Paper: Dissecting Cloud Gaming Performance with Decaf
Project: Project 1 – CPU Monitoring (Linux)
Team 11: Chong Yihui, Gan Wan Cheng Isaac, Goh Zhen Hao, Manish Kumar, Soedarsono

Summary

Cloud gaming platforms allow users to play video games without the need to invest in expensive high-end hardware. While the industry has experienced momentous growth in recent years, there is a lack of methodologies to measure and compare performance between platform providers. To address this, the paper proposes DECAF, a highly automated methodology that treats the platforms as black-box and measures their performance based on i) game delay and ii) robustness of video stream to network impairments. Game delay refers to the time taken between invoking an in-game action and observing its animation manifest on the screen. DECAF measures this by deploying a simple game bot to repeatedly execute a pre-specified action that should have immediate response and using a convolutional neural network (CNN) model to classify whether subsequent video frames contain the visual response of the action. This game delay time is further broken down into server-end processing delay, network delay, jitter buffer delay, decode delay, and render delay. Experiment shows that the server-end processing delay is the main bottleneck, accounting for up to 73.54% of the game delay time. To assess robustness to network impairments, DECAF measures the bitrate, frame rate, resolution, and freeze count of the video stream at various levels of bandwidth and packet losses. All the platforms in the experiment struggled to deliver at 1080p/60fps even when the available bandwidth is 8-20 times above the platforms' recommendation. Furthermore, a slight increase in packet losses from 0.1% to 1% resulted in a detrimental reduction in bitrate of up to 6.6 times. Lastly, each platform adopts a different video streaming strategy. Under network constraints, Nvidia GeForceNow (GFN) appeared to prioritize low freeze count while Google Stadia and Amazon Luna held the bitrate and resolution stable.

Review

The most interesting and surprising finding in the paper was that server-end processing delay takes up most of the game delay time. The team had expected network delay to be the main bottleneck in cloud gaming. A possible explanation for this is that the experiment might have been carried out remotely on cloud servers instead of deployed physically at the three locations. In this case, the network delay might have been underestimated due to the lower latency between server-to-server communication as compared to communication with home clients since the connections between server farms may have been more robust. Particularly, if the authors used AWS as clients, we would expect the network delay to Luna to be minimal, as Luna is deployed on AWS. Also, the use of keep alive messages instead of ICMP pings to measure network delay could also be a source of inaccuracy since these are usually sent at relatively longer intervals (defaults at 75 sec [6]) as compared to the game delay (in msec). As such, they will not capture any spike in network delay (if any) at the point of measurement.

Next, we will discuss several limitations (apart from those already stated in the paper) of the proposed DECAF methodology and suggest directions for future work for some of these limitations. The authors have commented that previous work such as that proposed in ITU [1] is lacking due to: i) lack of generality, ii) limited to measuring network delay and not the server processing delays, and iii) lack of interpretability due to use of singular QoE value. The core model in [1] also uses performance measures that are generally applicable to any games such as the resolution, bitrate, framerate, packet loss and delay. Therefore, we do not observe the highlighted

lack of generality. Furthermore, since the singular QoE is derived using the various measures of streaming quality and network performance, these measures can be trivially displayed alongside the QoE values for interpretability if needed. In fact, our team believes that the ideal monitoring system should offer a singular value as a quick summary of the platform performance and support ad-hoc drill-down analysis of the various measures. The lack of a singular score in DECAF has made it unclear to us which is the most performant platform. Nevertheless, given that server-end processing might be a bottleneck, we agree with the authors that the ITU's model could be enhanced by including the server-side delays.

The authors have proposed some novel measures that are not found in previous work [1] such as the game delay time and freeze count. We believe that the introduction of these new measures should be substantiated to show that they are useful or indicative of user experience. We have assumed here that we are measuring performance as a gauge for user experience since DECAF treats the platforms as black-boxes and attempts to measure performance from the perspective of a platform's user/client. Particularly, we believe that user experience surveys can be also conducted alongside the experiments. Correlation analysis between the measures and the user experience scores can then be carried out to determine which are the better measures. Generally, measures with higher degree of correlation should be more indicative of the user experience.

