DiveReal: a Social Virtual World without Rooms

Joaquín Keller
Orange Labs Research
44 avenue de la République
92326 Châtillon, France
joaquin.keller@orange.com

Mathilde Laurent
École Nationale Supérieure d'Informatique
pour l'Industrie et l'Entreprise
1 rue de la Résistance – 91000 Évry, France
mathilde.laurent@ensiie.fr

Abstract—**DiveReal is a social virtual world: people get in primarily to meet and talk to others. But unlike its classical counterparts, it does not rely on servers to put people together. Instead, it uses an overlay that grows as more people get in. For the user, this means a different experience: the world is not anymore divided in "rooms", "islands" or "realms" but accepts everyone, possibly millions, in a unique seamless space.**

I. Introduction

In his 1992 science fiction classic novel *Snow Crash*, author Neal Stephenson imagined a collective virtual reality called the *Metaverse*, where user-controlled avatars hung out in 3D venues. It is a virtual world intended to meet people, do business, socialize.

Since then, despite several tentatives to realize Neal Stephenson's dream, social virtual worlds still attract orders of magnitude less users than other popular social online services.

In the first section of this article we identify the server-based architecture –and the induced space division in rooms– as a major issue for social virtual worlds. We then describe DiveReal, a virtual world implemented to overcome the problem on an overlay. This novel architecture allows a unlimited number of users to come together and interact through their avatars in a continuous virtual space.

II. Issues with rooms

In virtual worlds, the common way to have people together interacting is connecting them to the same server. As servers have bounded ressources, this limits the size of groups to few hundreds, and usually much less[1].

This is perfectly acceptable for games with rules compatible with this constraint. However, having a limit in the number of people together ruins the social experience, first by restraining with whom a user can interact but even worse, by leaving many users mostly alone for extended periods of time.

This is not intuitive, and this section is aimed at explaining the consequences, from the user's perspective, of the server-based architecture. Let's start first with some vocabulary:

Scene: The 'scene' is what is seen by users. Here, for the sake of simplicity and because it suffice for basic socialization, we will consider only scenes with avatars evolving in a static landscape.

Decor: The static elements of a scene.

Room: The portion of virtual space corresponding to a given server. Anyone connected to the server has her avatar in the room. Actually, a 'room' does need to look like a real room and may have any size or shape.

Region: A virtual space can be divided in rooms, each with a different decor. These rooms are called 'regions' of the overall space.

Shard: When many rooms have the same decor, they are called 'shards' of the same place. Shards and regions can be combined and multi-region shards are not uncommon.

Regarding social virtual worlds –which primary purpose is to be with others– rooms have unwanted and awkward effects. First, when a user arrives in an empty room, since her goal is to socialize, she do not stay long. Hence, empty rooms tend to remain empty. One common solution to this issue is to organize a social event to be held in the room at a given time so many will arrive simultaneously. But then arise another problem, if the tentative of attracting people is successful, soon the room reach its maximum capacity and start refusing people.

In region-based worlds, like Second Life®, the consequence is that most regions are empty or nearly empty while few regions have actually some social activity but are too crowded and not accepting anyone [2]. Then, even if many users are connected at a given moment, the standard experience for a new-coming user is being alone or with at most one or two others [3].

The other way to get people *apparently* in the same place is by having enough shards to host them all. However users who are assigned to a given shard can interact with users in the same shard but not with users in other shards. This restriction can be specially annoying [5] for a user willing to be with someone in particular.

Moreover the fluctuation of users arriving and leaving the system eventually put some shards bellow the threshold needed to sustain a social activity thus accelerating further their depletion.

We can note that in the happy case of an steady stream of users joining the system, when all shards are at maximum capacity a new –empty– shard has to be made available. But then the users arrive in an empty room so they may change their mind and decide not to stay.

Dividing a social virtual world in many rooms –either by sharding or by partition in regions– not only isolated users, lessening the social value of the service, but repel new users often putting them in empty rooms. This is hardly a design choice but more an unwanted consequence of the server-based architecture. It is then admirable that some social virtual worlds have succeed at all, reaching, however orders of magnitude less users than other popular social services.

III. Using Kiwano Overlay

To address these issues with rooms we have designed and implemented Kiwano, an overlay aimed at replacing the servers running today's classical virtual worlds. Unlike a server, the Kiwano overlay can grow at will –by adding more machines– and accommodate an unlimited number of users in a continuous space.

As the internals of Kiwano are detailed in a previous paper [6], this section is devoted on describing how Kiwano is used to enable an actual virtual world: DiveReal.

The first step to connect to Kiwano overlay is to ask an *allocator* for an *entry point*. The allocator is called only once but the connection to the entry point remains for the whole session.

