Cloud Strife: Expanding the Horizons of Cloud Gaming Services

Joseph Doyle*, Syed Islam*, Rabih Bashroush* and Donal O'Mahony†
*School of Architecture, Computing and Engineering,
University of East London, United Kingdom
Email: j.doyle@uel.ac.uk, syed.islam@uel.ac.uk, r.bashroush@qub.ac.uk
†Computer Science Department, Trinity College Dublin, Ireland
Email: Donal.OMahony@cs.tcd.ie

***Abstract*—Cloud gaming where computer games are executed on cloud based resources is a new service which can be extremely challenging for service providers. Unlike other web and video streaming applications it requires both a relatively high constant downlink bandwidth and low latency simultaneously. Hence, cloud gaming providers recommend that clients be relatively close to data centers to ensure reasonable Quality of Service. Latency, however, is also dependent on the compression type used for the video streaming and there is a trade-off between compression efficiency and complexity which affects client computer capability. In this paper we propose the `Strife` system which examines the processing capabilities of the client as well as the network conditions between the client and data center to select the compression type that maximizes the size of the client base for cloud gaming services for which reasonable Quality of Service can be provided. We model a cloud gaming service using data which reflects gaming hardware in common use, the bandwidth available in the regions and optimize the number of users that can receive reasonable Quality of Service. Finally, we compare other schemes with the `Strife` system to evaluate its performance.**

I. Introduction

Cloud based services providers (CBSP) are now offering a wide range of services and applications using the cloud computing paradigm. Some of these (such as email) utilize few networking resources. Others such as video streaming can strain these resources to their limits. As the networking requirements for a service increases it becomes more difficult for the provider to maintain a reasonable quality of service (QoS).

Recently, a new type of cloud service has emerged which can be extremely challenging for providers. This service, known as cloud gaming, functions by executing a game in a data center and streaming its video to the user in high definition (HD). The gamer can use any electronic device which has a high speed internet connection and can display HD video. Hence, it is therefore no longer dependent on specific gaming hardware. While games have used network resources before, they have traditionally only relied on them to provide multiplayer functionality. In this setup several clients are connected to one server or host. The host receives input commands, uses them to alter the game state and sends status updates to the clients so that they can update their local version of the gaming state. Input commands and status updates are usually quite small and hence, the network traffic required for this application is also small. In *Cloud Gaming*, however, the entire video rendered as part of the game must be delivered to the client across the network. This is what differentiates *Cloud Gaming* from the more traditional use of network connectivity by games.

It is predicted that the size of the global computer gaming market revenue will grow to 78 billion U.S. dollars in 2017, among which cloud gaming is expected to be the main portion [1]. Hence, there is significant financial incentive to overcome the networking challenges in operating this service so that users have a QoS which is consistent with a satisfactory gaming experience. Cloud gaming differs from other video streaming and web application in requiring both a relatively high constant downlink bandwidth and low latency while other applications only require one of these. As a result two cloud gaming providers namely *Gaikai* [16] and *OnLive* [26] recommend that users have a minimum network connection speed of 5Mbps. In addition, *OnLive* recommends that the distance between a client and the closest data center should be less than 1600km. These requirements mean that many areas are unable to access cloud gaming services. In addition, it means that game cloud operators must have sufficient hardware at each data center to handle the local peak load as they are not able to redirect clients to different data centers.

The bandwidth required and latency[1] achieved are dependent on the type of compression used for video streaming as well as the hardware used to encode and decode the stream. If a single compression type is used for all cloud gaming clients, many will receive poor QoS that could be avoided with a different compression type tailored to their network and hardware conditions. In this paper we propose the `Strife` system which analyzes the processor hardware of the client and the networking conditions between the client and the data center to expand the client base of the cloud gaming service while maintaining a reasonable QoS. This will allow a larger client base for cloud gaming providers to be achieved and

[1]It should be noted that unless otherwise stated we use the term latency to refer to the sum of network latency and the latency caused by compression rather than the gaming latency. These aspects are components of gaming latency. Other aspects such as latency introduced by the graphic pipeline and display latency are assumed to be equal for both cloud gaming and traditional gaming. Hence, they are not affected by the network and can be ignored.

lessen the hardware requirements for peak load at each data center.

There are a number of video compression standards in common use such as H.262, MPEG-2 [3], H.263 [4], H.264/AVC [37], [30] and H.265 [23]. Each of these standards can be configured in a number of different ways and hence, there are a number of compression types for each standard. The H.264/AVC standard is of greatest interest in this paper as it is the prevailing standard used for HD video compression. In this standard there are a number of options which result in a trade-off between the compression efficiency and complexity. As the bandwidth required to stream the video decreases due to greater compression efficiency the delay increases as the hardware must handle the additional complexity. The `Strife` system exploits this by selecting compression options which results in low delay for clients which are distant and have connections with large bandwidths, as well as selecting compression options which result in high compression efficiency for clients which are close but have connections with small bandwidths. This paper makes the following contributions:

- We present the `Strife` system which attempts to expand the client base of cloud gaming services while maintaining a reasonable QoS.
- Data for the hardware and bandwidth used by gamers as well as a representative set of round trip times between various geographical regions is presented.
- The performance of the `Strife` system is evaluated for various scenarios using the data obtained.

