The Brewing Storm in Cloud Gaming: A Measurement Study on Cloud to End-User Latency

Sharon Choy and Bernard Wong
University of Waterloo
{s2choy, bernard}@uwaterloo.ca

Gwendal Simon
Telecom Bretagne
gwendal.simon@telecom-bretagne.eu

Catherine Rosenberg
University of Waterloo
cath@uwaterloo.ca

***Abstract*—Cloud computing has been a revolutionary force in changing the way organizations deploy web applications and services. However, many of cloud computing's core design tenets, such as consolidating resources into a small number of datacenters and fine-grain partitioning of general purpose computing resources, conflict with an emerging class of multimedia applications that is highly latency sensitive and requires specialized hardware, such as graphic processing units (GPUs) and fast memory.**

In this paper, we look closely at one such application, namely, on-demand gaming (also known as cloud gaming), that has the potential to radically change the multi-billion dollar video game industry. We demonstrate through a large-scale measurement study that the current cloud computing infrastructure is unable to meet the strict latency requirements necessary for acceptable game play for many end-users, thus limiting the number of potential users for an on-demand gaming service. We further investigate the impact of augmenting the current cloud infrastructure with servers located near the end-users, such as those found in content distribution networks, and show that the user coverage significantly increases even with the addition of only a small number of servers.

I. Introduction

Cloud computing has, in the past few years, become the predominant environment for hosting web applications and services. Its rapid adoption largely stems from the intrinsic benefits of resource consolidation, such as higher resource utilization resulting in lower costs, and more elastic resource acquisition diminishing the need for accurate growth forecasting. However, in order to maximize the benefits of resource consolidation, most cloud providers only offer general purpose computing resources that are located in a relatively small number of large datacenters. Furthermore, many cloud datacenter locations are chosen to minimize cooling and electricity costs, rather than to minimize latency to end-users.

Unfortunately, these architectural decisions are in conflict with the needs of an emerging class of multimedia applications that is interactive (hence highly latency-sensitive), and requires specialized hardware resources, such as GPU and fast memory. Hosting these applications requires a new cloud paradigm.

In this paper, we select on-demand gaming as a representative case-study for this new class of applications. On-demand gaming, also known as cloud gaming, is a new video gaming application/platform. Instead of requiring end-users to have sufficiently powerful computers to play modern games, on-demand gaming performs the intensive game computation, including the game graphics generation, remotely with the resulting output streamed as a video back to the end-users. The shift from traditional gaming platforms, such as game consoles and desktop computers, to the cloud is the result of users wanting platform independence.

The two main technical challenges to on-demand gaming are linked to latency and the need for servers with expensive, specialized hardware that cannot simultaneously serve multiple gaming sessions. By offloading computation to a remote host, on-demand gaming suffers from encoding latency, that is, the time to compress the video output, and network latency, which is the delay in sending the user input and video output back and forth between the end-user and the cloud. Although the video encoding latency will likely fall with faster hardware encoders, a significant portion of network latency is unavoidable as it is bounded by the speed of light in fibre. Past studies [1]–[3] have found that players begin to notice a delay of 100 ms. However, 20 ms of this latency may be attributed to playout and processing delay. Therefore, 80 ms is the threshold network latency that begins to appreciably affect user experience. This strict latency requirement limits the potential user-base per server. This coupled with the cost for specialized hardware call for a solution based on multiple specialized servers distributed over the required coverage area.

To validate the need for a new cloud paradigm, we performed a large-scale measurement study consisting of latency measurements from PlanetLab and EC2 to more than 2,500 end-users in the US to determine the percentage of users that the existing cloud infrastructure can serve. We found that EC2 is capable of providing a median latency of 80 ms or less to fewer than 70% of our measured end-hosts. We also found that a substantial increase in the total number of datacenters is required to significantly increase user coverage. Adding datacenters specifically for on-demand gaming is therefore prohibitively expensive. These results suggest that the existing cloud infrastructure is a poor platform for hosting highly latency-sensitive applications, as a sizeable portion of the population would experience significantly degraded quality of

service. Additionally, we found that user-coverage increases by 28% when the deployment incorporates a small number of specialized servers that are near the end-users. Hence, we propose augmenting the existing cloud infrastructure by adding specialized hardware to the existing datacenters and a small number of dedicated, specialized servers that are distributed over the coverage area.

