Twinverse Technology

A scalable architecture for virtual worlds

Index

1 A Scalable Architecture

1.1 Introduction

Historically, scalability[1] has always been a major issue when implementing virtual worlds. In other words, the problem is not actually, to build virtual worlds but to get them big . However, virtual worlds can be big in many senses.

More precisely, the values that are difficult to scale are the following:

1. Number of online users
2. Surface of the world

By organizing the way a virtual world is represented within the computer system and how data flow between these computers, scalability can be improved.

Since the first virtual world, architectures have evolved and they now scale better. However current solutions have their drawbacks and remain costly. But recently, Twinverse had come with a major technological breakthrough that changes everything.

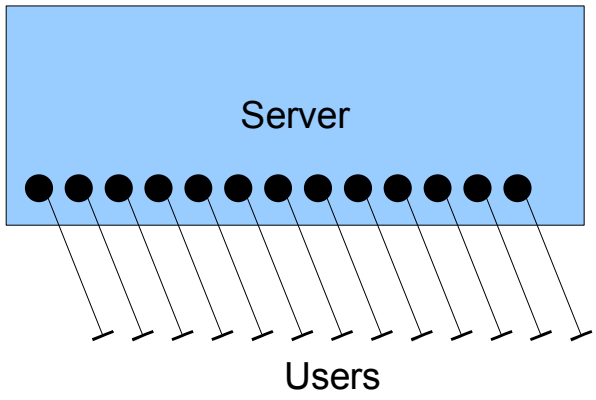
1 Scalability: saying that a system is scalable regarding a given value (number of users, peta-bytes of data, etc...) means that when the value grows the cost of the system will grow at a lower or at the same pace than the value.

1.2 Previous Architectures

1.2.1 The naive "one server" solution

The first idea that comes to mind when building a virtual world is to connect every user to a server and let the server take in charge all the virtual world logic.

The server provides -to the user's terminal- the static elements of the world, the decorum at the avatar location. Also, the server keeps track of all avatars' movements. It computes neighborhoods for each avatar.



The one server architecture: no more than ~300 users can connect simultaneously.

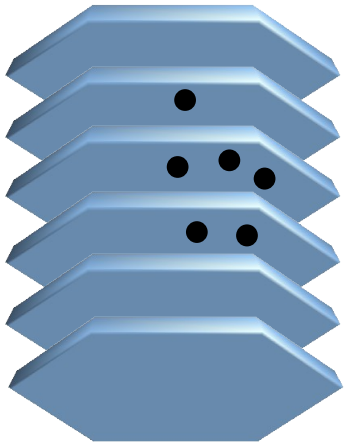
Resource consumption is roughly proportional to the number of users so there is a limit on the number of users that can connect to the server. This limit is low and after a few hundred connexions the server is overloaded and users can see their avatars becoming less and less reactive. The server may even crash if more users try to connect.

The one server solution is still in use for many multiplayer games and is the basis of the more scalable solutions used by popular systems like second life or world of warcraft.

1.2.2 Shards

The shards architecture consists on a collection of identical servers. As each server is able to handle a fews hundreds of simultaneous connections, this architecture scales well: servers can be added as new users arrive.

However, this architecture does not really implement a unique virtual world but it implements indeed a set of nearly identical small virtual worlds, the so called shards .



Shards architecture: a collection of small separated virtual worlds.

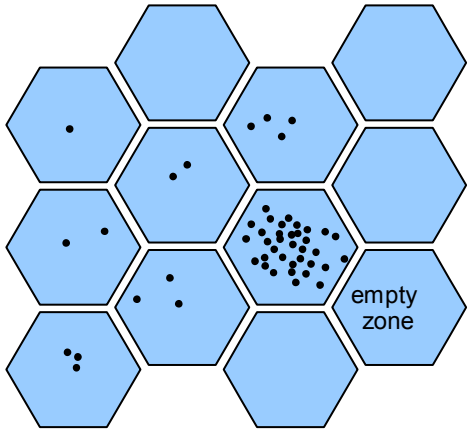
This means that when you are in a given shard you can only interact play or chat with the people also present (connected) in this same shard.

An essential feature of virtual worlds, the ability of socialization, cannot be fulfilled by this architecture. For instance, if a user wants to meet with someone, he cannot say: let's meet at 10pm on Main Plaza , he has to say: let's meet at 10pm on Main Plaza of shard 231 and if at 10pm the shard 231 is full the encounter will not even be possible.

This architecture is the one used by the popular game World of Warcraft in which the shards are called realms .

1.2.3 Zones

Before Twinverse, the solution to build a *unique* virtual world is to partition the world in parcels or zones, a different server taking the responsibility of each zone. This ensures the scalability in size of the virtual world that can grow by adding servers. Also, the scalability in number of users can be ensured by growing the world. The virtual world Second Life is built using this architecture.



Zones architecture: when a zone is too crowded it crash. A zone consume resources even empty or nearly empty.

However, this architecture for virtual worlds have several pitfalls.

1. Each zone can only host a limited number of avatars, the limit is the per-server limit, at most a few hundreds (in Second Life: less than 200). This means that if a zone becomes popular it would be soon too crowded.
2. A zone needs a server even if it is empty or nearly empty. Hence, the number of servers does not depends on how many people uses the virtual world but on how big is the world.

So a virtual world built with a zones architecture is indeed a patchwork of non scalable virtual worlds. That means that -anyhow- the requisite is at least one server per ~300 hundreds users. Indeed, in Second Life, since almost all zones are empty, one server hosts -in average- five users.

Despite the per-user cost is x50 comparing with shards architecture, zones architecture breaks the ~300 hundreds users in a unique world limit and this is a important step.

