RV - AULA 08 - PSI3502/2018

Networked VR

Outline

Networked VR systems are used to power distributed VE. Such environments can be deployed over various types of computer networks including Local Area Networks (LAN) and Wide Area Networks (WAN) such as Internet.

We will discuss:

1. The architecture of the network, and
2. The method of synchronization of the state of VE.

In addition, we will talk a bit about web-based VR solutions.

Networked VE

Also known as "Cyberspace".

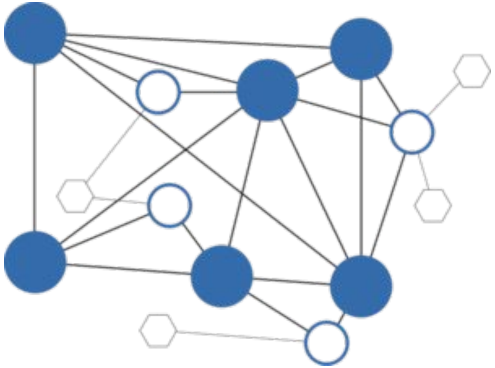
Networked VE are a form of distributed computer systems.

Applications of Networked VE

1. Military simulations
2. Telepresence / Teleconferencing
3. Distance learning
4. Online communities
5. Entertainment, etc.

Distributed Environment Issues

1. How to maintain the consistency of VE
2. The speed of data transfer and latency of the network
3. Compatibility of systems running VE
4. Heterogeneity of system running VE
5. Failure management methods, etc.



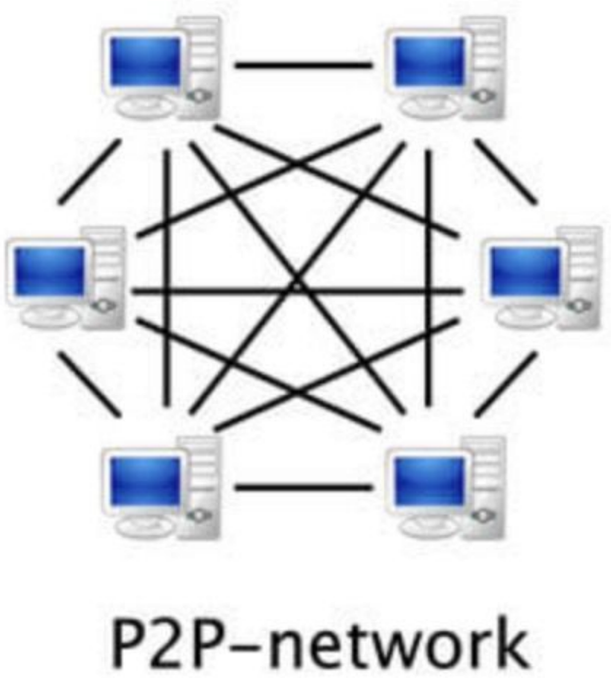
Architectures

Serverless systems, such as P2P.

Centralized server-based architectures: small single server systems, multi-server systems, very large coordinated multi-server platforms.

Serverless Architectures

The serverless architectures have no dedicated server to facilitate the exchange of data. Instead, the network consists of equal peers, which need to communicate very directly.



https://youtubemusicsucks.com/peer-to-peer-p2p-networks-explained/

Peer-to-Peer Pros and Cons

Pros:

No single point of failure of the system.

No central communication bottleneck.

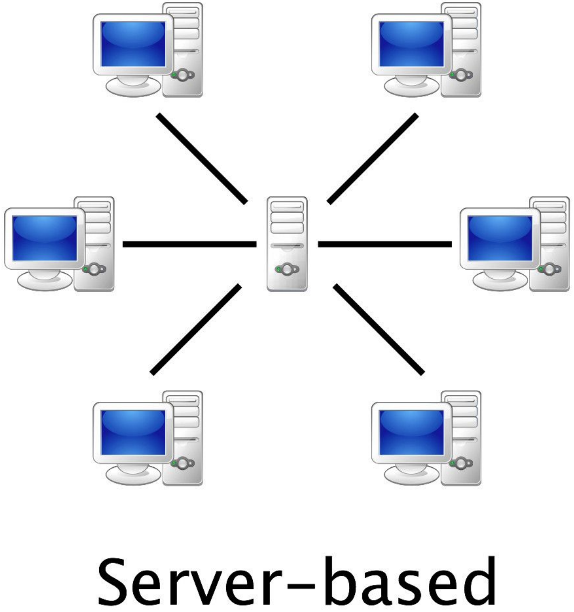
Cons:

Management difficulty.

