peer Games," Proc. of NOSSDAV'04, 2004:134-139.

[14] Lu Fan, "Solving Key Design Issues for Massively Multiplayer Online Games on Peer-to-Peer Arechitetures," Doctor Dissertation of Heriot-Watt University. 2009

[15] P.Kabus, W. W. Terpstra, M. Cilia, *et al*. "Addressing Cheating in Distributed MMOGs," Proc. of 4th ACM SIGCOMM Workshop on Network and System Support for Games, 2005:1-6.

[16] J. Mulligan and B. Patrovsky, "Developing Online Games – An Insider Guide," New Riders Publishing, ISBN:1592730000, 2003

[17] T. Alexander, "Massively Multiplayer Game Development," Charles River Media, ISBN: 1584503904, 2nd edition, 2005.

[18] Luo Jia, Chang Huiyou, *et al*., "Resource Assignment Alrorithm Under Multi-Agent for P2P MMOG," Joural of Computer Research and Development.47(12), 2010:2067-2074.

[19] L. Chan, K. Leng Yong, *et al*. "Hydra: A Peer-to-Peer Architecture for Massively Multiplayer Games," Proc. of 9th ACM SIGCOMM Workshop on Network and System Support for Games, 2007

[20] L. Fan, H. Taylorandp, Trinder, "Mediator: A Design Framework for P2P MMOGs," Proc. of 9th ACM SIGCOMM Workshop on Network and System Support for Games, 2007.

[21] L. Dong, Z. Yue-long, "Distributed Relational Data Sharing Based on P2P," Proc. of NISS'09, 2009:378-383..

[22] K. Kim, I. Yeom, and J. Lee, "Hyms: A Hybrid MMOG Server Architecture," IEICE Trans. Info. and Sys., vol. E87-D, no.12, 2004.

[23] Ignasi Barri, Francesc Gine, *et al*. "A Hybrid P2P System to Support MMORPG Playability," Proc. of International Conference on High Performance Computing and Communications, 2011:569-574.

[24] Ginhung Wang, Kuochen Wang, "An Efficient Hybrid P2P MMOG Cloud Architecture for Dynamic Load Management," Proc. of 2012 International Conference on Information Networking, Feb. 2012:199-204.

[25] DU Ling, SHI Xiang-bin *et al*. "Scalable P2P Overlay Architecture for MMOG," Journal of Chinese Computer Systems.33(1), 2012:58-63.

[26] John L. Miller, Jon Crowcroft, "The Near-Term Feasibility of P2P MMOG's," Proc. of 9th ACM SIGCOMM Workshop on Network and System Support for Games, 2010.

[27] http://www.onlive.com

[28] http://www.gaikai.com

[29] Y.W. Bernier, "Latency Compensating Methods in Client/Server in-game Protocol Design and Optimization," Proc. of 15th GDC, Mar 2001

[30] L. Pantel and L. C. Wolf, "On the suitability of dead reckoning schemes for gmes", Proc. of ACM NetGames'02, 2002

[31] Yeng-Ting Lee, Kuan-Ta Chen, Han-I Su, *et al.* "Are All Games Equally Cloud-Gaming-Friendly? An Electromyographic Approach," Proc. of ACM NetGames'12, 2012

[32] M. Claypool, K.T. Claypool, "Latency and Player Actions in Online Games," Communications of The ACM, Vol. 49, 2006: 40-45