II. Related Work

There have been a number of studies which attempt to quantify the effect of latency on game performance [7], [8], [10], [18], [28]. Claypool *et al.* detail how latency affects different types of games [13]. Recent work by Suznjevic *et al.*, however, suggest that certain aspects of role playing games (RPGs) games such as player vs player (PVP) combat may have stricter latency requirements [31] than the 500ms threshold given in [13]. Hoßfeld *et al.* examine how the shift from local hardware to the cloud will affect the QoS of games [19]. Ojala *et al.* examine the operation of a cloud gaming business [25].

The are considerable challenges in designing a gaming cloud and there has been some research into tackling these. Choy *et al.* describe the EdgeCloud architecture which is more geographically diverse and contains more specialized hardware than the standard cloud to improve performance [11]. Zhao *et al.* build a game cloud using a hybrid CPU/GPU cluster of Xeon E7-8870 CPUs and NVIDIA Tesla M2070Q GPUs [39]. Wu *et al.* propose a forward error correction framework for improving the performance of cloud gaming on mobile devices [38]. Liao *et al.* propose an open source cloud gaming platform which improves performance by implementing compressed graphic streaming [24].

Dynamic compression techniques are also used in Active Transcoding [17], [14] when a video stream or image is transcoded in the network to adapt to the networking conditions. This work, however, is concerned with applications that do not have strict latency requirements and hence tries to minimize the overall connection time rather than the latency for each packet. The additional decoding and encoding caused by Active Transcoding make it impractical for cloud gaming due to the strict latency requirements.

Netflix which is the leading subscription service for online movies and television shows uses the Dynamic Streaming over HTTP (DASH) protocol to ensure reasonable QoS [5]. In DASH the video is encoded at several quality levels and is divided into small "chunks" which are video sequences of no more than a few seconds in length. The client selects the chunk best suited to current network conditions. A DASH system would be difficult to implement in cloud gaming as several video chunks would have to be created in real-time.

A closely related field to this work is adaptive playout [15]. This is the alteration of the frame rate depending on the buffer size or buffer fill level to reduce delay in video streaming. A time scale modification algorithm such as Waveform Similarity Overlap-Add (WSOLA) [35] is used to preserve the pitch of the audio and hence, maintain a natural sound. Similar techniques cannot be used in cloud gaming as they would result in poor responsiveness in games.

III. Problem Formulation

Ideally there should be no noticeable difference between cloud gaming and traditional gaming. To achieve this graphics must be rendered at sufficiently high quality and the QoS must be such that the response time for the game is satisfactory. The exact conditions needed for reasonable QoS depend on the type of game and have been studied in great detail [13], [10], [8]. The two factors which affect the QoS[2] for game users are bandwidth and latency. If the bandwidth available is insufficient and congestion occurs, frames will be lost or delayed and the playback of the video will not be smooth. If the latency is too high the game will feel unresponsive.

Let N be the number of users and connections for the cloud gaming service. Let each connection i use a particular type of the H.264/AVC compression c_i which is denoted by an integer. The bandwidth required so that congestion does not occur $B_i(c_i)$ is a function of the particular type of compression. The latency achieved $L_i(c_i)$ is a function of the type of compression and the round trip time between the data center and the client. Let M be the maximum latency allowable before the QoS of the game is affected. The recommended value of M ranges from 80ms–1000ms [28], [9], [8], [11], [22], [13]. The related field of video conferencing, however, recommends that the one-way latency should be less than 150ms as values higher than this will result in a noticeable delay [32]. A compression type which results in a latency less than the maximum allowable value $L_i(c) \leq M$ and a

[2]It should be noted that the term Quality of Experience (QoE) is usually used to evaluate the performance of a game but it has been shown that the network QoS is strongly correlated with QoE [21]. In the context of this paper the terms are equivalent.

```
1  InitializeConnection()
2       p= GetProcessorDetailsFromClient()
3       l = ExamineLatency()
4       b= ExamineBandwidth()
5       c= SelectCompressionMethod(p,l,b)

6   SelectCompressionMethod(p,l,b)
7       for j = 1 : C
8         if (ol[p][i] + l) ≤ M
9           return i
10        endif
11      endfor
12      return 1

13  UpdateConnection
14      l= UpdateLatency()
15      b= UpdateBandwidth()
16      if (ol[p][c] + l) > M —— ob[c] > b
17        c= SelectCompressionMethod(p,l,b)
18      endif
```

Fig. 1. Pseudo-code for `Strife` System Functions

bandwidth less than the connection bandwidth of the client should yield a reasonable QoS. The unpredictable nature of available bandwidth along a shared path, however, makes it advisable to select a compression type that yields the smallest bandwidth that keeps the latency value from exceeding the maximum allowable value. This can be formulated as an optimization problem:

$$\begin{array}{lll} \text{minimize} & B_i(c_i) & \forall i \\ \text{subject to} & L_i(c_i) \leq M & \forall i \end{array}$$

This is a discrete optimization problem. The number of compressions types available C, however, is small enough that a brute force method can be used to solve this problem quickly.