Issues with bandwidth size.

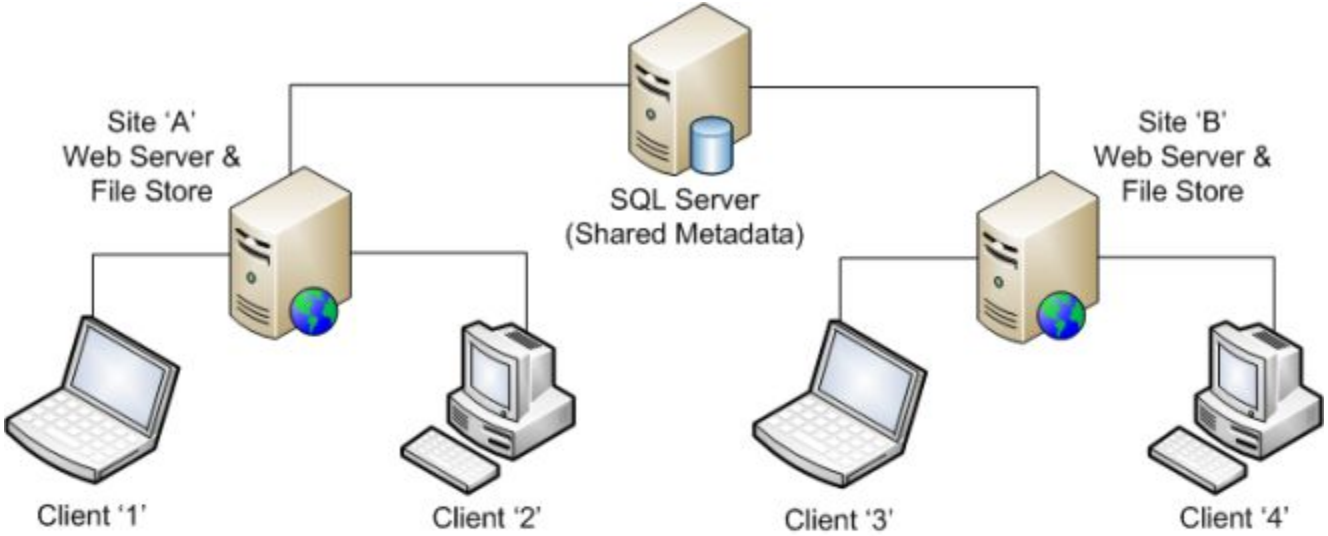
Single Server Architecture

This architecure makes the network less complex and reduces the amount of the needed updated messages. However, the server becomes the critical point in the architecure.



https://security.stackexchange.com/question/166188/p2p-network-how-to-determine-the-main-server

Multi-Server Architecture



http://help.autodesk.com/view/VAULT/Help/ENU/?guid=GUID-329BDED8-B76E-42A5-AAF1-2ADA4BC0C1BF

Coordinated Multi-Server Architecture

Coordinated multi-server architecture is an extension of standard multi-server solutions in order to accomodate a very large number of servers.

This network architecture relies on a hierarchy of servers.

This approach is used for very large distributed systems.

Example: Google Search Engine.

Coordinated Multi-Server Architecture Pros and Cons

Pros:

Dynamic sharing of the workload between clusters of servers and individual servers within one cluster and a large redundancy.

Cons:

Very complex network structure, as communication between servers within clusters needs to be treated separately from the communication between clusters and between servers and client machines.