We noted that DECAF has limited application due to the need for the gaming platforms to support WebRTC and share common games. Since DECAF uses WebRTC on Chromium for data collection, platforms that exclusively stream via its own application client such as Sony's PS Now and Shadow [2,3,4] cannot be measured. More importantly, to get relative performance, DECAF requires the game to be available on all the platforms. This can be rather limiting as it is common for games to be exclusive to selected platforms due to their distribution rights. In addition, we expect the number of common games among platforms to decrease as we increase the number of platforms for comparison. This is evidenced in the paper, where the racing game used in experiment Crew2 is available on Stadia and GFN but only the older prequel Crew is available in Luna. On a side note, the fact that Crew and Crew 2 are developed 4 years apart also makes us question the fairness to use them for the experiment. As such, we believe that this heavily limits the application of DECAF to only a small number of titles that are available on all the platforms. A way to overcome this may be to compare relative performance of a home setup and the other cloud gaming platforms. For example, we can attempt to find the minimum hardware specification (for a consumer PC) that is required to achieve the performance for any game on any of the cloud gaming platforms. This allows comparison of each platform's performance to be independent. If we wish to compare across platforms, we can still do so by comparing their respective minimum hardware specifications. We believe this would be also meaningful to the consumers as it can help them to decide between investing on their own PC or subscribing to a cloud gaming platform.

As noted in the discussion section of the paper, the DECAF's bot does not intelligently play the game. This brings about questions on the reliability of the measurement to reflect delay under real game play since the load generated by the bot may be distant from actual gameplay by a player. Particularly, certain game scenarios might be more computationally expensive such as fighting against multiple enemies while surrounded by continuous explosions in Far Cry 5. The DECAF's bot is unable to generate this load since it is stuck with doing simple movements and shooting bullets in the air. Arguably, we will expect performance comparison under lighter loads to be somewhat indicative of that under heavier conditions (e.g. GFN to have the least game delay in Far Cry 5 under load generated by an actual player). However, we can only speculate on this as the degradation due to the additional load is likely non-linear and different platforms may have different strategies for dealing with it. While the paper has proposed to overcome this by using Deep-Q networks (DQN) to train the bot, we believe that some games such as Far Cry 5 are too

complex to be learnt using DQN and a better approach might be to train bots that exhibit similar behavior to an actual player through imitation learning methods such as GAIL [5]. With this in mind, DECAF will also need to be extended to measure the response for more than 1 activation pairs for each game.

Lastly, we do not think that DECAF is very scalable due the need to construct a training dataset by labeling 5000 frames for each game (4000 for training and 1000 for validation). As the complexity of game scenarios increases (in the future with smarter bots) and the number of measured activation pairs increases, we may also need more training data to achieve the same 99% accuracy due to the associated increase of visual complexity. For example, a game scenario with multiple explosions in the background may cause bright light that confuses the model as noted in Appendix A of the paper. We would propose exploring the use of unsupervised or self-supervised ML methods to remove the need for training data. A possible method is to use metric learning to track differences between each subsequent frame. Based on a pre-specified distance threshold, we can look for the most recent frame after an action invocation that exceeds the distance threshold from its previous frame. Intuitively, the first frame with visual response (e.g. muzzle flash from firing a gun, Fig 17a in paper) should be quite different from its directly preceding frame (e.g. frame with just the gun, Fig 17b in paper).

References

[1] G.1072 : Opinion model predicting gaming quality of experience for cloud gaming services. https://www.itu.int/rec/TREC-G.1072/en.

[2] Shadow. https://shadow.tech/.

[3] PS Now. https://www.playstation.com/en-us/ps-now/.

[4] Di Domenico, Andrea, et al. A network analysis on cloud gaming: Stadia, GeForce Now and PSNow. Network 1.3 (2021): 247-260.

[5] Ho, Jonathan, and Stefano Ermon. Generative adversarial imitation learning. Advances in neural information processing systems 29 (2016).

[6] TCP(7) Linux User's Manual. https://man7.org/linux/man-pages/man7/tcp.7.html