Each time the user moves her avatar, the coordinates of the new position are sent to the entry point. Whenever an avatar moves, leaves or arrives in the neighborhood a notification is sent by Kiwano so at each moment, the user has an updated list of the neighbors and positions.

The decor is handled outside of Kiwano by a spatial database that provides the static elements of the world at avatar's position. For DiveReal first version we use Google Earth™ and Street View™ data and servers so our avatars evolve in a mirror world, a digital copy of our planet.

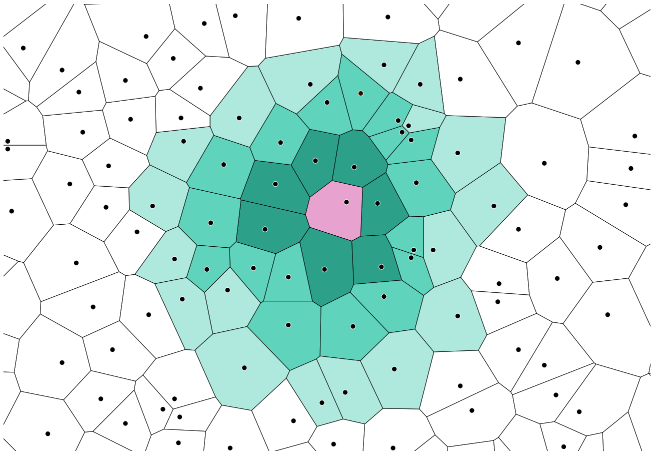

Fig. 1. The first three levels of an avatar's neighborhood

The neighbor relation between avatars is based on the Delaunay triangulation of their positions and is defined as follow: two avatars are said to be k-neighbors when they are at distance k in the Delaunay graph (see figure 1). And they are said to be neighbors whenever they are k-neighbors for some $k \leq n$ with $n = 3$ in this first version (this a parameter easy to change in Kiwano). The article on Kiwano gives an estimate of the number of neighbors at each level: On average an avatar has 6 1-*neighbors*, 21 2-*neighbors*, 43 3-*neighbors*. For $n = 3$, the probability of having more than 150 neighbors in not significant and as a practical matter, the number of neighbors should be considered bounded.

Hence, regardless the total number of users connected to the overlay, each one is only aware of a bounded number of avatars. For instance in a crowd only the few dozens neighboring avatars are visible and available for interaction.

Even though, this could be too demanding for the hardware and –besides the classical crowd rendering techniques [8] which alleviate the display– the levels of neighborhood are used to determine the amount and frequency of data exchanges between avatars. And thanks to the symmetric nature of the Delaunay graph and thus of the k-neighbor relations, no one receives unwanted data from irrelevant non-neighboring avatars.

The Kiwano overlay allows to send messages to neighbors. This ability serves as the signaling channel for the ICE protocol[9] and direct connections –peer-to-peer or via TURN servers[10]– are established between neighbors. These direct connections are used by DiveReal to enable audio communication among participants and to transmit avatar states and animations.

IV. Distributed simulation and consistency

Usually, virtual world servers do more than just ensure the communication among the connected. In the standard implementation the server is a simulator that computes the state of the world and broadcast it to all, as in figure 2. When a user wants her avatar to perform some action such as ”move forward” or ”shoot monster”, the request is forwarded to the server which computes the resulting effects on the world. For example, order ”move forward” would yield an effect such as ”avatar moved 20 cm before hitting a wall” while ensuing order ”shoot monster” could end up in ”too late, the monster just ate your avatar”.

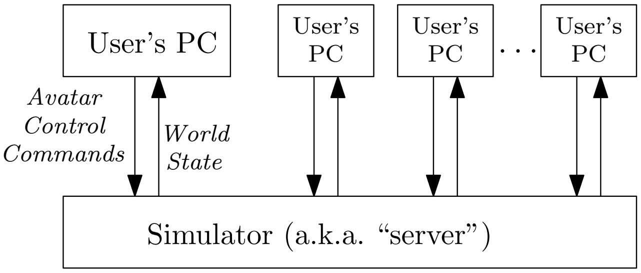

Fig. 2. Centralized simulation

Having a server in the middle receiving all data and computing simulations is clearly a bottleneck and can incur in annoying delays [3], but as the state of the world has a unique source, the consistency is thoroughly and naturally ensured, with all participants eventually receiving the same data about the scene.

At the contrary, when using Kiwano, there is no central point where every state of the world is computed. Instead, avatars' actions are broadcasted to the neighbors and a simulation of the immediate surroundings of the user's avatar is performed on each user's computer (see figure 3).