State Synchronization

Different synchronization strategies are chosen depending on the type of the distributed system. These strategies can be grouped in three big categories:

1. Shared repository
2. Frequent state regeneration
3. Dead reckoning.

Shared Repository

In this method, individual clients send the information about the changes that they have made in the VE to the server and receive from the server the updates about the changes made by others.

Frequent State Regeneration

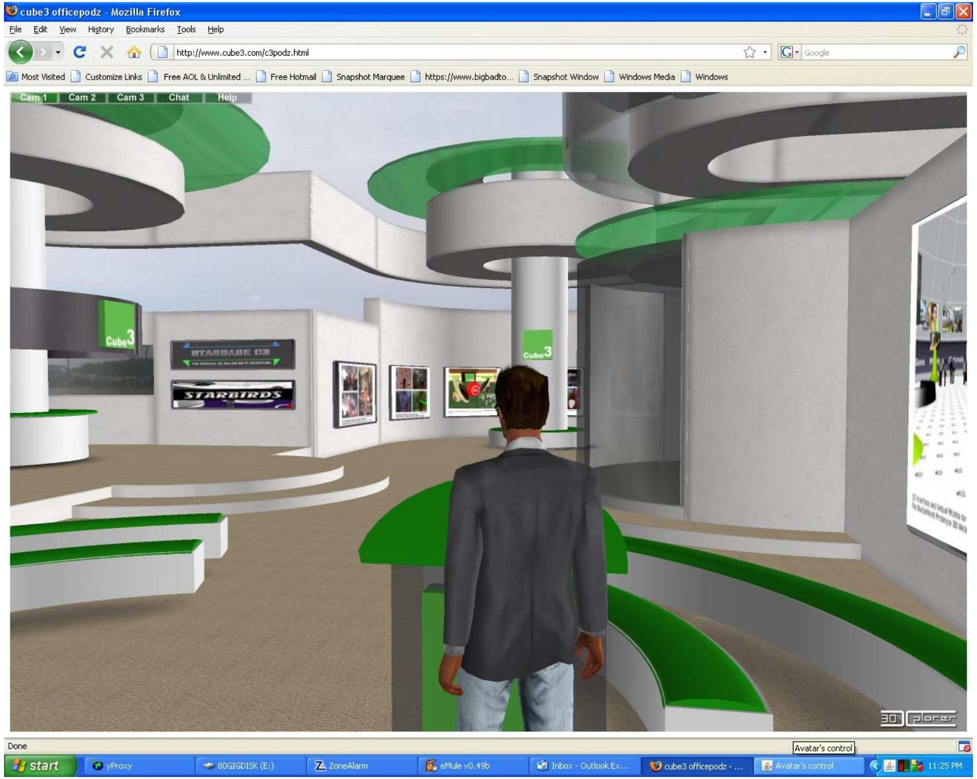
Clients or peers need to broadcast the update information to every other node in the network. In order to maintain the true state synchronization, these updates needs to be frequent and in regular time intervals.

Dead Reckoning

Dead reckoning refers to a class of algorithms that are typically employed to predict the state of objects and mitigate the effects of network latencies for clients in a networked game to improve player experience.

Each client or peer in the network maintains its own copy of VE. However, state updates between nodes are relatively rare. The fusion of data predicted by an individual client with the actual data is known as **convergence.**

