A Networked Virtual Environment over KAD

Matteo Varvello
Eurecom-Thomson
France
matteo.varvello@eurecom.fr

Ernst Biersack
Eurecom
Sophia-Antipolis, France
ernst.biersack@eurecom.fr

Christophe Diot
Thomson
Paris, France
christophe.diot@thomson.net

1. INTRODUCTION

A Networked Virtual Environment (NVE) is a digital world where multiple participants interact via virtual characters called *avatar*. A popular application for NVEs is Second Life (SL)[5].

SL is a NVE built by its citizens. The world is composed of *dynamic* objects, volatile entities responding to user behaviors, and *static* objects, persistent entities that are user-created building blocks of the world.

A virtual world has to fulfill different requirements. It has to be *consistent*, i.e. all players have to agree in the same state of the world. It has to be *persistent*, i.e exist even if there are no current users of the system, it has to be *interactive*, i.e. fulfill latency requirements in the management of the dynamic entities. Finally, it has to be *scalable* to an unlimited number of users.

Current NVEs rely on a a client-server (C/S) architecture. Users are content producers and continuously send state updates to the central server. The server collects all updates, computes the state of the world and sends it back to the clients for rendering.

A C/S architecture easily achieves persistency and consistency, since it exploits the server as a rendez-vous point. However, a two-way communication is required, forcing the use of latency-hiding techniques to achieve interactivity. Moreover, to sustain a continuously growing demand, multiple servers are deployed.

Recently, p2p architectures have been proposed to solve the scalability issue of NVEs system [1]. Ideally, a p2p system is a self-scalable solution, unlimited in the number of concurrent users. Moreover, a "one-to-one" communication implies an increase in the interactivity if compared to a C/S scheme.

*CoNEXT'*07, December 10-13, 2007, New York, NY, U.S.A.

A p2p network is characterized by churn, heterogeneity and unpredictable user behaviors, i.e no stable resources. In a p2p NVE the state of the world is maintained by all players in a distributed way. This makes consistency and persistency of the world an hard task.

Distributed hash tables (DHTs) can be an interesting approach for the management of static objects in a p2p NVE. The main idea behind this work is to map a virtual space into a distributed hash space and to study how an existing DHT behaves to maintain a NVE.

We propose a dynamic partitioning scheme of the virtual world (see Figure 1) into cells or geographical areas. Each cell is assigned a unique identifier in the DHT and users adopt a publish/subscribe mechanism per cell.

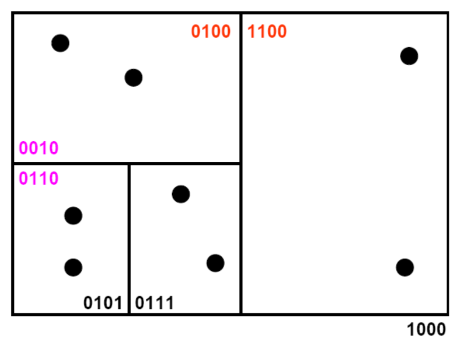

Figure 1: Cell-based management scheme

We integrated our scheme into an existing DHT and we ran trace-driven simulations to emulate the evolution of a virtual world in an existing p2p network. The results obtained show the effectiveness of our solution in constructing a consistent, persistent and scalable virtual world in an existing DHT.

2. A DHT-BASED NVE

A distributed hash table (DHT) is a decentralized distributed system that provides a lookup service similar to a hash table. Users in a DHT are uniquely identified by an ID, which is randomly generated via a cryptographic hash function. Contents are stored by selecting multiple responsible nodes in the DHT.

In order to map a virtual space into a distributed hash space, we introduce a cell-based management scheme. Each cell is assigned a cell-ID in the hash space and a

XOR minimum distance rule is used to identify in the DHT the responsible nodes of a specific cell. Objects in the virtual world are mapped to the cell they are contained in and then published on R nodes in the DHT.

A given user in a NVE is interested only in a small portion of the entire world, which we call Area of Interest (AoI). In order to maintain a consistent view of the world, participants of the NVE adopt a publish/subscribe mechanism with the group of cells that intersect their avatars' AoI.

A static division of the NVE in cells is totally inefficient, since the popularity in a NVE has been shown to follow a power law [1]. This is why we adopt a dynamic partitioning scheme. When it is created, the virtual world is composed by a single cell c_0. Then, if a user i notices that the number of objects in c_0 becomes larger then a threshold t_1, a split operation is performed. So, cell c_0 is divided into two equal size cells c_1 and c_2. In this way, we dynamically adapt the cell management scheme to the distribution of static objects in the world.