IV. The Strife System

The pseudo-code for the `Strife` system is depicted in Figure 1. We use $ob[]$ to denote a table which stores the bandwidth required to decode a video stream which is encoded by a certain compression type. The table $ol[][]$ is a two dimensional table used to store the latency added by the compression when a specific processor is used to decode a video stream which is encoded by a certain compression type. We assume that $ol[][]$ is ordered across the second dimension from the smallest bandwidth required to allow the optimal solution to be selected as quickly as possible. We also assume the $ol[][]$ and $ob[]$ are populated by values which have been generated by tests using various processors by the cloud gaming provider. It is possible that a client may have hardware that the cloud gaming provider has not encountered and hence would not be able to select a compression type. This, however, can be prevented as most cloud gaming services use a client program. It would be possible to evaluate the processor of a client during client program installation if information on the client's processor is not contained in the tables. The tables could then be updated with this information.

When a user connects to a cloud gaming service to run a game the *InitializeConnection* function is called. This function first determines what processor the client is using. This is used as an index for the latency and bandwidth tables. The network latency is then determined using the PING utility, the time difference between the SYN request and SYN+ACK response in a TCP connection or similar method. The bandwidth available can then be determined by utilizing a tool such as Iperf [33]. The *SelectCompressionMethod* function is then called.

This function operates by examining the additional latency caused by the compression types and adding each value to the network latency. If this value is less than or equal to the maximum allowable latency this compression type is selected as it uses the least bandwidth while not violating the latency constraint. It should be noted that many games which utilize a network already measure latency so that it can be displayed to users. Hence, this latency measurement could be obtained as part of the game, reducing redundant communication overhead. If the latency constraint cannot be satisfied the compression type which uses the least bandwidth is selected. This is done as a fallback mechanism as Henderson *et al.* have shown that some players are tolerant of bad QoS [18].

Once the connection has been established its status must be periodically checked so that the system can react to changes in the network. While tools like Iperf can be used initially, as the game is loading and not being played they cannot be used to determine the bandwidth during the game as they flood the link with traffic and would therefore affect the QoS. Hence, a bandwidth value must be stored and regularly updated on the client. This value must be sent to the data center upon request. In addition, the time between the request being sent and the reply being received can be used to update the latency value. If the current compression type has too high a latency or requires too much bandwidth the *SelectCompressionMethod* is called. Otherwise, the compression type stays the same.

V. Analysis

In this section we examine various aspects of a gaming cloud. Data from *Gaikai* and *OnLive* was not available so we examined a hypothetical gaming cloud based on Amazon's EC2 [6] service as it is considered to be the largest public cloud provider today [11]. First we looked at the network latency between a number of different geographical regions and three data centers in the EC2 cloud. We then inspected the bandwidths available to clients of the cloud gaming service using data from the broadband testing site Net Index [27]. Next, we considered the hardware used by clients of the cloud gaming service using data from the digital game distribution platform *Steam* [34]. Finally, we investigated how latency is affected by different compression types and hardware decoders.

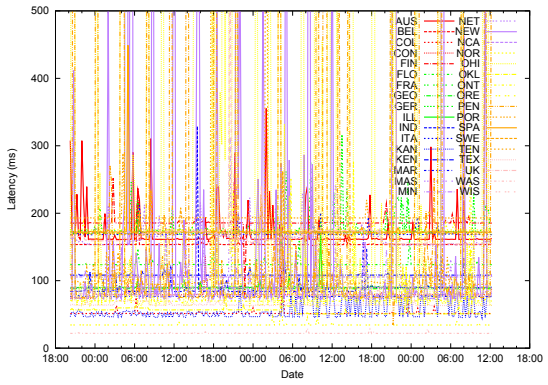


Fig. 2. Network latency between California and different geographical regions at fifteen minute intervals.

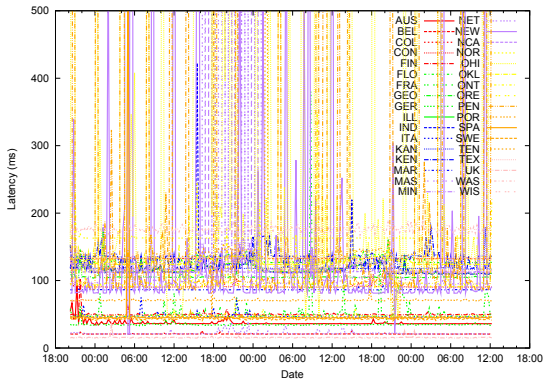


Fig. 3. Network latency between Ireland and different geographical regions at fifteen minute intervals.

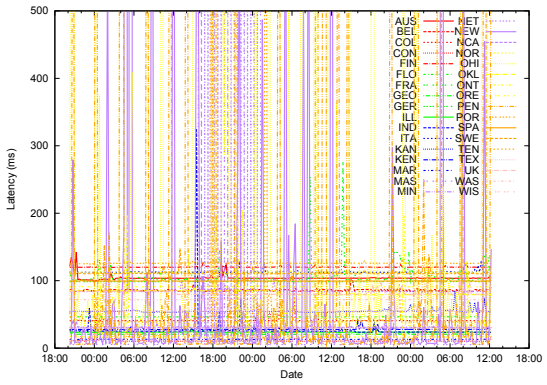


Fig. 4. Network latency between Virginia and different geographical regions at fifteen minute intervals.