Overall, this paper makes three contributions. Firstly, it identifies and defines the challenges associated with on-demand gaming. Secondly, it demonstrates, through a large-scale measurement study, that more than one quarter of the population cannot play games from an EC2-powered cloud gaming platform. Finally, this paper presents a possible design that addresses the requirements of on-demand gaming, and shows that the addition of a small number of servers at the network edge can improve user coverage by more than 28%.

II. Background

In order to understand the latency experienced by the end-user, it is necessary to determine the response time of on-demand gaming. The interactive *response time* is defined as the elapsed time between when an action of the user is captured by the system and when the result of this trigger can be perceived by the user. Past studies have found on-demand gaming to be highly demanding with respect to response time [2]–[4]. The work in [5] demonstrates that a latency around 100 ms is highly recommended for dynamic, action games while response time of 150 ms is required for slower-paced games.

The overall interactive response time T of an application includes several types of delays, are defined as follows:

$$T = t_{client} + \overbrace{t_{access} + t_{isp} + t_{transit} + t_{datacenter}}^{t_{network}} + t_{server}$$

We define t_{client} as the *playout delay*, which is the time spent by the client to (i) send action information (*e.g.* initiating character movement in a game) and (ii) receive and play the video. Only the client's hardware is responsible for t_{client}.

Additionally, we define t_{server} as the *processing delay*, which refers to the time spent by the server to process the incoming information from the client, to generate the corresponding video information, and to transmit the information back to the client. For the purpose of gaming, the work in [6] shows that *processing delay* is affected by the amount of computational resources provisioned by the cloud provider, and this delay ranges from 10 ms to more than 30 ms. The cloud provider is responsible for the *processing delay*.

Both playout and processing delays can only be reduced with hardware changes. For our purposes, we optimistically estimate that playout and processing amount to 20 ms of delay; however, we recognize that this value can vary. By subtracting the 20 ms playout and processing delay from the target 100 ms latency, it confirms that 80 ms is the threshold network latency for on-demand gaming. The remaining contribution of total latency comes from the network. We further divide the network latency into four components: t_{access}, t_{isp}, $t_{transit}$, and $t_{datacenter}$.

Firstly, t_{access} is the data transmission time between the client's device and the first Internet-connected router. Three quarters of end-users who are equipped with a DSL connection experience a t_{access} greater than 10 ms when the network is idle [7], and the average access delay exceeds 40 ms on a loaded link [8]. The behavior of different network access technologies can greatly vary, as the latency of the access network can differ by a factor of three between different Internet Service Providers (ISP) [8]. Additionally, the home network configuration and the number of concurrent active computers per network access can double access link latency [9].

The second component of network delay is t_{isp}, which corresponds to the transmission time between the access router and the peering point connecting the ISP network to the next hop transit network. During this phase, data travels exclusively within the ISP network. Although ISP networks are generally fast and reliable, major ISPs have reported congestion due to the traffic generated by new multimedia services [10].

The third component is $t_{transit}$, which is defined as the delay from the first peering point to the front-end server of the datacenter. The ISP and cloud provider are responsible for $t_{transit}$; however, the networks along the path are often owned by third-party network providers. Nonetheless, the ISP and the cloud provider is responsible for to ensuring good network connectivity for their clients.

Lastly, $t_{datacenter}$ is defined as the transmission delay between the front-end server of the datacenter and the hosting server for the client. The cloud provider is responsible for $t_{datacenter}$. Network latencies between two servers in modern datacenters are typically below 1 ms [11].

III. A Reality Check on Infrastructures for On-Demand Gaming

In this section, we study the ability of today's cloud to offer on-demand gaming services. We focus on the network latency since the other latencies, especially the generation of game videos, have been studied in previous work [1], [12].