The cell-IDs are k bits identifiers organized in a k levels binary tree, so that for level $0 \leq l < (k-1)$ each cell-ID can originate two new cell-IDs. Participants of the NVE agree in a initial cell-ID as root of the binary-tree; then, each time a split operation is due, users derive the two new cell-IDs as a function of the originating cell-ID and its level l in the binary tree.

In order to maintain a persistent world over time, a distributed republication scheme is adopted. Each user dedicates a portion of its resources to republish objects he encounters in the NVE. Timestamps are used to give priority to the republishing of the oldest objects.

3. EVALUATION

KAD is a Kademlia-based [4] peer-to-peer DHT routing protocol that has been deployed on a large scale [2]. We integrated in our system the kad primitives for routing, publishing and searching, in order to exploit an existing p2p network as a testbed for our scheme.

In order to emulate the evolution of a "real" virtual world, we deployed a "crawler" via libsecondlife [3] that allows us to monitor a generic land in SL. We used the traces collected on SL to emulate avatars' behavior, but they do not give any insight about static objects distribution within the land. For this reason, we added an initialization phase where avatars create objects with a random publication rate.

The set-up of the experiment is the following: two machines located at the University of Mannheim (Germany) and at the Eurecom Institute (France) run each a maximum of 50 concurrent instances of our application. SL users' traces are collected at the Money Tree Island for 10 hrs. During this time, about 3000 different avatars were recorded with about 100 maximum concurrent avatars. The initialization phase has a duration of $T_i = 10$ min where about $N_o = 150$ objects are created. The maximum number of objects per cell is $t_1 = 20$. Objects are stored in the DHT with a replication factor $R = 20$. Each avatar has an Area of Interest radius set to 20 $units$, i.e. about 2% of the land.

We compute the NVE "consistency" as the ratio of objects seen by an avatar within his AoI and the actual objects located in his AoI. Static object position is recorded at the time these objects are created.

Figure 2 shows the evolution over time of the average consistency of the world computed among all active avatars. During the first 10 min, the consistency is affected by concurrent modifications of the world. Then, it rapidly grows and stabilizes between 95 and 100 percent. Glitches in the curve are related to churn in SL and avatars moving very fast among cells.

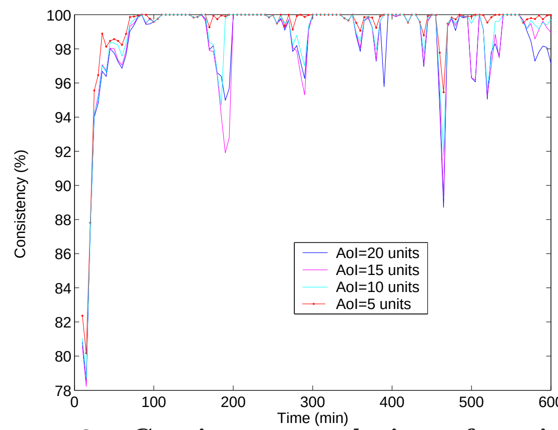

Figure 2: Consistency evolution of a virtual world over KAD - Money Tree Island (SL)

4. CONCLUSIONS AND FUTURE WORK

This paper presented a p2p scheme for the deployment of a persistent, consistent and scalable NVE. A dynamic cell partitioning scheme is proposed to map a virtual space into a distributed hash space.

We integrated our scheme in the eMule Kademlia DHT (KAD) and we deployed a crawler for SL to collect traces on users' behavior. Then, we emulated the evolution of a SL land maintained with our scheme over KAD. The results obtained show that our solution is effective in constructing a consistent, persistent and scalable virtual world in an existing p2p network.

Future work foresees to emulate multiple SL lands and to propose a comparison between the performance of a SL server and our p2p architecture.

5. REFERENCES

[1] A. Bharambe, J. Pang, and S. Seshan. Colyseus: A Distributed Architecture for Online Multiplayer Games. In *NSDI '06*, 2006.

[2] eMule. `http://www.emule.com/`.

[3] libsecondlife. `http://www.libsecondlife.org/`.

[4] P. Maymounkov and D. Mazieres. Kademlia: A Peer-to-peer Information System Based on the XOR Metric. In *IPTPS*, 2002.

[5] Second Life. `http://www.secondlife.com/`.