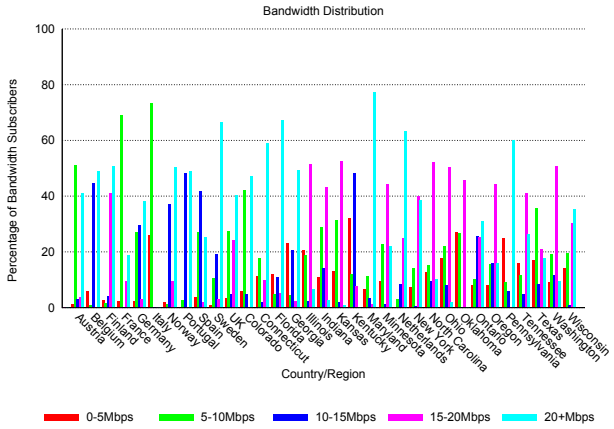


Fig. 5. Bandwidth distribution for internet subscribers in various regions.

This data will illustrate how the trade-off between compression complexity and efficiency can be exploited by the `Strife` system to improve cloud gaming performance.

A. Network Latency

Network latency is a key metric for the performance of a gaming cloud. To obtain the round trip times for our subset of the Internet, we established an experiment using EC2 and PlanetLab [12]. EC2 instances were instantiated in the California (US-West), Virginia (US-East) and Ireland (EU) data centers and a slice on a server in each region shown in Table I was established. The EC2 instances then pinged the other geographical regions at fifteen minute intervals for approximately three days. We used these values to represent the round trip times of cloud gaming clients connecting to the data centers. The average network latency established from this experiment can been seen in Table I.

Figure 2, 3 and 4 depict the measured latency between the data centers in California, Ireland and Virginia and all other regions.

B. Bandwidth

Another key performance metric for cloud gaming is the peak bandwidth available to cloud gaming clients. If the bandwidth required by a compression algorithm is higher than the peak bandwidth available to the client, congestion will occur and the client will not receive reasonable QoS. To analyze peak bandwidth available to cloud gaming clients we examine the bandwidth of broadband subscribers in the regions examined in the latency experiment. Data was obtained from Net Index [27] which stores bandwidth values received when broadband connections are tested for speed. This data contain an average bandwidth value for each internet service provider (ISP) in the region. The number of connections used to obtain this average value was also provided which allowed us to establish what proportion of the region's broadband subscribers used each ISP. We assume that cloud gaming users are divided proportionally among the ISPs in a region and hence, the proportion of broadband subscribers and cloud gaming users with a given bandwidth in a region is equal. The number of subscribers in the region ranged from approximately 80,000 to 3,700,000. As such the data can be viewed as representative of the bandwidth distribution in the region. The exact number of connections in each region can be seen in Table I. Figure 5 depicts the percentage of connections which fall within a certain range for each region.

From Figure 5 we can see that there is considerable difference in the bandwidth distribution between the regions. Less than 1% of the connections in Italy exceed 10Mbps while over 75% of connections exceed 20Mbps in Maryland. From this we can conclude that the high bandwidths of cloud gaming users in Maryland can be exploited by selecting a compression

TABLE I
AVERAGE ROUND TRIP TIME BETWEEN DATA CENTERS AND REGIONS ALONG WITH THE NUMBER OF SUBSCRIBERS IN THE REGION.

Region	California (ms)	Ireland (ms)	Virginia (ms)	Number of Subscribers
Austria (AUS)	170.43	37.31	104.22	314029
Belgium (BEL)	154.07	20.88	85.84	223434
Colorado (COL)	52.15	132.63	42.97	217839
Connecticut (CON)	79.73	91.42	12.87	103468
Finland (FIN)	186.01	49.83	120.22	301624
Florida (FLO)	124.67	130.94	46.39	570811
France (FRA)	177.64	52.29	106.20	748455
Georgia (GEO)	88.40	105.35	20.80	313153
Germany (GER)	172.92	34.86	98.70	902312
Illinois (ILL)	90.58	116.25	23.81	402247
Indiana (IND)	87.66	117.64	26.64	146207
Italy (ITA)	169.50	46.67	112.86	2300795
Kansas (KAN)	57.20	118.36	54.87	79851
Kentucky (KEN)	108.54	131.37	28.01	106611
Maryland (MAR)	78.41	86.45	13.00	163195
Massachusetts (MAS)	84.92	508.99	416.07	196875
Minnesota (MIN)	89.76	118.47	30.25	150725
Netherlands (NET)	164.40	21.94	84.27	641343
New York (NEW)	240.44	242.18	161.21	616925
North Carolina (NCA)	106.43	113.55	31.10	247469
Norway (NOR)	175.53	47.32	112.61	253186
Ohio (OHI)	480.27	515.11	448.27	281601
Oklahoma (OKL)	55.28	127.92	81.24	91922
Ontario (ONT)	77.02	98.73	21.41	849236
Oregon (ORE)	34.78	163.10	91.05	127325
Pennsylvania (PEN)	240.58	279.24	188.76	296524
Portugal (POR)	193.72	70.94	125.92	333551
Spain (SPA)	173.7	47.09	99.66	438632
Sweden (SWE)	171.04	43.41	110.83	125180
Tennessee (TEN)	110.82	137.58	53.29	129218
Texas (TEX)	64.85	134.08	55.29	925845
UK (UK)	166.45	15.60	81.80	3734513
Washington (WAS)	22.03	175.87	72.12	314324
Wisconsin (WIS)	79.72	118.28	30.48	125730

