Compensating for Latency in Cloud-based Game Streaming using Attribute Scaling

Xiaokun Xu*, Michael Bosik*, Adam Desveaux*, Alejandra Garza*, Alex Hunt*, Cameron Person*, James Plante*, Joseph Swetz*, Nina Taurich*, Brian Clark†, Doris Hung†, Philip Lamoureux†, and Mark Claypool*
* Worcester Polytechnic Institute, Worcester, MA, USA
email: xxu11, mbosik, acdesveaux, agarza, amhunt, caperson, jplante, jsswetz, claypool @wpi.edu
† Google Inc., Mountain View, CA, USA
email: brclark, dhung, plamoureux @google.com

***Abstract*—Cloud-based game streaming has the disadvantage of added latency from the thin client to the cloud-based server and back, decreasing player performance and degrading their experience. Attribute scaling can make the game easier, potentially exactly counteracting the difficulty added by the latency. We incorporate attribute scaling models into two different games, deploy them on a commercial cloud-based game streaming system and evaluate their efficacy by measuring impact on player performance and Quality of Experience (QoE). Analysis through a user study shows that our compensation methods improve player performance and may improve QoE compared to no latency compensation.**

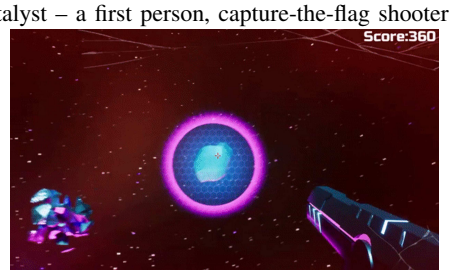
I. Introduction

Cloud-based game streaming differs from traditional online games in that game streaming clients are relatively lightweight, only sending user input (e.g., keyboard, mouse) and receiving game output as video [1]. While this architecture offers advantages over traditional game systems, it has the significant disadvantage of client-server round-trip latency for all game actions. Prior work has shown even modest amounts of latency can impact player performance and Quality of Experience (QoE) [2], [3], [4], [5], [6], [7]. Approaches to compensate for network latency have been widely used in traditional multiplayer network games [8], [9], but many cannot be applied to cloud-based game systems since the client is "thin," not having the processing power and game information needed to compute the game state and render the game world.

One promising technique for cloud-based game streaming is attribute scaling [10], [11], [12], [13], [14]. To compensate for latency, the game object attributes (e.g., precision and deadline [15]) can be scaled according to a client's latency in order to provide for a "just right" difficulty that the game designer envisioned. This can provide for a consistent challenge for the player, regardless of the client's latency. Attribute adjustments would not necessarily be appropriate for all game objects, but instead could be decided by a game developer (e.g., tagging an object attribute). The cloud-based game engine would then make automatic adjustments on the fly based on the round-trip latency experienced by each client. This can provide for a uniform game experience, the game experience envisioned by the game designer, for all players, regardless of their clients' latencies. While some cloud-based game streaming systems may seek to deliver single-player games without source-code changes, others, such as Google Stadia, provide and encourage development targeted for their platform where attribute scaling can be of use to game developers.

978-1-6654-8794-8/22/$31.00 ©2022 IEEE

(a) Catalyst – a first person, capture-the-flag shooter game.



(b) Nova – a first person, target selection rhythm game.

Fig. 1: Screen shots of custom-built games used in our study.

This paper presents a step towards attribute scaling for latency compensation in a game engine. We develop two custom games each with a different attribute scaling technique that incorporates a model to automatically scale object attributes based on a client's latency, game difficulty and the desired player performance. We deploy both games to the Google Stadia cloud-based game streaming service and use them in two user studies to evaluate the efficacy of using attribute scaling models [16], [17], [18] in actual, full-featured games. The cloud-based servers stream the games over a custom router that controls the added latency for our user study. Analysis of the results shows the attribute scaling is effective for improving player performance and can improve

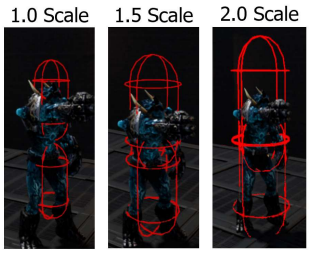


Fig. 2: Catalyst – attribute scaling by adjusting hitbox size.

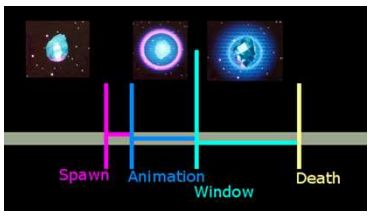


Fig. 3: Nova – attribute scaling by adjusting window duration.

player QoE.

