OneSim: Scaling Second Life with Kiwano

Joaquín Keller, Raluca Diaconu

OneSim: Scaling Second Life with Kiwano

[Extended Abstract]

Joaquín Keller
Orange Labs
Issy-les-Moulineaux, France
joaquin.keller@orange.com

Raluca Diaconu
LIP6, Université P&M Curie
Paris, France
raluca.diaconu@lip6.fr

ABSTRACT

Kiwano is a distributed infrastructure to scale virtual worlds developed within our team. OpenSim is an open source, enhanced implementation of Second Life servers. Combining these two systems we have designed OneSim, a distributed system to allow an unlimited number of users to be together in one sim: all players run a OpenSim instance of the same region populate with their respective neighboring avatars.

1. INTRODUCTION

Second Life (SL) virtual world, with a million unique visitors every month [8], is still popular ten years after its birth. However, despite a vibrant research community and an exponential growth in hardware performance, the per-region scalability has remained low with a maximum of few hundred users [9] at best in a single region.

Since the early 1970s, when the first multi-user graphic virtual world appeared, the algorithmic complexity of running Massively Multiuser Virtual Environments (MMVE) is $\mathcal{O}(N^2)$, where N is the number of users together in the same region. Reducing the "visibility" to a smaller surrounding area [7, 6] does not solve the problem: user density may vary sharply by some orders of magnitude, thus making the size of the area inadequate.

With Kiwano[4], we have taken a radical approach: users can see and interact only with their "neighbors". Kiwano's *neighboring* relation is designed such that the number of neighbors remains roughly constant regardless of the avatar density distribution. Thus, the complexity for computing and transmitting interactions has been dramatically reduced to $\mathcal{O}(N)$. In Kiwano the overall complexity to maintain the neighborhood relation has a mean complexity of $\mathcal{O}(N)$. Or, in different words: the per-user computing cost is constant and does not grow with N. This is the first step to scalability.

Taking inspiration in Manycraft[5], an implementation to scale Minecraft, we have designed OneSim, a distributed architecture to run OpenSim[2] on top of Kiwano. In this architecture, where each user hosts a node, there is no limit in the number of users.

The first two sections describe briefly Kiwano and OpenSim. In section III the OneSim architecture is described and discussed.

2. KIWANO

Kiwano is a scalable distributed infrastructure for virtual worlds, designed to support an unlimited number of moving objects updating their position at arbitrary high frequencies. In Kiwano the set of avatars is distributed onto many servers, each taking care of a group of avatars based on their geographical proximity. The positions are indexed using a distributed Delaunay triangulation for an efficient neighborhood access.

To free the application layer from the distributed internals, Kiwano provides an API [1] to developers. When connecting, a client is assigned a proxy, the entry point to Kiwano for the entire session. Each proxy is connected to a subset of all users and, for each of them, maintains the communication with the corresponding zone and neighborhood. The proxy also maintains and regularly updates the states of the connected clients. And, in order to scale, Kiwano spawns as many zone servers and proxies as needed.

Kiwano API receives two types of messages from clients: *update* entity state and *message to neighbors*. On the other side, it sends four types of notifications: the full list of *neighbors*, neighborhood *updates*, *message* from another client, and *remove* neighboring entities. All Kiwano messages are encoded in json, common now in most programming languages. Messages have an *appdata* field to carry data specific to the virtual world to scale. This allowed Kiwano to be successfully employed for Manycraft[5], a distributed architecture to scale Minecraft, and HybridEarth[3], a social mixed reality world covering the whole planet.

3. OPENSIM

In OpenSim, to enter the virtual world in a particular region, users connects their *viewer* to the simulator (a server) hosting this region. Simulators contain one or more regions and can be interconnected to form larger, continuous spaces known as grids. All simulators (sims) in a grid share a set of resources: login service, users accounts storage, assets storage, inventory storage, avatar storage.

The viewer sends the avatar intended actions and movements, the sim applies physical rules and computes the interactions with the elements of the decor and the other avatars

before replying with the actual action or movement –as seen by everyone.

The cost of the simulation implies a limit in the number of avatars that can be hosted by a sim. The maximum number of users reported in a single sim is around one hundred.

4. ONESIM

In our architecture each viewer is associated to a OneSim node. All the nodes are connected and communicate through Kiwano. A OneSim node consists of an instance of a regular sim, holding a local copy of the region, and a proxy to forward the avatars messages to Kiwano.

