Scaling Peer-to-Peer Multiplayer Games with Doppelgängers

Jeffrey Pang* Frank Uyeda† John Douceur‡ Jacob R. Lorch‡

jeffpang@cs.cmu.edu fuyeda@ucsd.edu {johndo,lorch}@microsoft.com

*Carnegie Mellon University † U.C. San Diego ‡Microsoft Research

1 Introduction

Multiplayer online games have become an important part of the computing landscape. There is a growing desire to use the machines of participants themselves rather than separate dedicated servers to provide resources for serving games. This approach reduces subscription costs, eliminates dependency on centralized infrastructure, and allows automatic scaling to an arbitrary number of clients. To achieve these benefits, several peer-to-peer architectures have been proposed [1, 2]. In these architectures, each peer is responsible for a subset of all game objects. Each peer sends updates about its objects to other peers that are interested in them.

An obvious limitation of these architectures is that they only work when each peer has sufficient upload capacity to send updates to all other interested peers. However, residential broadband upload rates are typically quite limited, e.g., only 100s of kb/s in the United States, and are insufficient to support the update rates required by games with dozens or hundreds of interacting players. Area-of-interest filtering can help, but cannot eliminate the problem; e.g., if updates require 16 kbps and a player with 128 kbps is in sight of more than 8 other players, his peer simply does not have enough bandwidth to send updates as frequently as the game requires. For this reason, we are developing a new architecture, called Donnybrook, to enable large-scale peer-to-peer games even in environments with such highly constrained bandwidth.

Donnybrook uses a novel type of replica called a *doppelgänger* to improve the playability of peer-to-peer games in low-bandwidth conditions. To demonstrate the feasibility of our approach, we modified Quake III, a popular first-person shooter (FPS) game, to run on the Donnybrook architecture. In addition, we conducted a large user study comparing Donnybrook with the current state of the art. Our results show that using doppelgängers substantially improves the playability of peer-to-peer Quake III in low-bandwidth environments. Moreover, players are nearly as satisfied by Quake III with doppelgängers in a low-bandwidth environment as they are by Quake III in a high-bandwidth environment (e.g., a LAN). Based on this study, we estimate that Donnybrook enables games with an order of magnitude more players over existing architectures given the same amount of bandwidth.

2 Doppelgängers

Doppelgängers are based on three principles:

Players have bounded attention. A human has a fixed "attention budget" and, hence, can only focus on a bounded number of objects at the same time. For example, even in a large firefight, a player will tend to focus on his or her current target. This principle has two implications. First, an object's update rate to a player should vary based on his or her attention on the object. Second, the aggregate amount of attention in the game grows only linearly with the number of players, not quadratically. Thus, there is always sufficient capacity (within a constant factor) to maintain update rates proportional to player attention. We use a *focus set* to implement this differentiation in update rates.

Interaction must be timely and consistent. The focus of online games is player interaction. Therefore, it is critical that interactions occur in a timely and consistent fashion. For example, when a player shoots and kills another player, both players should observe the death immediately. Even small delays would be jarring because of the expected timing of the sequence of events, which can be spaced only milliseconds apart in FPS games. Moreover, if an interaction is not observed consistently, the inconsistency may be revealed via out-of-band channels (e.g., chat). Because interactions are rare relative to other updates, this principle suggests prioritizing interactions (i.e., inter-object writes) over all other updates. We use *pairwise rapid agreement* to implement this prioritization in a consistent fashion.

Realism should not be sacrificed for accuracy. If objects are not in a player's focus, then it is more important that they appear "realistic" than accurate. For example, it is more important to ensure that the change of a doppelgänger's position obeys game physics than to minimize the error with respect to the primary. An out-of-focus object that violates game realism is more likely to be noticed than one that is portrayed with small inconsistencies. Therefore, doppelgängers are controlled by a *guidable AI*. This guidable AI causes the doppelgänger to act in a realistic manner between position updates, unlike traditional replicas which only perform dead-reckoning of position between updates. With guidable AI, the doppelgänger also converges to the true state of an object when a player focuses attention on it, so it becomes more accurately represented.

3 Demonstration

Those in attendance at OSDI '06 will be able to play a demo of Donnybrook during the poster session.

References

[1] BHARAMBE, A. ET AL. Colyseus: A distributed architecture for online multiplayer games. In *NSDI* (May 2006).

[2] KNUTSSON, B. ET AL. Peer-to-peer support for massively multiplayer games. In *INFOCOM* (July 2004).