We conduct two measurement experiments to evaluate the performance and latency of cloud gaming services on existing cloud infrastructures in the US. Firstly, we perform a measurement campaign on the Amazon EC2 infrastructure during May 2012. Although EC2 is one of today's largest commercial clouds, our measurements show that it has some performance limitations. Secondly, we use PlanetLab [13] nodes to serve as additional datacenters in order to estimate the behavior of a larger, more geographically diverse cloud infrastructure.

A. Measurement Settings

As emphasized in previous network measurement papers [7], [8], it is challenging to determine a representative population of real clients in large scale measurement experiments. For our measurements, we utilize a set of $2{,}504$ IP addresses, which were collected from twelve different

BitTorrent [14] swarms which were participating in popular movie downloads.

We choose BitTorrent as the platform for our measurement experiments since we believe that BitTorrent provides a realistic representation of end-users and their geographic distribution. Using the *GeoIP* [15] service, we locate our collected peers, and we filter out end-users that are located outside of the US. The geographic distribution of the end-users, whom we refer to as *clients*, is illustrated in Figure 1. We believe this user distribution to be similar to the user distribution of on-demand gaming since the collected clients are also using their machines for recreational purposes. After determining the clients, we send TCP measurement probe messages in order to determine the latency from the server to the client. Note that we do not download content from BitTorrent participants as we only require the latency rather than the bandwidth.

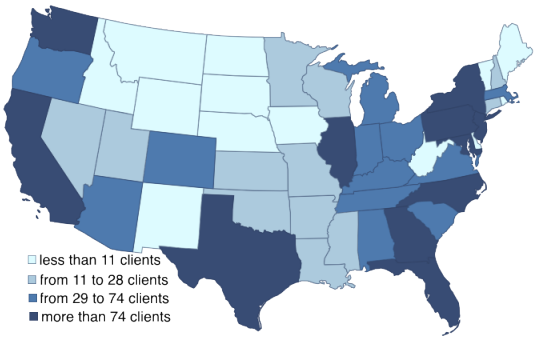

Fig. 1. Geographic distribution of clients.

One main advantage of retrieving IP addresses from a BitTorrent system is that BitTorrent tracker provides both an IP address and an open TCP port. By connecting to an open TCP port, we can measure the round-trip-time from the initial TCP handshake, which is more reliable than a traditional *ping*, since the *pings* are frequently filtered by network operators.

B. Case study: Amazon EC2

The Amazon EC2 cloud offers three datacenters in the US to its customers. We obtain a virtual machine instance in each of the three datacenters. Every 30 minutes, we measure the latency between each datacenter to all of the 2,504 clients. We use the median latency as the representative value in our measurements. Figure 2 depicts the ratio of covered end-users that have at least one network connection to one of the three datacenters with a latency below x ms. Two observations can be made from this graph:

- *More than one quarter of the population cannot play games from an EC2-powered cloud gaming platform*. The thin, vertical gray line in Figure 2 represents the 80 ms threshold network latency yielding a 70% coverage, which we deem unacceptable.
- *Almost 10% of the potential clients are essentially unreachable*. In our study, unreachable clients are clients that have a network latency over 160 ms, which renders them incapable of using an on-demand gaming service. Although we filter out the IP addresses that experienced highly variable latency results, we still observe that a significant proportion of the clients have a network latency over 160 ms. This result confirms the measurements made by previous work, which identified that home gateways can introduce a significant delay on data transmission [8].