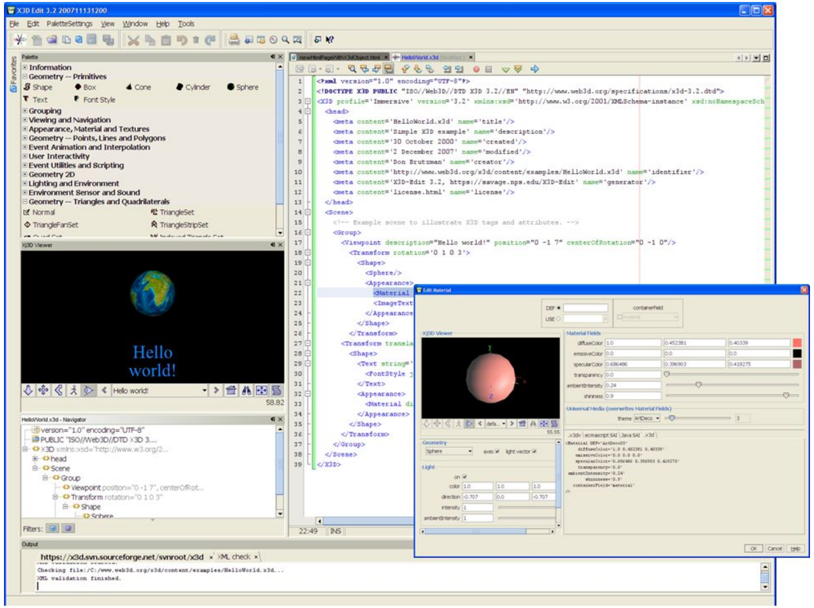
WEB 3D


cube3 officepodz - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://www.cube3.com/c3podz.html
Google
Most Visited Customize Links Free AOL & Unlimited ... Free Hotmail Snapshot Marquee https://www.bigbadto... Snapshot Window Windows Media Windows
Cam 1 Cam 2 Cam 3 Chat Help
Cube3
STARBASE C3
STARBIRDS
Cube3
3DXplorer
Done
Avatar's control
start yProxy 80GIGDISK (E:) ZoneAlarm eMule v0.49b Inbox - Outlook Ex... cube3 officepodz - ... Avatar's control 11:25 PM

https://3dxplorer.wordpress.com/2008/09/10/meet-in-3d-where-web3d-meets-web-30/

X3D

X3D is a member of the XML family of markup languages. It was designed specifically to be easily integrated into HTML documents and interpreted by HTML-enabled browsers.