II. Methodology

A. Games and Attribute Scaling

We designed and developed two custom games from scratch using Unreal Engine (UE4 v4.25).[1]

Catalyst is a first person, team, capture-the-flag game where players cast spells (a total of 10 are available) instead of using weapons. Figure 1a shows a screen shot. The player chooses combinations of water, fire and air elements at the bottom of the screen to determine the spell cast. The cast spell then launches towards the reticle aimed at an opponent.

The spatial attribute scaling method for Catalyst is shown in Figure 2. The hitboxes are highlighted with red ovals and vary in size with the scaling factor. The scaling factor increases with latency to keep player accuracy at the no latency condition. In the actual game, the hitboxes are invisible.

Nova is a first person rhythm game where players shoot invading asteroids at times aligned with the music. Figure 1b shows a screen shot. The player fires the gun in the bottom right corner, using a reticle to aim at the asteroid, shown glowing in the center of the screen, when the asteroid is targetable (in time to the music).

The temporal attribute scaling for Nova is depicted in Figure 3, with time progressing left to right. The target spawns and then animates as it moves towards the player until it enters the targetable time window. The player must shoot the asteroid during the window to destroy it or suffer death/damage. The time window duration is scaled (lengthened) with latency.

Both games are deployed to the Google Stadia cloud-based game streaming service. Latency is computed using the Google Stadia API by polling a client's instant (latest) latency, called through a UE4 Blueprint, smoothed with an exponentially weighted moving average (α 0.9).

[1]https://www.unrealengine.com/

For both games, the main performance metric is accuracy. For Catalyst, accuracy is shots hit divided by shots fired, and for Nova, accuracy is notes (asteroids) hit divided by total notes spawned. Our previous work took user study data and derived models that relate attribute scaling to player performance, game difficulty and scale factor [18].

From [18], for Catalyst, the hitbox scaling factor (s) is:

$$s = \frac{a + 0.09l + 0.1d - 73}{20} \quad (1)$$

where l is the latency (in milliseconds) and d is the difficulty (the speed of the opponent avatar, in cm/s).

For Nova, the time window scaling factor (s) is:

$$s = \frac{a + 0.1l - 12d + 63}{10} \quad (2)$$

where l is the latency (in milliseconds) and d is the difficulty (the cooldown between notes spawning, in seconds). Nova has several songs built-in, with cooldowns ranging between 0.75s (easy) and 0.5s (hard).

For a given game level, the game designer chooses the intended accuracy based on the challenge they envision for the player (not too easy, not too hard). The game system then adjusts the attribute scale factor (Eq. 1 for Catalyst, Eq. 2 for Nova) to keep the challenge consistent despite latency.

B. User Study

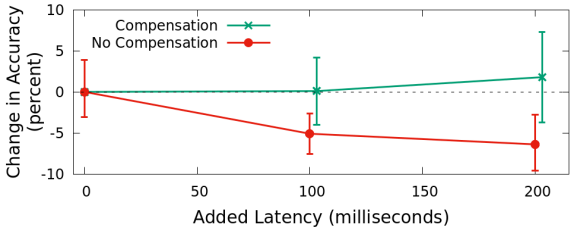
We setup a testbed where we deployed our games – Catalyst and Nova – to the Google Stadia servers and added controlled amounts of latency to/from the clients. The clients were 13" laptops, connected via wired Ethernet through our campus network to the Google Stadia servers where our games were deployed. The laptops had 1920x1080 @ 60 Hz displays and ran Microsoft Windows 10 and Stadia via Chrome. The local system latency measured with a 1000 f/s camera (a Casio EX-ZR100) was 39.2 ms (σ 6.9 ms) for the Catalyst laptop and 39.6 ms (σ 5.9 ms) for the Nova laptop.

The Google Stadia servers mean base round-trip latency from our testbed was 11 ms (σ 2.3 ms). Clumsy[2] v0.2 (64-bit), a network manipulation tool, was used to add additional latency to the client network uplink and downlink.

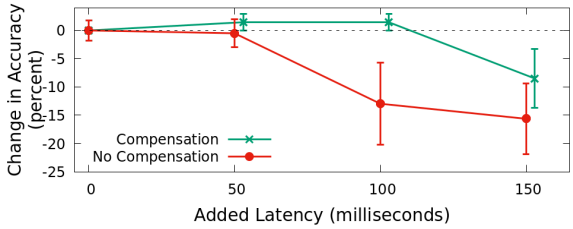
While Catalyst is intended to be a 3v3 game with human players, we developed AI players ("bots") so the user study could have only one human participant at a time on each team.