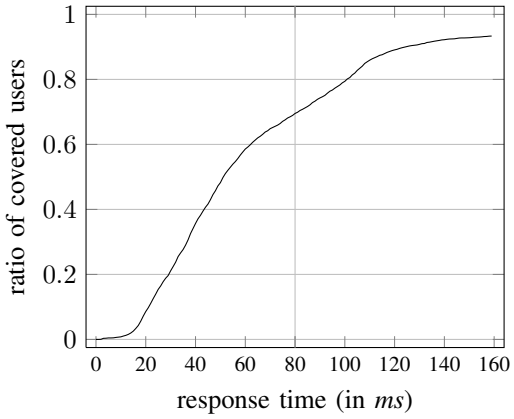

Fig. 2. Population covered by EC2 cloud infrastructure

C. Effects of a larger cloud infrastructure

The multi-billion dollar gaming industry has the potential to bring new cloud operators into the cloud computing market. An alternative to deploying a small number of large datacenters is to instead use a large number of smaller datacenters. Providers such as Gaikai [16] or Onlive [17] claim to possess up to a dozen of datacenters within the US [18], [19] in order to guarantee a better coverage of the population. Since a large datacenter is economically more efficient than a small datacenter, cloud providers should carefully determine if it is economically beneficial to build a new datacenter. In this section, we investigate the gain in population coverage when new datacenters are added into the existing EC2 infrastructure.

We create a simulator which uses our collected BitTorrent latencies in order to determine how many users are able to meet the latency-requirement for gaming. We utilize 44 geographically diverse PlanetLab nodes in the United States as possible locations for installing datacenters. We consider a cloud provider that can choose from the 44 locations to deploy a k-datacenter cloud infrastructure. Our simulator calculates the latencies between end-users and PlanetLab nodes, given our collected BitTorrent latencies. We design two strategies for deciding the location of datacenters:

- **Latency-based strategy**: the cloud provider wants to build a dedicated cloud infrastructure for interactive multimedia services. The network latency is the *only* driving criteria for the choice of the datacenter locations. For a given number k, the cloud provider places k datacenters such that the number of covered end-users is maximal.[1]

[1]When k is greater than four, we approximate the optimal results by taking the best k-subset out of five thousand randomly generated subsets.

- **Region-based strategy**: the cloud provider tries to distribute datacenters over an area. However, it takes into account various criteria for the location of its datacenters (for example: electricity cost, workforce, infrastructure quality and natural risks). We divide the US into four regions as set forth by the US Census Bureau: Northeast, Midwest, South, and West. Every datacenter is associated with its region. In every region, the cloud provider chooses *random* datacenter locations. For a given total number of datacenters k, either $\lfloor \frac{k}{4} \rfloor$ or $\lceil \frac{k}{4} \rceil$ datacenters are randomly picked in every region.

For cloud providers, the main concern is determining the minimum number of datacenters required to cover a significant portion of the target population. Figure 3 depicts the ratio of covered populations for two targets network latencies: 80 ms, which enable good response times for action games, and 40 ms for even more demanding games. We observe that a large number of datacenters is required if one wants to cover a significant proportion of the population. Typically, a cloud provider, which gives priority to latency, reaches a disappointing coverage ratio of 0.85 with ten datacenters for a target latency of 80 ms. Using the region-based strategy requires nine datacenters to reach a (poor) 0.8 ratio. In all cases, a 0.9 coverage ratio with a 80 ms response time is not achievable without a significant increase in the number of datacenters (around 20 datacenters). Similarly, one cannot expect the majority of the population to have a response time that is particularly suited for demanding games. Even if 20 datacenters are deployed, less than half of the population would have a response time of 40 ms.

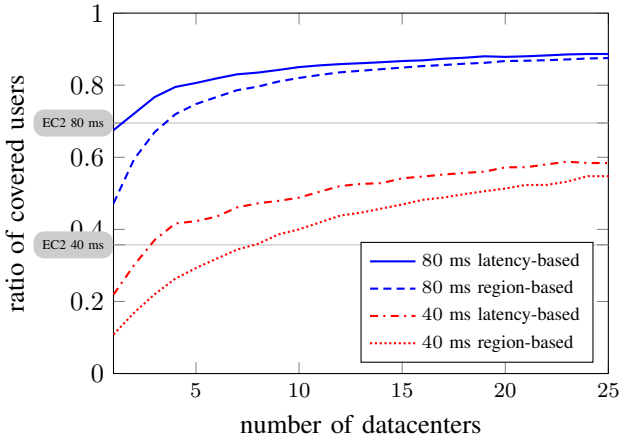

Fig. 3. Coverage vs. the number of deployed datacenters

We also emphasize that EC2 is a reasonable 3-datacenter deployment. Particularly, it performs as well as a latency-based 3-datacenter deployment for the 40 ms target response time. The performance of EC2 is also comparable to a region-based 3-datacenter deployment that targets 80 ms response time. The similarity between our measurements from PlanetLab and EC2 suggests that PlanetLab nodes can simulate datacenter sites.