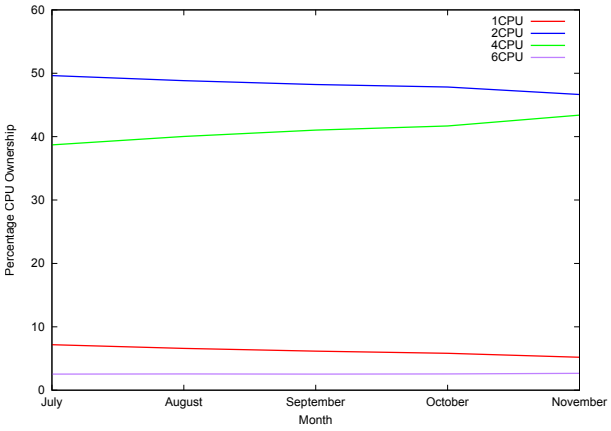


Fig. 6. Percentage of CPU Ownership for Steam Users.

type which has low latency requirements and sending service requests to distant data centers. This will allow cloud gaming providers to increase the utilization of their servers as peak load from different time zones can be serviced at the same data center. Little can be done, however, with regions like Italy as the low bandwidths mean that compression types with low bandwidth requirements must be used in conjunction with a nearby data center to attempt to prevent congestion.

C. Hardware

The hardware used by clients of cloud gaming services will have a significant effect on the latency caused by compression encoding and decoding. To this end we examine the hardware utilized by game users. We were unable to find data for cloud gaming services, but we were able to find hardware data for *Steam* which is a digital game distribution platform. It also functions as a subscription management service and users are usually logged in when they are playing games. Hence, the data is regularly updated and is representative of game users who have an internet connection. This is a similar group to cloud gaming users.

Figure 6 shows the percentage of game users with one, two, four and six core CPUs from July to November in 2012. From Figure 6 we can see that a broad range of processor hardware is utilised. It should be noted that it is also possible to use GPUs and dedicated hardware such as the Intel's "Quick Sync Video" [20] technology to decode video streams. If these are used, the decoding latency may be significantly smaller than the type of processor would normally suggest. As one of the main advantages of cloud gaming is that it can be utilized on any device that can display HD video and possesses an internet connection we focus our attention on the most popular hardware.

D. Compression

In this section we examine the latency caused by encoding and decoding the video stream for a variety of compression types and hardware. To evaluate this experimentally we recorded a raw video of the game "Mass Effect 2" which was available on *OnLive* and *Gaikai*. We recorded the video in the 1080p formats at 60 frames per second (fps). We choose the 1080p format, as it is the highest resolution supported by a large portion of GPUs as well as the XBox 1 and PlayStation 4 consoles. As such we postulate that it is the graphical quality enjoyed by a large portion of game users and cloud gaming providers will offer this format in the near future. This will help to achieve the aim of making cloud and traditional gaming indistinguishable. It should be noted that higher resolutions are possible but GPUs capable of this are priced for the "enthusiast" market and as such represent a small portion of the potential market.

We used a freely available H.264/AVC encoder known as x264 [36] to encode the video in a number of different ways using a six core CPU. We used x264 as it is a popular H.264/AVC encoder used by video streaming services such as YouTube. The six core CPU used was an Intel Xeon E5-2640. We choose to encode the video with a six core CPU as we assume that cloud gaming service providers will use high quality equipment at the server side to ensure that latency is as low as possible. The time required to encode the video was recorded for each compression type and this was used to calculate the encoding time for a single frame. The size of the encoded video was also noted so that the bandwidth required could be calculated. x264 has several presets which range from VerySlow to UltraFast. We used the VerySlow, Slower, Slow, Medium, SuperFast and UltraFast presets as our compression types. This was done to test a wide variety of compression types.

Each encoded video was then decoded using a six, four, two and single core CPU. The six core CPU was the same as the one used to encode the video stream. The four core CPU was an Intel Core i7 2600. The two core CPU was an Intel Xeon E3113. The single core was a 3.00GHz Intel Pentium 4. It should be noted that *Steam* users possess a wide variety of single, two, four and six core CPUs with different clock speeds and cache sizes. We did not have the resources to examine all of these so representative examples of the single, two, four and six core CPUs were examined. We used the H.264 decoder that is part of the ffmpeg [2] tool to decode the videos. The time required to decode the entire video was again noted so that the decoding time for a single frame could be calculated.

The encoding and decoding time for a 1080p format 60fps video for the various compression types for a single, two, four and six core CPU decoder are depicted in Figure 7(a), 7(b), 7(c) and 7(d), respectively. From Figure 7(a)-7(d) we see that as we move through the compression types the encoding and decoding times decrease. Figure 7(e) depicts the bandwidth required for each compression type for a 1080p format 60fps video. From Figure 7(e) we can see that the bandwidth required increases as we move through the compression types.

VI. Simulation Setup

In this section we describe the setup for the simulations used to evaluate the performance of the `Strife` system. The goal of `Strife` is to adapt the service to individual clients so that more clients of cloud gaming services receive reasonable QoS. In the simulations we examine how well QoS is maintained in the three data centers examined in Section V over a three day period while the number of users and latencies vary. The goal of these simulation is to investigate how many users receive reasonable QoS when the `Strife` system is used. This is then compared with other schemes.

