Paper Review for: **Dissecting Cloud Gaming Performance with DECAF**

Introduction

In recent times, utilization of cloud gaming platforms have gained strong traction. Cloud gaming gave gamers the ability to play games anywhere, independent of device specifications, without having to worry about updates, patches or upgrades. Although the demand of cloud gaming platforms grew, performance measurements crucial to quantifying the quality of experience (QoE) were lacking the ability to generalize performance measurements across game genres to deploy at scale, limited in their scope in leading to difficulty interpreting the quality of experience.

In this paper, the authors goals are to develop a systematic methodology that can be deployed at scale and allows better understanding and interpretable results. First we need to define that total game delay (GD) is the sum of the network's round trip (RTT) and the processing delay on the server side, which includes game logics and video encoding and the processing delay on the client side, which includes video decoding .Then they broke down the problem into two parts. First, by separately understanding the components of total game delay. Second, by understanding the effects of video streaming strategies.

Overview

In short, the DECAF steps are: First, manually playing the game to generate datasets, identifying actions that are measurable, manually labeling each frame. Using a simple CNN architecture, they were able to achieve 99% accuracy.

Next, they make use of a game bot to continuously generate, capture and timestamp each frame. All frames captured then run through the trained CNN model to classify and calculate the total game delay.

All games chosen in the experiments worked on browsers and could make use of the ReceiveStatisticsProxy module of Chromium to measure details of the client side delay and the Chromium WebRTC module to measure the network delay. Thus allowing the authors to derive the server side processing delays through subtraction from the total game delay.

Next, in order to understand the video stream performance, 2 broad categories of network situations were experimented. A situation with a bandwidth constrained network, by throttling the connection bandwidth at various levels and a congested network, where Linux netem utility were made use of to emulate packet losses at different levels. All experiments were tested with Ethernet connections

Interestingly, results showed that over 70% of the total game delay were caused by server processing delays and RTT delays only contributed to about 10% of total game delay. RTT and client side delays were relatively the same amongst various game genres and cloud gaming platforms. However, server side processing delays vary among game genres and cloud gaming platforms. This results challenges the previous benchmarks and models used to quantify cloud gaming platforms.

In the bandwidth constrained experiment, they found that Luna and GFN used the gradual ramp up approach while Stadia probed for available bandwidth, delivering at maximum bitrates when possible. While GFN and Stadia displayed high variation in bitrates, Luna provided a much stable bitrate throughout the experiments. All platforms do not deliver at maximum bandwidth limit, they do not deliver specified FPS or resolution even when sufficient bandwidth is available. Luna adopts a strict rule to 720p resolution when available bandwidth falls below 12.5Mbps. GFN adopts an even stricter delivery on resolution, at times falling to 540p.

In the congested network experiment, they realized that all platforms differ in handling packet losses and especially for GFN, displaying a performance cliff when packet loss increases from 0.1% to 0.5%. All platforms reduce their frame rate when packet loss is observed. However, GFN has the lowest game freezes, possibly due to their dramatic reduction in bitrate and stricter resolution constraints.

Bottleneck @ server processing time. Cloud provider needs to analyse server processing time.

What is one thing that stands out to u the most? Good and Bad

- Bitrate suffice for games with unequal graphics quality
    - Games across different generations

Bad:
- Fig 17 (a) plot x-axis confusing.
- Exploration of other metrics
- Qualitative experiments of how metrics' affect QOE
    - Might be due to platform target audience. GF game more time sensitive.
    - differences in server processing delay are significant; past work has shown that humans can perceive latency differences as small as 8.3 msec [55]. Various gaming interest blogs have highlighted the negative impact a few milliseconds of latency can have on user experience. For example, [16] states that an increase in latency by 20ms can make a game unplayable.

Points
- ~~Thin client hardware~~
- ~~Multiplayer games - local vs online~~
- All games are available on web platforms (?)
    - Methodology hinges on chrome webRTC

- Availability of games across all platforms
    - Crew for Luna, Crew2 for the rest

- Variability of RTT
    - Network delay measured by keep alive
    - Default keep alive interval is 75s

- Scalability (?)
    - Data labelling

- Bandwidth control at OVS level
    - Try to explore where the bandwidth limit is
    - Balance between local throttling and overall internet throughput

- Deployment of client
    - Explore RTT

- Explore Benchmark of games locally

https://www.nvidia.com/en-us/data-center/gpu-cloud-computing/amazon-web-services/
https://aws.amazon.com/nvidia/
https://www.spglobal.com/marketintelligence/en/news-insights/latest-news-headlines/amazon-s-cloud-gaming-service-poised-to-steal-google-s-thunder-8211-analysts-60530597\
https://www.thegamer.com/geforce-now-10-million-subscribers/
https://www.pocket-lint.com/games/news/google/143589-google-stadia-explained-price-release-date-games-list