For Catalyst, users played 7 rounds total. After a tutorial (results discarded), users played 3-minute rounds with 0, 100, or 200 ms of added latency, and hitbox scaling latency compensation either on or off. Only one player had added latency while the opponent had no added latency nor latency compensation. For Nova, users played 17 forty-five-second rounds. After a tutorial (results discarded), users played with 0, 50, 100 or 150 ms of added latency, with temporal window scaling latency compensation either on or off. Conditions were randomly shuffled for both games.

[2]https://jagt.github.io/clumsy/



(a) Catalyst



(b) Nova

Fig. 4: Accuracy versus latency.

TABLE I: Participant demographics

Study	Users	Age (yrs)	Gender	Gamer
Catalyst	26	21.4 (2.4)	20 ♂ 6 ♀	3.5 (1.3)
Nova	27	20.2 (1.2)	18 ♂ 9 ♀	3.6 (1.3)

After each round, participants filled out a survey to indicate their quality of experience via a Mean Opinion Score (MOS) "Please rate your experience". Input was via a text box with 1.0 (low) to 5.0 (high) point numeric entry, shown along with a scale: Excellent, Good, Fair, Poor, Bad.

Participants were recruited via University mailing lists. Students could receive credit for game or psychology courses.

III. Analysis

A. Participants

Catalyst had 26 users participate in the study. When only one participant signed up for a slot, the same study proctor played the role of the non-lagged participant. The proctor data was discarded from our analysis. This resulted in 18 participants with latency conditions and 8 participants without. Nova had 27 participants. Table I summarizes, showing means with standard deviations in parentheses. Gamer score was a self-rating for the question "Rate your experience as a gamer", from 1-low to 5-high.

B. Performance

Catalyst has 11 possible spells, but only Fireball is analyzed here due to space constraints and since it accounted for 60% of all spells cast. We computed hit accuracy only for rounds were a player cast fireball 5 or more times, yielding a total of 95 rounds for accuracy analysis.

Figure 4a depicts the accuracy of the fireball spell versus latency. The horizontal axis is the added latency and the vertical axis is the change in accuracy from the base condition without added latency. Each point is the mean for all users for that condition, shown with a 90% confidence interval. The red data are for rounds with no latency compensation and the green data are with hitbox attribute scaling. From the data, without latency compensation, accuracy decreases with latency, degrading by almost half from 0 to 200 milliseconds of latency. In converse, with hitbox attribute scaling, accuracy remains high despite the latency.

Differences in accuracy with latency compensation were statistically significant (α 0.05) at all added latency values. Overall, there was a significant difference in accuracy for rounds using compensation (M=13, SD=17) and no compensation (M=9, SD=11), $t(44)=8.01$, $p<0.001$.

Figure 4b shows the change in accuracy for Nova with added latency, with axes and data as for Figure 4a. From the graph, without latency compensation, accuracy decreases with 100+ ms of added latency, dropping about 15% at 150 ms. With temporal window attribute scaling, accuracy is resilient through 100 ms of latency, and about 5% better at 150 ms.

Differences in accuracy with latency compensation were statistically significant (α 0.05) at all added latency values. Overall, there was a significant difference in accuracy for rounds using compensation (M=93.5, SD=9.9) and no compensation (M=89.1, SD=14.6), $t(98)=3.03$, $p=0.003$.

C. Quality of Experience

For both Catalyst and Nova, participant QoE decreased with added latency both with and without latency compensation.

For Catalyst, QoE values without added latency averaged 4.3, decreased slightly to 4.0 with latency 100 ms and then dropped sharply to about 2.8 with latency 200 ms. With latency compensation on, QoE values were about 10% higher at 100 ms but only about 5% higher at 200 ms. However, given the relatively small sample size (only 18 participants for the paired t-test), some spells without compensation, and the team aspects of the game, there was not a significant effect on the difference in QoE for compensation on (M=3.7, SD=1.2) and compensation off (M=3.5, SD=1.1), $t(53)=-0.73$, $p=0.23$.

For Nova, QoE values without added latency averaged 4.5, and decreased about 0.5 points per 50 ms of latency. With latency compensation on, QoE values were 3% higher at 50 ms and 10% higher at 150 ms. Overall, there was a significant effect on the QoE for compensation on (M=3.9, SD=1.2) and compensation off (M=3.7, SD=1.2), $t(197)=1.87$, $p=0.03$.

IV. Conclusion

Game attribute scaling can improve cloud-based game streaming by adjustments to keep the challenge as intended by the game designer. We incorporate mathematical models for attribute scaling with latency into two games and deploy them on Google Stadia's cloud-based game servers. Evaluation via two user studies shows attribute scaling latency compensation can keep player performance high despite network latency, with additional tuning work needed for QoE. Future work is also to build attribute scaling into a game engine (e.g., UE4) to scale select game object attributes automatically.