We then focus on the performance of two typical cloud infrastructures: a 5 and 20-datacenter infrastructure. We assume a region-based location strategy since it is a realistic trade-off between cost and performance. We present the ratio of covered populations for both infrastructures in Figure 4.

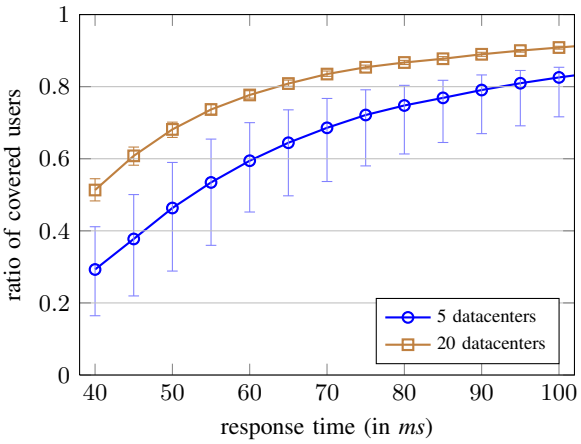

Fig. 4. User coverage for a region-based datacenter location strategy (average with min and max from every possible set of locations)

We observe that there can be significant performance gaps between a 5 and 20-datacenter deployment. Moreover, five datacenters do not guarantee good performance, despite the expectation that a region-based location strategy provides good coverage. Typically, a well-chosen 5-deployment can achieve 80% coverage for 80 ms. However, a poorly chosen 5-datacenter deployment can result in a disastrous 0.6 coverage ratio. In contrast, a 20-datacenter deployment exhibits insignificant variances in the coverage ratio.

IV. Moving to the Edge

As demonstrated in the previous section, the current datacenter scheme that serves on-demand gaming is not well suited to meet the latency requirements of end-users. One solution is to use existing CDN infrastructure. In particular, we can use CDN edge servers, which are located near end-users, to serve end-users. Unfortunately, current CDN edge servers lack computational resources for gaming as their purpose is to serve web-content. To address this issue, existing CDN servers can be enhanced with additional processing units and GPUs.

In this section, we explore the effectiveness of using such equipment, which we refer to as *smart edge*, in addition to using cloud resources. We demonstrate the potential of using additional resources to support existing infrastructure, and our experiments show that this augmented infrastructure increases the ratio of covered users. We determine the critical ratio of smart edges to clients, and we show that we can achieve a better coverage ratio if a smart edge can host more games.

A. Settings

Out of the 2,504 IP addresses collected in our measurement study, unless otherwise specified, we select at random 1500 clients and 300 smart edges that are chosen among a pool of 330 candidate smart edges. A smart edge is an additional,

non-cloud server that is used to serve a client for on-demand gaming. A candidate smart edge has the potential to be a smart edge; however, this is dependent on the supply of candidate smart edges in a given area. Each smart edge stores five applications. Because we do not know the latency between our collected IP addresses, we consider that a smart edge is actually located at its closest PlanetLab node, and we added an extra latency (randomly chosen between 0 to 15 ms) to the latency between the PlanetLab nodes and clients. The extra latency models inaccuracies in placing smart edge nodes at the PlanetLab node locations.

B. Potential of using the edge

We first explore the potential of an augmented infrastructure. Figure 5 presents the ratio of covered users by both EC2 and the combination of EC2 and smart edges. We consider here an idealized system without constraints on client-smart edge matching. In this scenario, there is only one game, and each smart edge can serve an unlimited number of clients.

