Peer-to-Peer Support for Low-Latency Massively Multiplayer Online Games in the Cloud

Richard Süselbeck, Gregor Schiele, Christian Becker
Universität Mannheim, Germany

Abstract

Cloud gaming has recently been proposed as an alternative to traditional video game distribution. With this approach, the entire game is stored, run and rendered on a remote server. Player input is forwarded to the server via the Internet and the game's output is returned as a video stream. This adds network delay, which can negatively impact the gameplay. The delay is acceptable as long as the user is located geographically close to the cloud servers. However, for Massively Multiplayer Online Games (MMOGs), this delay is added on top of the existing delay between MMOG client and server. As MMOGs are highly delay-sensitive, this can significantly degrade their playability. To deal with this issue, we propose to use peer-to-peer techniques to distribute the MMOG server functionality and place it at the cloud server centers. This allows us to reduce the additional delay introduced by running the MMOG clients in the cloud.

Figure 1: OnLive (www.onlive.com) is one of several companies preparing a Cloud Gaming service for launch. The service uses a so-called Micro-Console to connect to the Cloud Servers.

Our Approach

Currently, any MMOG that is run in the cloud is incurring a delay penalty. After a player initiates a game event, e.g., an attack on another player, it takes four hops for the associated update to reach the other player. This situation is depicted in Figure 2.

However, if all players of an MMOG were running their MMOG clients on the same cloud server center, the MMOG server could be co-located with the cloud server. In this case, the communication between the MMOG clients and the MMOG server would be completely within the local network of the server center, which results in minimal added delay. This means the delay traditionally experienced between the client and the MMOG server would be almost eliminated, with only the input and streaming delay remaining. Unfortunately, the geographic locations of an MMOG's players are widely distributed, sometimes across the entire world.

To enable the co-location of Cloud and MMOG servers, while allowing multiple geographically distributed server centers, we propose to use techniques developed for peer-to-peer-based MMOGs.

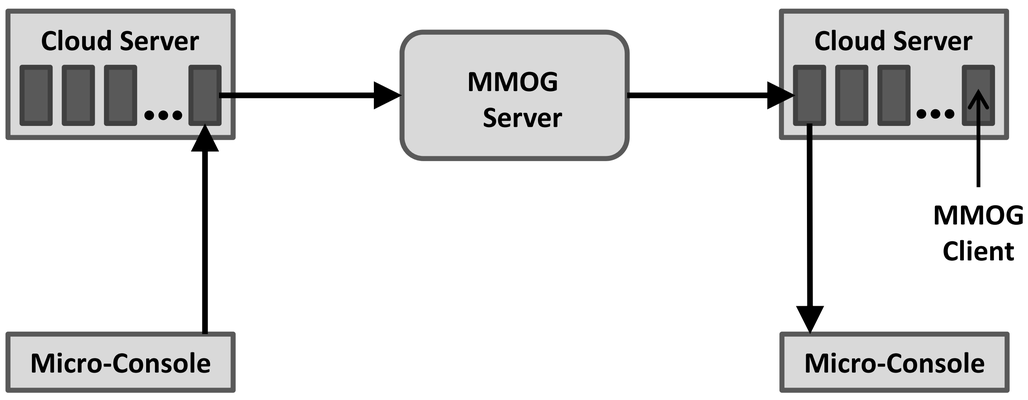

Figure 2: Client/Server-based MMOG in the Cloud. Note that an update takes four hops to reach the other player.

These techniques allow us to distribute the MMOG server's functionality among several servers. The servers can then be co-located with the cloud servers. Specifically, we suggest to base the MMOG on the peer-to-peer middleware that we are developing in the peers@play project (www.peers-at-play.org).

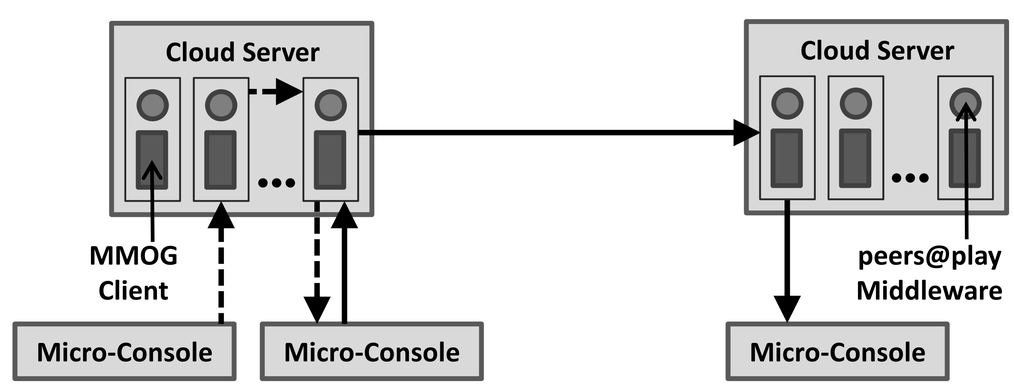

Figure 3: Peer-to-Peer-based MMOG in the Cloud. Note that an update takes between two (dashed line) and three hops (solid line) to reach the other player.

In our approach, the combined cloud/MMOG server runs both the game client and an instance of our peer-to-peer middleware for each player that connects to it. As far as the middleware is concerned, it is still running a fully distributed peer-to-peer system, unaware that its peers are running on only a few separate server centers, instead of the systems of the individual users. Any given update now takes either two or three hops to reach, depending on Cloud Server to the recipient of the update is connected to. This is depicted in Figure 3.

Our existing prototype of the middleware supports this approach without any modifications, as we have in fact run experiments with a similar setup, where we ran a large number of peers on our IBM BladeCenter (see Süselbeck et al.: Adaptive Update Propagation for Low-Latency Massively Multi-user Virtual Environments. Proc. IEEE IC3N 2009 for details).

Conclusion

In conclusion, our approach allows an MMOG developed as a peer-to-peer-based system to be run in the cloud. This reduces the delay penalty that would normally be associated with cloud-based MMOGs. Our approach can reduce the number of hops an update has to traverse from four to between two and three. While our existing prototype supports this concept without modifications, we are currently implementing several optimizations in order to evaluate our concept.

The peers@play Project: http://www.peers-at-play.org