In the simulations the clients for the cloud gaming service are spread throughout the regions detailed in Table I. We assume that the number of cloud gaming clients in each region corresponds to the number of broadband subscribers in the region given by Table I divided by the total number of broadband subscribers in our subset of the internet, multiplied by the total number of cloud gaming clients. Hence, we assume that the users are distributed proportionally across the regions. In the simulations we obtained data from *Steam* detailing the number of active users in hourly intervals over a four day period and we evaluate `Strife` at each interval during the first three days. We evaluated the first three days as only three days of latency data were available. This data is depicted in Figure 8. It is interesting to note that gaming usage follows a diurnal pattern like other cloud applications [29].

For the simulations we selected a value of 150ms for the maximum allowable latency before QoS is affected ($M = 150ms$). We selected this as many games experience significant performance problems once this value has been exceeded [13].

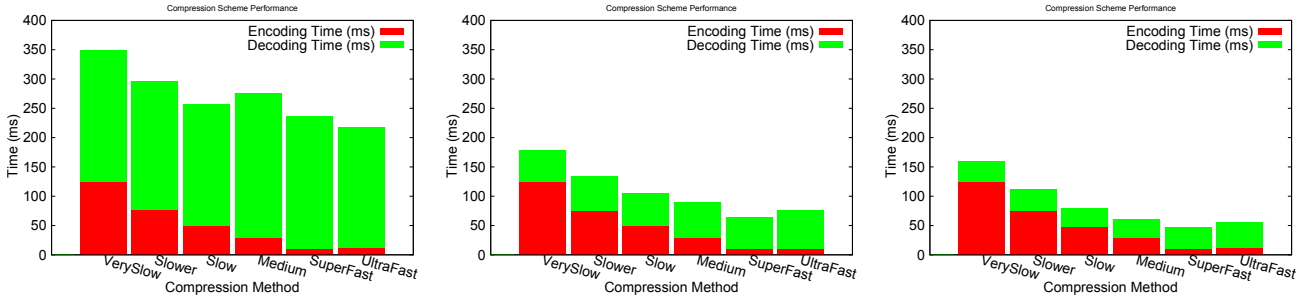
We assume that cloud gaming clients have bandwidths reflective of the data obtained from Net Index. In order to ensure the evaluation is as accurate as possible the exact values of bandwidth for each set of ISP connections are used and not the average of the ranges depicted in Figure 5. We use the data from Figure 2, 3 and 4 for the network latency to examine the systems dynamic performance. We also assume that the cloud gaming clients have hardware reflective of the data obtained from Steam.

To the authors' knowledge there is no system that adapts the compression types used to individual cloud gaming clients like the `Strife` system. We, therefore, compare `Strife` with a system that uses the same compression type for all clients. Hence, to evaluate the performance of the system we compare its performance with a system which uses the *Slower*, *Medium* and *UltraFast* preset for all connections.

VII. Results

The key performance metric for these simulations is the number of cloud gaming clients receiving reasonable QoS as defined in Section VI. We examine different data centers to show the `Strife` system's versatility.

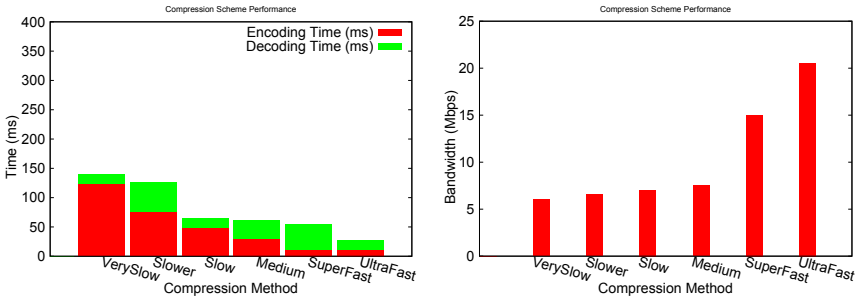
Figure 9(a), 9(b) and 9(c) depict the number of users receiving reasonable QoS over a two day period when the cloud gaming service is streaming 1080p format 60fps video from



(a) Encoding and decoding times for various compression types using a one core processor to decode

(b) Encoding and decoding times for various compression types using a two core processor to decode

(c) Encoding and decoding times for various compression types using a four core processor to decode



(d) Encoding and decoding times for various compression using a six core processor to decode

(e) Bandwidth required for various compression types to function without congestion.

Fig. 7. Performance of compression algorithm on 1080p format 60fps video with various hardware and compression types.

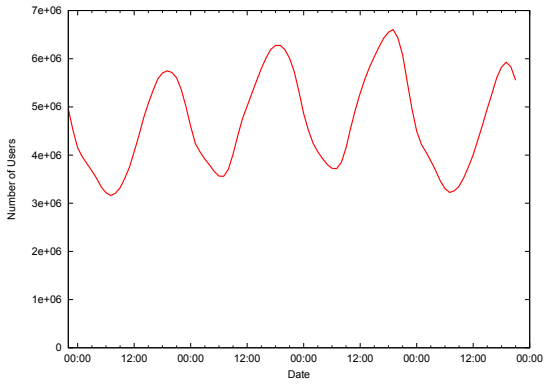


Fig. 8. Number of Steam Users online over a four day period. It shows UTC±00:00 local time.

the Ireland, Virginia and California data centers respectively. Each line represents the performance of a static compression scheme or the `Strife` system and are labeled as such in the legend. There is also a line which details the total number of clients requesting service at a given time so that the relative performance of the `Strife` system can be ascertained.