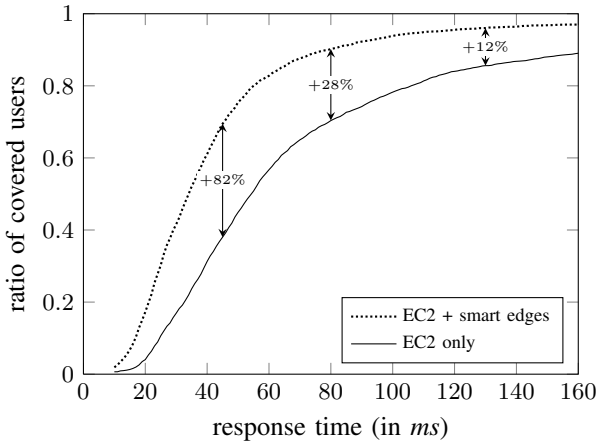

Fig. 5. Maximum achievable performances for an augmented infrastructure

These results demonstrate that significant gains can be achieved by the additional non-cloud servers. This nearly doubles the ratio of covered users whose target response time is below 45 ms and achieves a 28% increase in the ratio of users for a 80 ms target response time. Our results in Section III-C show that more than 20 datacenters are required to achieve a similar improvement.

C. The critical ratio of smart edges to clients

We now focus on the 80 ms target response time, and we consider the factors that affect the performance when additional servers are added to the existing datacenter infrastructure. Upon closer inspection, we can clearly determine whether clients are covered by EC2 or not. The *EC2-uncovered clients* can then also be differentiated between those who may be covered by an smart edge and those who are unreachable for a given response time.

Figure 6 illustrates the number of smart edges required to serve a given ratio of the population. We emphasize hereafter that an augmented infrastructure is bounded in two aspects.

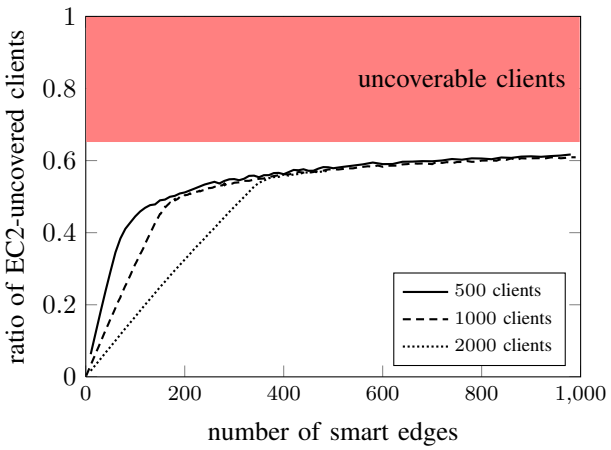

Fig. 6. Ratio of covered clients among the EC2-uncovered clients (for a catalogue of 100 games and each edge server stores 5 games)

- Given our dataset, out of the clients who cannot be served by a datacenter, only 65% can be covered by smart edges. The remaining clients exhibit excessive delay to all smart edges and datacenters, which is likely due to non-network delays that are outside of our system's control. Thus, the system's performance with respect to the ratio of covered clients is limited by this "ceiling".
- All clients that cannot be served by datacenters must be served by smart edges. Since we assume that a smart edge can serve only one client, there must be at least as many smart edges as there are clients that are not covered by datacenters.

Ideally, an augmented infrastructure would be able to achieve a perfect matching between the clients, which are not covered by EC2, and smart edges. However, smart edges host only a subset of games, and they are geographically distributed; therefore, not all smart edges can serve all clients.

We use 80 ms as our benchmark in our measurement study and experimentation. However, work such as [2] indicate that some game genres have more relaxed latency requirements. If the target network latency is relaxed from 80 ms to 100 ms, Figure 2 indicates that only 80% of the population can achieve the target latency, and Figure 5 demonstrates that the addition of smart edges can still improve the ratio of covered users.