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.2//EN" "http://www.web3d.org/specifications/x3d-3.2.dtd">
<X3D profile='Immersive' version='3.2' xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance' xsd:noNamespaceSch
  <head>
    <meta content='HelloWorld.x3d' name='title'/>
    <meta content='Simple X3D example' name='description'/>
    <meta content='30 October 2000' name='created'/>
    <meta content='2 December 2007' name='modified'/>
    <meta content='Don Brutzman' name='creator'/>
    <meta content='http://www.web3d.org/x3d/content/examples/HelloWorld.x3d' name='identifier'/>
    <meta content='X3D-Edit 3.2, https://savage.nps.edu/X3D-Edit' name='generator'/>
    <meta content='license.html' name='license'/>
  </head>
  <Scene>
    <!-- Example scene to illustrate X3D tags and attributes. -->
    <Group>
      <Viewpoint description="Hello world!" position="0 -1 7" centerOfRotation="0 -1 0"/>
        <Transform rotation='0 1 0 3'>
          <Shape>
            <Sphere/>
            <Appearance>
              <Material
              <ImageText
            </Appearance
          </Shape>
        </Transform>
      <Transform transla
        <Shape>
          <Text string='
            <FontStyle j
          </Text>
          <Appearance>
            <Material di
          </Appearance>
        </Shape>
      </Transform>
    </Group>
  </Scene>
</X3D>
```

https://x3d.svn.sourceforge.net/svnroot/x3d

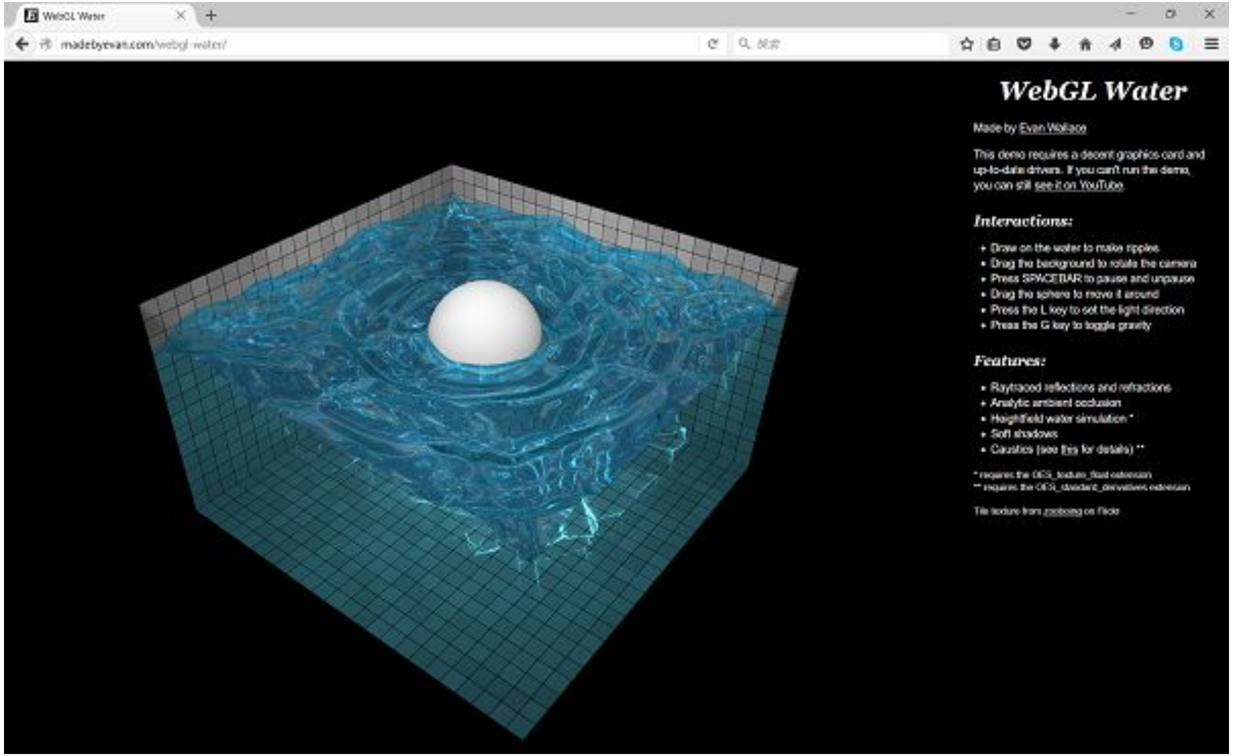
```
Checking file:/C:/www.web3d.org/x3d/content/examples/HelloWorld.x3d...
XML validation finished.
```

http://plugins.netbeans.org/plugin/6191/x3d-edit

WebGL

WebGL is a JavaScript API that supports rendering of interactive 3D graphics.

http://labotech.dmm.com/entry/2016/06/30/122403



WebGL Water

Made by Evan Wallace

This demo requires a decent graphics card and up-to-date drivers. If you can't run the demo, you can still see it on YouTube.

Interactions:

- Draw on the water to make ripples
- Drag the background to rotate the camera
- Press SPACEBAR to pause and unpause
- Drag the sphere to move it around
- Press the L key to set the light direction
- Press the G key to toggle gravity

Features:

- Raytraced reflections and refractions
- Analytic ambient occlusion
- Heightfield water simulation *
- Soft shadows
- Caustics (see this for details) **

* requires the OES_texture_float extension
** requires the OES_standard_derivatives extension

Tile texture from zooboing on Flickr

Referências

Virtual Reality and Virtual Environments in 10 Lectures - Stanislav Stankovic.

A. L. Ames, D. R. Nadeau, J. L. Moreland, *VRML 2.0 Sourcebook*, Wiley, Second edition, December 2006.

E. Lengyel, *Mathematics for 3D Game Programming And Computer Graphics*, Second Edition. Charles River Media, 2004.

T. Parisi, *WebGL: Up and Running*, O'Reilly Media, Inc., August 2012.

S. Singhal, M. Zyda, *Networked Virtual Environments: Design and Implementation*, Addison-Wesley Professional, July 1999.