We can see from Figure 9(a), 9(b) and 9(c) that the `Strife` system allows additional clients to receive reasonable QoS at the data centers and this improvement is maintained as the latencies and total number of users change. From the figures we can see that the magnitude of the improvement varies depending on the static compression scheme to which the `Strife` system is compared and the network conditions at the data center. We can also see that the changing network conditions can be exploited to improve the performance of the `Strife` system from Figure 9(b). In the figure we can see that the difference between the Medium static compression scheme and the `Strife` system widens. The cause of this is the changing network conditions to which the `Strife` system can adapt.

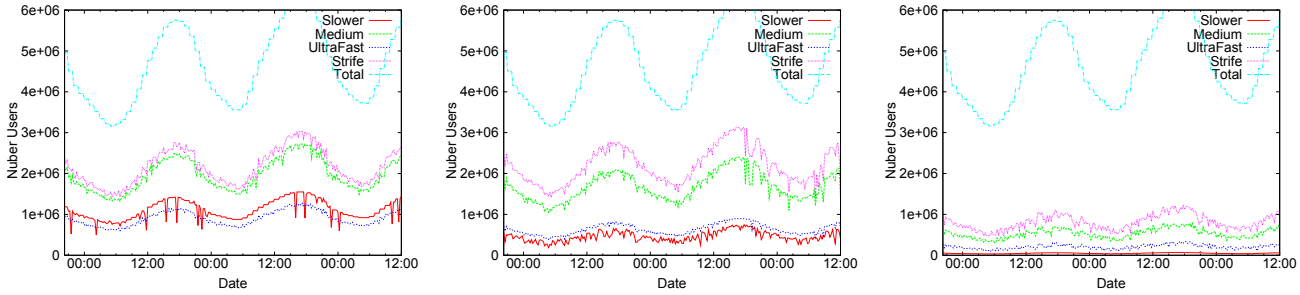
In all of these simulations we have shown that by analyzing the network conditions and hardware of the client base the `Strife` system can ensure more clients receive reasonable QoS when compared with static compression except in cases of extreme congestion.

VIII. Conclusions and Future Work

Cloud gaming is a new service which can be extremely challenging for service providers as it require a relatively high constant downlink bandwidth and low latency. Other network applications typically require only one of these. In this paper we have shown how the trade-off between compression efficiency and complexity can be exploited to improve the number of clients receiving reasonable QoS. We have shown that tens of thousands of additional clients can receive reasonable QoS from data centers if connections are adapted to individual clients. Thus, the cloud gaming service is more economically feasible as the cloud owners do not need to buy sufficient hardware at each data center to handle the local peak because some of the load can be directed to other data centers.

References

[1] Distribution and monetization strategies to increase revenues from cloud gaming. http://www.cgconfusa.com/report/ documents/Content-5minCloudGamingReportHighlights.pdf.



(a) Number of users receiving reasonable QoS at Dublin data center with 1080p format 60fps video

(b) Number of users receiving reasonable QoS at Virginia data center with 1080p format 60fps video

(c) Number of users receiving reasonable QoS at California data center with 1080p format 60fps video

Fig. 9. Number of users receiving reasonable QoS under various schemes over a three day period.

[2] ffmpeg. http://ffmpeg.org/.

[3] Generic Coding of Moving Pictures and Associated Audio Information–Part 2: Video. *ITU-T Rec. H.262 and ISO/IEC 13818-2 (MPEG-2 Video), ITU-T and ISO/IEC JTC 1*, November 1994.

[4] Video Coding for Low Bit Rate communication. *ITU-T Rec. H.263, ITU-T*, Version 1: November 1995, Version 2: January 1998, Version 3: November 2000.

[5] V. K. Adhikari, Y. Guo, F. Hao, M. Varvello, V. Hilt, M. Steiner, and Z.-L. Zhang. Unreeling Netflix: Understanding and Improving Multi-CDN Movie Delivery. In *Proceedings of INFOCOM*, pages 1620–1628, Orlando, 25 - 30 March 2012.

[6] Amazon. Elastic Compute Cloud. http://aws.amazon.com/ec2.

[7] G. Armitage. An experimental estimation of latency sensitivity multiplayer Quake 3. In *Proceedings of ICON*, pages 137–141, Sydney, 28 September - 1 October 2003.

[8] K.-T. Chean, Y.-C. Chang, P.-H. Tseng, C.-Y. Huang, and C.-L. Lei. Measureing the Latency of Cloud Gaming Systems. In *Proceedings of ACM MM*, pages 1269–1272, Scottsdale, 28 November - 1 December 2011.

[9] K.-T. Chen, P. Huang, and C.-L. Lei. How sensitive are online gamers to network quality? *Communications of the ACM*, 49(11):34–38, 2006.

[10] K.-T. Chen, P. Huang, G.-S. Wang, C.-Y. Huang, and C.-L. Lei. On the Sensitivity of Online Game Playing Time to Network QoS. In *Proceedings of INFOCOM*, pages 1–12, Barcelona, 23 - 29 April 2006.

[11] S. Choy, B. Wong, G. Simon, and C. Rosenberg. EdgeCloud: A New Hybrid Platform for On-Demand Gaming. Technical Report CS-2012-19, University of Waterloo, 2012.