V. Discussion

On-demand gaming is attractive to many end-users since it offers hardware independence by offloading computation to the cloud. Examples of on-demand gaming providers include On-live [17] and Gaikai [16]. Studies such as [1], [3], [20] demonstrate that short latency times are required in order to maintain an enjoyable user-experience; however, our simulations demonstrate that end-users geographically distant from these datacenters experience unacceptable latency. The serious financial difficulties experienced by OnLive [21] illustrate the need for a cheaper distribution infrastructure that improves client coverage and offers lower latency.

In addition to single-player games that OnLive and Gaikai provide, we also consider multi-player games that experience additional latency due to the coordination of different players. Therefore, multi-player games require lower network latency than single-player games. However, multi-player coordination latency is independent of on-demand gaming, and the existing solution of geographically segregating users can effectively reduce the coordination overhead of multi-player games.

Technologies for on-demand gaming are closely related to technologies for video streaming, because on-demand gaming consists of sending a video stream of the game back to the client. Current large-scale video-on-demand delivery solutions include dynamic datacenter provisioning (*e.g.* [22]) and peer-assisted delivery architectures (*e.g.* [23]). However, the goals of such systems are to effectively utilize bandwidth for video streaming applications so that service providers incur a lower bandwidth cost. Although cost-reduction is vital for the sustainability of a system, one must also consider latency beyond start-up latency when accessing video. The aforementioned mechanisms ensure that video is effectively distributed; however, these mechanisms may not be applicable for the gaming environment as games are far more latency-sensitive, and on-demand gaming cannot benefit from large video playback buffers to improve user experience.

The results of our measurement study points to a new cloud infrastructure that combines existing cloud datacenters with CDN servers. Furthermore, works such as [24], [25] confirm that edge-servers are not only used for serving static content and suggest that the use of the CDN edge may be viable for on-demand gaming. As studied in [26], current CDN infrastructure has the ability to serve millions of end-users and is well-positioned to deliver game content and software [27]. However, CDN edge-servers are generally built from commodity hardware that have relative weak computational capabilities and often lack GPUs. Moreover, although these servers are designed to serve thousands of simultaneous end-users, current virtualization technologies do not enable many instances of a game engine to run concurrently on the same physical machine [6]. Therefore, the transition to on-demand gaming requires CDN providers to modify their existing infrastructure, such as increasing CPU capacity and adding GPUs to CDN servers, to support the requirements of on-demand gaming.

VI. Conclusions

In this paper, we demonstrated through a large scale measurement study that existing cloud infrastructure is unable to meet the requirements of an emerging class of latency-sensitive multimedia applications. In particular, we demonstrated that there is a significant fraction of the population that is not sufficiently covered by Amazon's EC2 cloud infrastructure to meet the 80 ms latency requirement of on-demand gaming. To address these requirements, we proposed extending existing cloud infrastructure by equipping local content distribution servers with the necessary hardware to serve gaming demands. We demonstrated that an additional 28% of end-users can meet the 80 ms target response time if we use an augmented cloud infrastructure. By incorporating existing resources into cloud datacenters, we can significantly improve the viability of offering on-demand gaming.

References

[1] M. Jarschel, D. Schlosser, S. Scheuring, and T. Hoßfeld, "An Evaluation of QoE in Cloud Gaming Based on Subjective Tests," in *IMIS*, 2011.

[2] M. Claypool and K. T. Claypool, "Latency and player actions in online games," *Communications of The ACM*, vol. 49, pp. 40–45, 2006.

[3] M. Claypool and K. Claypool, "Latency can kill: precision and deadline in online games," in *MMSys*, 2010.

[4] T. Hoßfeld, R. Schatz, M. Varela, and C. Timmerer, "Challenges of QoE Management for Cloud Applications," *IEEE Comm. Mag.*, vol. 50, no. 4, Apr. 2012.

[5] M. Jarschel, D. Schlosser, S. Scheuring, and T. Hoßfeld, "Gaming in the clouds: QoE and the users' perspective," *Mathematical and Computer Modelling*, Dec. 2011.