But as each acre of virtual land needs a server to back it even if there is nobody in the zone this architecture is not suitable to implement large virtual worlds. To show this impossibility, MIT's Technology Review calculates the number servers needed to build a virtual world as vast as the earth (minus oceans) using Second Life technology: 2.3 billions, even before any user arrived.

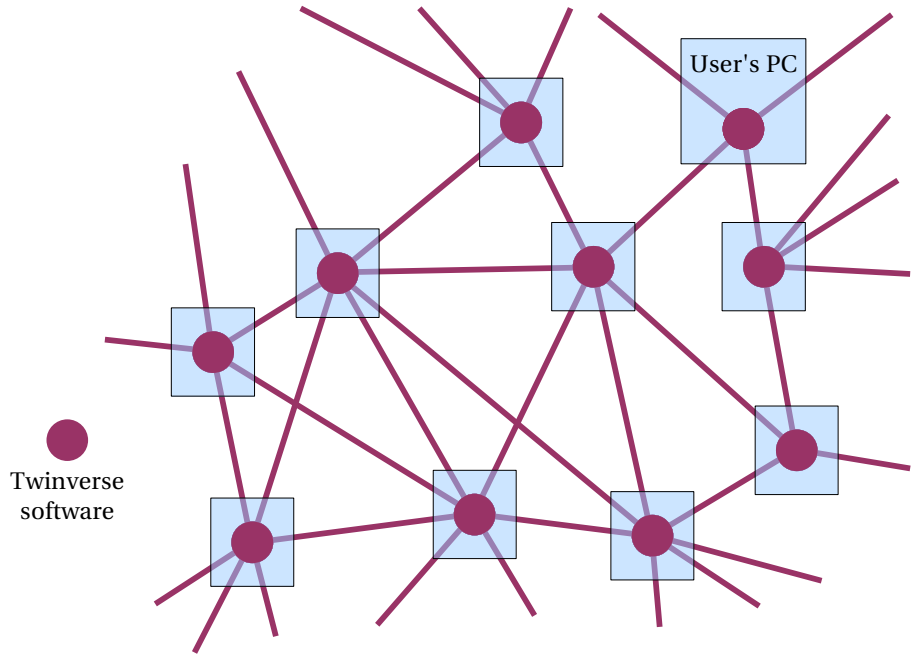
1.3 Twinverse Architecture

1.3.1 A peer-to-peer technology

Twinverse needs no servers to run its virtual world. The hardware that runs the virtual world is the mesh of users' computers, tightly interconnected and running Twinverse software. To say it short: the users do not merely sign in or connect to Twinverse system, they run it.

From the user's point of view this architecture is not very different than a server based one. He has to download a specific software as online gamers are accustomed to and for the rest he can ignore the peer-to-peerness of the system.

But on the company point of view, peer-to-peer makes the difference. Sure, servers are still needed to authenticate, store user's data or stream ads, but the at least one server per 300 users online cost is avoid. With Twinverse technology, the cost of running a



Peer-to-peer architecture: Twinverse virtual world runs on users' PCs

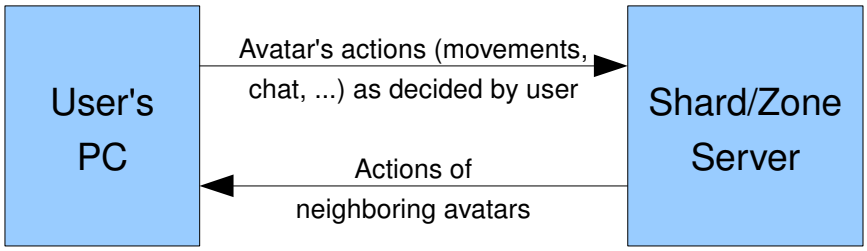
virtual world is similar as the cost of running a social network service like Facebook or MySpace. This cost structure impacts on business models that do not need to rely on per-month subscription fees to pay the servers.

1.3.2 Basic principles

Implementing a virtual world (or shared virtual reality) essentially means find a way to maintain the following properties:

1. The user has to know what happens around his avatar
2. An avatar has to be visible (by others) at its virtual position

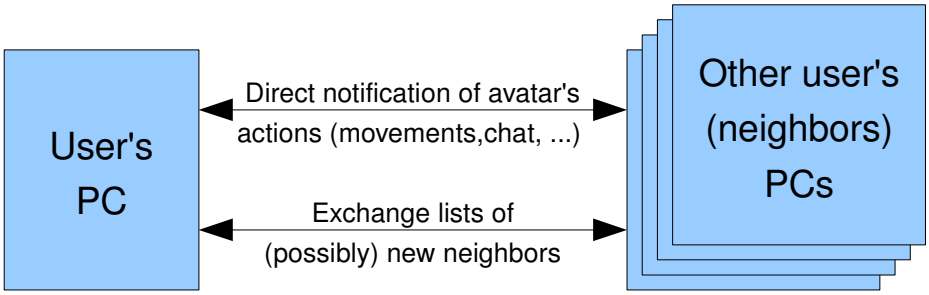
What is far away does not count for the perception of reality. The connection with the neighboring avatars is mandatory, to be aware of the theirs actions. In a server based system, the server computes who is neighbor of who and put them in touch.



Server-based shared virtual reality:

The server computes the neighborhood of every avatar

In Twinverse serverless architecture, the nodes collaborate to maintain an up-to-date list of their respective neighbors. When a node gets aware of a new avatar in the neighborhood, it forwards the information to the concerned nodes it knows. Doing so, the nodes update their neighborhood whenever a change occurs.



Twinverse peer-to-peer shared virtual reality:

The neighborhood of the user's avatar is computed by its own computer

The collaboration rules between the nodes, the protocol used to maintain the knowledge of nodes' vicinity and the connectivity with their neighbors, are the core of Twinverse technology.