REFERENCES

[1] R. Shea, L. Jiangchuan, E. Ngai, and Y. Cui, "Cloud Gaming: Architecture and Performance," *IEEE Network*, vol. 27, no. 4, pp. 16–21, Jul-Aug 2013.

[2] K. Chen, Y. Chang, H. Hsu, D. Chen, C. Huang, and C. Hsu, "On the Quality of Service of Cloud Gaming Systems," *IEEE Transactions on Multimedia*, vol. 26, no. 2, Feb. 2014.

[3] M. Claypool and D. Finkel, "The Effects of Latency on Player Performance in Cloud-based Games," in *Proceedings of ACM Network and System Support for Games Workshop (NetGames)*, Nagoya, Japan, Dec. 2014.

[4] A. Sackl, R. Schatz, T. Hoßfeld, F. Metzger, D. Lister, and R. Irmer, "QoE Management Made Uneasy: The Case of Cloud Gaming," in *IEEE Conference on Communications Workshops (ICC)*, Kuala Lumpur, Malaysia, May 2016.

[5] V. Clincy and B. Wilgor, "Subjective Evaluation of Latency and Packet Loss in a Cloud-based Game," in *Proceedings of the International Conference on Information Technology: New Generations (ITNG)*, Las Vegas, NV, USA, April 2013.

[6] C. Huang, C. Hsu, Y. Chang, and K. Chen, "GamingAnywhere: An Open Cloud Gaming System," in *Proceedings of the ACM Multimedia Systems Conference (MMSys)*, Oslo, Norway, Feb. 2013.

[7] M. Jarschel, D. Schlosser, S. Scheuring, and T. Hoßfeld, "An Evaluation of QoE in Cloud Gaming Based on Subjective Tests," in *Proccedings of the Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, Seoul, Korea, 2011.

[8] S. Liu, X. Xu, and M. Claypool, "A Survey and Taxonomy of Latency Compensation Techniques for Network Computer Games," *ACM Computing Surveys*, vol. 1, no. 1, Feb. 2022.

[9] Y. Bernier, "Latency Compensating Methods in Client/Server In-game Protocol Design and Optimization," in *Proceedings of the Game Developers Conference (GDC)*, San Francisco, CA, USA, Feb. 2001.

[10] S. Sabet, S. Schmidt, S. Zadtootaghaj, B. Naderi, C. Griwodz, and S. Moller, "A Latency Compensation Technique Based on Game Characteristics to Mitigate the Influence of Delay on Cloud Gaming Quality Of Experience," in *Proceedings of the ACM Multimedia Systems Conference (MMSys)*, Istanbul, Turkey, May 2020.

[11] I. Lee, S. Kim, and B. Lee, "Geometrically Compensating Effect of End-to-End Latency in Moving-Target Selection Games," in *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*, May 2019.

[12] S. Saeed, S. Schmidt, S. Zadtootaghaj, C. Griwodz, and S. Moller, "Towards Applying Game Adaptation to Decrease the Impact of Delay on Quality of Experience," in *Proceedings of IEEE International Symposium on Multimedia (ISM)*, Dec. 2018.

[13] J. Xu and B. Wah, "Concealing Network Delays in Delay-sensitive Online Interactive Games based on Just-noticeable Differences," in *IEEE International Conference on Multimedia and Expo (ICME)*, San Jose, CA, USA, 2013, pp. 1–6.

[14] R. Salay and M. Claypool, "A Comparison of Automatic versus Manual World Alteration for Network Game Latency Compensation," in *Proceedings of the ACM Symposium on Computer-Human Interaction in Play (CHI Play)*, Virtual Conference, Nov. 2020, Work in Progress.

[15] M. Claypool and K. Claypool, "Latency and Player Actions in Online Games," *Communications of the ACM*, vol. 49, no. 11, Nov. 2006.

[16] S. Liu and M. Claypool, "Game Input with Delay - A Model of the Time Distribution for Selecting a Moving Target with a Mouse," in *Proceedings of the 27th International Conference on MultiMedia Modeling (MMM)*, Virtual Conference, June 22-24 2021.

[17] M. Long and C. Gutwin, "Characterizing and Modeling the Effects of Local Latency on Game Performance and Experience," in *Proceedings of the ACM Symposium on Computer-Human Interaction in Play (CHI Play)*, Melbourne, VC, Australia, 2018.

[18] E. Carlson, T. Fan, Z. Guan, X. Xu, and M. Claypool, "Towards Usable Attribute Scaling for Latency Compensation in Cloud-based Games," in *Proceedings of the Game Systems Workshop (GameSys)*, Instanbul, Turkey, Sep. 2021.