[6] S. K. Barker and P. Shenoy, "Empirical Evaluation of Latency-sensitive Application Performance in the Cloud," in *MMSys*, 2010.

[7] M. Dischinger, A. Haeberlen, P. K. Gummadi, and S. Saroiu, "Characterizing residential broadband networks," in *IMC*, 2007.

[8] S. Sundaresan, W. de Donato, N. Feamster, R. Teixeira, S. Crawford, and A. Pescapè, "Broadband internet performance: a view from the gateway," in *Sigcomm*, 2011.

[9] L. DiCioccio, R. Teixeira, and C. Rosenberg, "Impact of home networks on end-to-end performance: controlled experiments," in *Sigcomm workshop on Home networks*, 2010.

[10] S. Higginbotham, "Smart TVs cause a net neutrality debate in S. Korea," Giga OM, Feb. 2012.

[11] S. M. Rumble, D. Ongaro, R. Stutsman, M. Rosenblum, and J. K. Ousterhout, "It's time for low latency," in *HotOS*, 2011.

[12] K.-T. Chen, Y.-C. Chang, P.-H. Tseng, C.-Y. Huang, , and C.-L. Lei, "Measuring the latency of cloud gaming systems," in *ACM Multimedia*, 2011.

[13] A. Bavier, M. Bowman, B. Chun, D. Culler, S. Karlin, S. Muir, L. Peterson, T. Roscoe, T. Spalink, and M. Wawrzoniak, "Operating System Support for Planetary-Scale Network Services," in *NSDI*, 2004.

[14] "Bittorrent," http://www.bittorrent.com/.

[15] "Maxmind - geoip python api," http://www.maxmind.com/app/python.

[16] "Gaikai open cloud gaming platform," http://www.gaikai.com.

[17] "Play on-demand video games over the internet," http://www.onlive.com/.

[18] "Gaikai will be fee-free, utilize 300 data centers in the us," http://www.joystiq.com/2010/03/11/gaikai-will-be-fee-free-utilize-300-data-centers-in-the-us/.

[19] "Gdc09 interview: Onlive founder steve perlman wants you to be skeptical," http://www.joystiq.com/2009/04/01/gdc09-interview-onlive-founder-steve-perlman-wants-you-to-be-sk.

[20] K. Chen, P. Huang, G. Wang, C. Huang, and C. Lei, "On the Sensitivity of Online Game Playing Time to Network QoS," in *INFOCOM*, 2006.

[21] K. Stuart, "Why onlive's brave venture failed," The Guardian, 2012, http://www.guardian.co.uk/technology/gamesblog/2012/aug/21/what-happened-to-onlive.

[22] D. Niu, H. Xu, B. Li, and S. Zhao, "Quality-assured cloud bandwidth auto-scaling for video-on-demand applications," in *INFOCOM*, 2012.

[23] W. Wu and J. C. S. Lui, "Exploring the optimal replication strategy in P2P-VoD systems: Characterization and evaluation," in *INFOCOM*, 2011.

[24] A. Leff and J. T. Rayfield, "Alternative edge-server architectures for enterprise javabeans applications," in *Proceedings of the 5th ACM/IFIP/USENIX international conference on Middleware*, ser. Middleware '04. New York, NY, USA: Springer-Verlag New York, Inc., 2004, pp. 195–211.

[25] M. Desertot, C. Escoffier, and D. Donsez, "Towards an autonomic approach for edge computing: Research articles," *Concurr. Comput. : Pract. Exper.*, vol. 19, no. 14, pp. 1901–1916, Sep. 2007.

[26] A. Passarella, "Review: A survey on content-centric technologies for the current internet: Cdn and p2p solutions," *Comput. Commun.*, vol. 35, no. 1, pp. 1–32, Jan. 2012. [Online]. Available: http://dx.doi.org/10.1016/j.comcom.2011.10.005

[27] K. Alexander, "Fat client game streaming or cloud gaming," Akamai Blog, Aug. 2012, https://blogs.akamai.com/2012/08/part-2-fat-client-game-streaming-or-cloud-gaming.html.