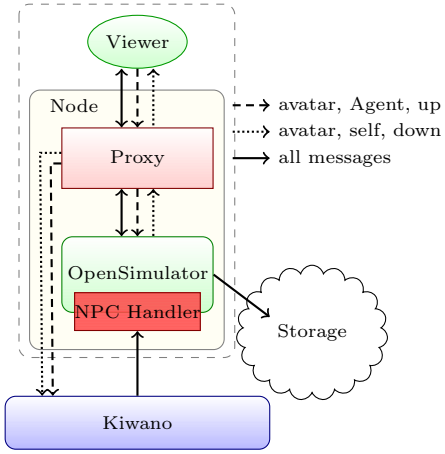

Figure 1: OneSim node

In the local sim, the neighbors, provided by Kiwano, evolve as Non-Player Characters (NPCs).

The packets between the viewer and the embedded sim are forwarded unmodified by the proxy. When the viewer sends a packet concerning the avatar (e.g. AvatarUpdate) the message in response from the sim (e.g. ImprovedTerseObjectUpdate) is intercepted to generate Kiwano updates and messages-to-neighbors. The message send by the sim contains the result of the simulation and the actual position and state of the avatar while the message from the viewer might carry extra data still needed for the final representation.

The messages sent to Kiwano generates notifications for all neighbors and are used to control the corresponding NPC avatar. In addition to position updates, these messages also bear avatar animation, orientation, appearance, chat and other avatar actions.

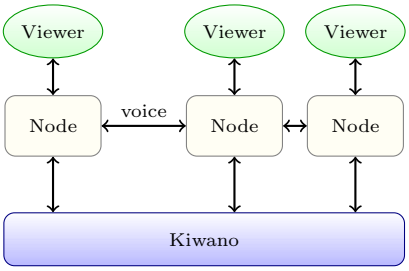

Figure 2: OneSim Architecture

As each node hosts a simulation of the world this may lead to inconsistencies. This why in OneSim every non-static object has a master sim. For instance, the avatar and the inventory of a user have their reference state and position computed by the associated sim. Remote sims may computed different states but the master sim view prevail and is propagated to others. So inconsistencies do no last. As sometimes the state computed by a local simulator differs from what the master simulator dictates, the laws of physics may be temporarily not respected. Using OneSim in games with rapid pace actions might produce poor user experience.

In dense crowds, the user experience might also differ from what we have been accustomed: we can only see the neighboring avatars and not the whole crowd in the visibility area.

Today's implementation of Kiwano provides, in average, 70 neighbors. However, the number of neighbors that can be handled by a node is limited by the capacity of the embedded sim. But as NPCs consume less resources than full avatars we can expect the total number of neighbors to be in the tens, for an end-user machine.

5. CONCLUSION

The infrastructure provided by Kiwano can be employed to scale diverse virtual worlds and a similar architecture, adapted to handle the Minecraft protocol messages, have been successfully implemented in Manycraft.

In its intent to contribute to a long lasting research problem, the scalability of virtual worlds, Kiwano provides a novel approach that complement the widely used region-based and shard-based solutions.

As all the elements in place to implement OneSim are in place we hope next year to have a first version running.

6. REFERENCES

[1] Kiwano api. `http://kiwano.li/`.

[2] Opensimulator. `http://opensimulator.org`.

[3] J. de Campredon, R. Diaconu, J. Keller, and E. Triponez. Hybridearth: Social mixed reality at planet scale. *CCNC'2014*.

[4] R. Diaconu and J. Keller. Kiwano: A scalable distributed infrastructure for virtual worlds. *HPCS'2013*.

[5] R. Diaconu, J. Keller, and M. Valero. Manycraft: Scaling minecraft to millions. *NetGames'2013*.

[6] Shun-Yun Hu, Jui-Fa Chen, and Tsu-Han Chen. VON: a Scalable Peer-to-Peer Network for Virtual Environments. *IEEE Network Journal*, July 2006.

[7] J. Keller and G. Simon. Toward a peer-to-peer shared virtual reality. *ICDCS'2002*.

[8] Chris Stokel-Walker. Second life's strange second life. `http://www.theverge.com/`, September 2013.

[9] A. Valadares, T. Debeauvais, and C.V. Lopes. Evolution of scalability with synchronized state in virtual environments. *MMVE workshop, HAVE'2012*.