[12] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, and M. Bowman. PlanetLab: an overlay testbed for broad-coverage services. *SIGCOMM Computer Communication Review*, 33(3):3–12, 2003.

[13] M. Claypool and K. Claypool. Latency and player actions in online games. *Commuications of the ACM*, 49(11):40–45, 2006.

[14] M. Dick, J. Brandt, V. Kahmann, and L. Wolf. Adaptive transcoding proxy architecture for video streaming in mobile networks. In *Proceedings of IEEE ICIP*, volume 3, pages III–700–3, Genoa, 11 - 14 September 2005.

[15] Eckehard Steinbach and Niko Färber and Bernd Girod. Adaptive Playout for Low Latency Video Streaming. In *Proceedings of the International Conference on Image Processing*, volume 1, pages 962–965, Thessaloniki, 7 - 10 October 2001.

[16] Gaikai. http://www.gaikai.com/.

[17] R. Han, P. Bhagwat, R. LaMaire, T. Mummert, V. Perret, and J. Rubas. Dynamic adaptation in an image transcoding proxy for mobile Web browsing. *IEEE Personal Communications,*, 5(6):8–17, 1998.

[18] T. Henderson and S. Bhatti. Networked games – a QoS-sensitive application for QoS-insensitive users? In *Proceedings of the ACM SIGCOMM Workshop RIPQoS*, pages 141–147, Karlsruhe, 25 - 29 August 2003.

[19] T. Hoßfeld, R. Schatz, M. Varela, and C. Timmerer. Challenges of QoE Management for Cloud Applications. *IEEE Communications Magazine*, 50(4):28–36, 2012.

[20] Intel. Quick Sync Video. http://www.intel.com/content/www/us/en/architecture-and-technology/quick-sync-video/quick-sync-video-general.html.

[21] M. Jarschel, D. Schlosser, S. Scheuring, and T. Hoßfeld. An Evaluation of QoE in Cloud Gaming Based on Subjective Tests. In *Proceedings of IEEE Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, pages 330–335, Seoul, 30 June - 2 July 2011.

[22] M. Jarschel, D. Schlosser, S. Scheuring, and T. Hoßfeld. Gaming in the clouds: QoE and the users' perspective. *Mathematical and Computer Modellings*, pages 1–12, 2011.

[23] H. Kalva. The H.264 Video Coding Standard. *IEEE MultiMedia*, 13(4):86–90, 2006.

[24] X. Liao, L. Lin, G. Tan, H. Jin, X. Yang, W. Zhang, and B. Li. Liverender: A cloud gaming system based on compressed graphics streaming. *IEEE/ACM Transactions on Networking*, 24(4):2128–2139, 2016.

[25] A. Ojala and P. Tryväinen. Developing Cloud Business Models: A Case Study on Cloud Gaming. *IEEE Software*, 28(4):42–47, 2011.

[26] OnLive. http://www.onlive.com/.

[27] Ookla. Net index. http://www.netindex.com/.

[28] L. Pantel and L. C. Wolf. On the impact of delay on real-time multiplayer games. In *Proceedings of NOSSDAV*, pages 23–29, Miami, 12 - 14 May 2002.

[29] A. Qureshi, J. Guttag, R. Weber, B. Maggs, and H. Balakrishnan. Cutting the electric bill for internet-scale systems. In *Proceedings of ACM SIGCOMM*, pages 123–134, Barcelona, 17-21 August 2009.

[30] H. Schwarz, D. Marpe, and T. Wiegand. Overview of the Scalable Video Coding Extension of the H.264/AVC Standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(9):1103–1120, 2007.

[31] M. Suznjevic, O. Dobrijevic, and M. Matijasevic. MMOPRG Player actions: Network performance, session patterns and latency requirements analysis. *Multimedia Tools and Applications*, 45(1-3):191–214, 2009.

[32] T. Szigeti and C. Hatting. *End-to-End QoS Network Design: Quality of Service in LANs, WANs and VPNs*. Cisco Press, Indianapolis, 2005.

[33] A. Tirumala, F. Qin, J. Dugan, J. Ferguson, and K. Gibbs. Iperf: The TCP/UDP bandwidth measurement tool. [Online]. Available: http://dast.nlanr.net/Projects/Iperf/.

[34] Valve. Steam. http://store.steampowered.com/.

[35] W. Verhelst and M. Roelands. An overlap-add technique based on waveform similarity (WSOLA) for high quality time-scale modification of speech. In *Proceedings of ICASSP*, volume 2, pages 554 – 557 vol. 2, Minneapolis, 27 - 30 April 1993.

[36] VideoLAN Organization. x264. http://www.videolan.org/developers/x264.html.

[37] T. Wiegand, G. J. Sullivan, G. Bjøtegaard, and A. Luthra. Overview of the H.264/AVC Video Coding Standard. *IEEE Tranactions of Circuits and Systems for Video Technology*, 13(7):560–576, 2003.

[38] J. Wu, C. Yuen, N. M. Cheung, J. Chen, and C. W. Chen. Streaming mobile cloud gaming video over tcp with adaptive source-fec coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(1):32–48, 2017.

[39] Z. Zhao, K. Hwang, and J. Villeta. Game Cloud Design with Virtualized CPU/GPU Servers and Initial Performance Result. In *Proceedings of ScienceCloud*, pages 23–30, Delft, 18 June 2012.