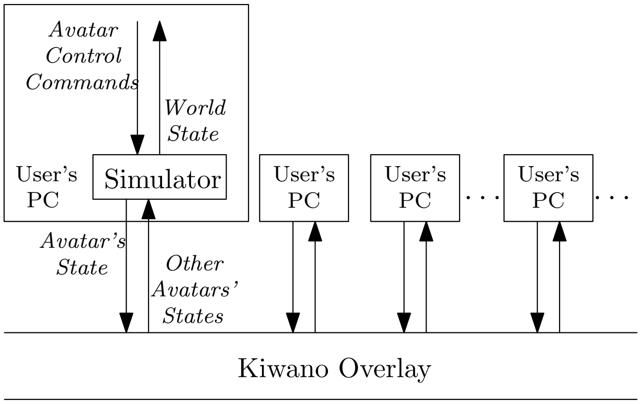

Fig. 3. Distributed simulation

Hence the simulation is distributed, which implies that at some point two users might have divergent views of the world. However what matters is not an absolute consistency but a consistency good enough so the gaming or social activities are not disrupted. DiveReal strategy to deal with discrepancies in replicated states is to allow some divergence in the short term with a guarantee the replicas will eventually converge in a delay compatible with the application. This strategy, called optimistic replication[4], is greatly simplified when the set of allowed operations is limited in order to avoid conflicts among replicas.

As DiveReal is intended for social interactions, we have, at first, restricted the set of allowed actions to a bare minimum: a user can only move her avatar and she is the only one allowed to do so. When an avatar moves not all simulators would know it simultaneously, but —as shown in virtual world behavioral studies [7]— most avatar movements are followed by long periods of immobility, giving thus time to replicas to synchronize theirs states.

When relying on a central server, consistency might arguably be better, but at the cost of an increased delay between the user action and its visible effects. This latency, known as *lag*, is induced by the round trip to the distant server and becomes even worse when the server approaches its maximum capacity and gets late in computing and transmitting the next state of the world.

V. Conclusion and Future Work

The DiveReal architecture around Kiwano overlay enables a virtual world to host thousands, and even millions, into a seamless space. To reach such scalability, we have had to rethink how virtual worlds are implemented and separate functions that are often seen mandatorily together. In the DiveReal architecture the decor comes for a spatial database, the communication is peer-to-peer –when possible, the simulation is distributed and connections are established by proximity through a Delaunay graph computed by a distributed algorithm.

However, major issues remain to be solved, first the simulation of the world is overly simplistic; and, more complex to tackle, the social rules for a virtual world with crowds have yet to be invented.

Servers acting as referees for a zone might allow playing FPS[1] games and, more generally, innovative architectures for distributed simulation as [11] will allow complex physics.

However, designing human computer interactions for crowds in a social virtual world is a messy process involving trial and error with humans. Our first experiments with users show, for example, that teleportation tend to disrupt conversations or that assembling people with enough things in common to enact meaningful interactions has no easy recipe. To progress on this front, further experiments should involve growing numbers of users, both real and simulated.

We can speculate that, eventually, a social virtual world will start attracting users, while having a increasing value as more people join it. Hopefully no ”room” limit will stop its growth.

References

[1] N. Gupta, A. Demers, J. Gehrke, P. Unterbrunner and W. White, *Scalability for Virtual Worlds*, International Conference on Data Engineering (ICDE), 2009

[2] W. J. Au, *The Crowded Empty Paradox: How A Virtual World With 75,000 Online Users Can Still Seem Abandoned*, http://nwn.blogs.com/nwn/2010/01/the-crowded-empty.html, Personal Blog, 2010

[3] M. Varvello, S. Ferrari, E. Biersack and C. Diot, *Exploring Second Life*, IEEE/ACM Transactions on Networking, 2011

[4] Y. Saito and M. Shapiro, *Optimistic Replication*, ACM Computing Surveys, March 2005

[5] J. Waldo, *Scaling in Games and Virtual Worlds*, Communications of the ACM, August 2008

[6] R. Diaconu and J. Keller, *Kiwano: Scaling Virtual Worlds*, Winter Simulation Conference (WSC), Washington D.C., USA, 2016

[7] D. Pittman and C. GauthierDickey, *A measurement study of virtual populations in massively multiplayer online games*, Workshop on Network and Systems Support for Games (NetGames), Melbourne, Australia, 2007

[8] A. Beacco, N. Pelechano and C. Andújar, *A Survey of Real-Time Crowd Rendering* Computer Graphics Forum, 2016

[9] J. Rosenberg, *Request for Comments: 5245, Interactive Connectivity Establishment (ICE)*, Internet Engineering Task Force (IETF), April 2010

[10] R. Mahy, P. Matthews and J. Rosenberg, *Request for Comments: 5766, Traversal Using Relays around NAT (TURN)*, Internet Engineering Task Force (IETF), April 2010

[11] D. Lake, M. Bowman and H. Liu, *Distributed scene graph to enable thousands of interacting users in a virtual environment*, Workshop on Network and Systems Support for Games (NetGames), Taipei, Taiwan, 2010

[1] First-Person Shooter, a popular game category