

Design of the Server Cluster for the Scalable Networked Virtual Environment

Jiung-yao Huang*, Yi-chang Du+
Department of Computer Science and
Information Engineering.
Tamkang University, Tamsui 251.
jhuang@mail.tku.edu.tw*
chax@tkvr.tku.edu.tw+

Chien-Min Wang
Institute of Information Science.
Academia Sinica, Nankang 115.
cmwang@iis.sinica.edu.tw

Abstract

This paper presents an architecture to design a server cluster for the scalable networked virtual environment. One of the key techniques to design such a scalable networked virtual environment is to partition the virtual world across multiple servers. However, the problem of the avatar migration from one server into another is the key issue of this approach. This paper proposes a mechanism to solve this problem. Furthermore, the mathematical model of data distribution management to control the message flow between the server and clients is also presented. With this mathematical model, the efficiency of the message flow control mechanism can be easily verified. Significantly, this model can also be used to estimate the performance of the designed server cluster. Finally, the message flow control algorithm for the proposed mechanism is presented at the end.

Keywords: Networked Virtual Environment, Spatial Culling, Area-Of-Interest, Server Cluster

1. Introduction

The research of the Networked Virtual Environment(NVE) is to study how to fuse the virtual reality technique with the networking environment to support real-time interaction among spatially distributed users in a computer generated synthetic environment. A networked virtual environment that supports hundreds of simultaneous participants can offer wider application domains and become more attractive to the users. However, as the number of the users inside the networked virtual environment increases, the amount of the message flow among these users can overwhelm the network bandwidth as well as the computational power on the client site. That is, as the number of participants increases, a host has to spend extra

CPU time to filter out the messages that are irrelevant to him. To solve this problem, the message flow inside the network must be carefully controlled so that dispensable messages can be filtered out before it is sent out onto the network. Singhal and Zyda[15] described this problem as the issue of the resource management for scalability and performance. They further divided the resource management issue into four areas, which are optimizing the communication protocol, controlling the visibility of data, exploiting human perceptual limitations, and changing the net-VE network software architecture.

The goal of this paper is to study the scalability issue of the networked virtual environment. Based upon the message communication mechanism, the networked virtual environment can be classified into three basic architectures, which are client/server, peer-to-peer unicast, and peer-to-peer multicast.[4] For the client/server architecture, the client sends his own status information to the server and receives the status information of other users from the server. However, the legacy client/server architecture uses a single server as a communication hub[11] that does not scale up for the number of the simultaneous participants. Different server cluster techniques are then actively explored to resolve the scalability problem. These techniques include partitioning clients across multiple servers, partitioning the net-VE across multiple servers, and server hierarchy.[15] This paper proposes a client/server architecture that combines the second and third techniques. Furthermore, the proposed system follows the specification and infrastructure of High Level Architecture(HLA),[9] which is the IEEE standard of the network virtual environment.

2. Previous Works

Various networked virtual environments were built in the past to explore the essential

techniques of scalability. For example, RING[6] proposed a client-server architecture to support real-time visual interaction among distributed users in a shared 3D virtual environment. DIVE[8] system has fully duplicated homogeneous and distributed databases to increase the number of participants. Both the BRICKNET[14] and MASSIVE[7] systems employ shared, distributed and client-server databases to achieve the same goal. In addition, BRICKNET took one step further to allow objects' dynamic behaviors to be shared among distributed players, and MASSIVE uses a spatial model for database partition among clients. NPSNET[12] uses the DIS(Distributive Interactive Simulation) standard[2] along with the proposed Area-of-Interest technique to communicate the entity status information among distributed players. Significantly, the MASSIVE and the Virtual Society(VS) by Sony Computer Science Laboratory Inc.[10] are two known networked virtual reality systems to be designed on the Web architecture. SharedWeb[11] is another type of web-based networked virtual reality system that is tightly integrated with the existent web server.

With the maturity of the Internet technology, US military are extensively using the network to interconnect existing simulators to perform the military drill in the recent years. In 1995, Department of Defense(DoD),US, proposed a new standard called High Level Architecture(HLA) to further promote the interoperability and reusability of different types of simulators.[5] HLA has become the international standard, IEEE 1516, in the year of 2000. According to the HLA terminology, the simulating system is called the federate and the simulation environment that is composed of multiple federates is called a federation. The HLA is defined by three concepts: [9]

- **Object Model Templates(OMT)** - provides a template for documenting HLA-relevant information about classes of simulation or federation objects and their attributes and interactions.
- **Interface Specification** - provides a specification for the DoD HLA functional interfaces between federates and the underline distributed operating system. The Interface Specification defines all of the necessary services to federate in a way that is analogous to how a distributed operating system provides services to its application.
- **Compliance rules** - delineates the set of rules that are applied to HLA federations and federates. These rules describe the principles of constructing an HLA

environment.

The Interface Specification defines the programming interfaces of six types of services, which are Federation Management, Declaration Management, Object Management, Ownership Management, Time Management, and Data Distribution Management as the interoperation interface among simulations. The software that realizes these six service groups is called Run Time Infrastructure(RTI). Since the original design goal of HLA is to interoperates more than 100,000 simultaneous players, the Data Distribution Management is the essential service that aims to achieve this scalability.[17]

3. Data Distribution Management

One of the essential concepts to achieve the scalability of the NVE is to partition clients among multiple servers and each server manages moderate number of participants only. As specified by Singhal and Zyda[15], this partition can be based upon the geographical location of the participants, or the coordination of the avatars within the virtual world. Furthermore, users within a server can be further classified so that the messages flow among them are filtered based upon their respective interests. Moreover, the deployment of multiple servers will also significantly affect the scalability of the simultaneous participants. Hence, the goal of this project is to study the multiple server hierarchy that partitions the clients into distinct groups according to their respective avatars' coordinates. Each server is only responsible for managing the data flow of a single region.

The method of filtering the messages among users can be performed on either logically separation or physically division their corresponded avatars' coordinates. The logically separation is to compute the message interest scope of every avatar on-the-fly and exchange the messages among users only when whose interest scopes are intersected. For example, MASSIVE used the Aura[7] to compute the awareness of an avatar. Different quantity of messages will be transmitted among clients depending upon the awareness of each avatar. The physical division is to partition a virtual world into cells beforehand. Area-Of-Interest(AOI)[13] of an avatar is then dynamically computed based upon the cell that it resides in. AOI represents the scope in space that a client is interesting to receive messages. Therefore, how to determine the Area-Of-Interest of a client to filter the messages and, hence, reduce the network bandwidth and computational load is the key to build a scalable

networked virtual environment.

Locale[3] by the Spline system proposed a different approach to filter the messages and, hence, control the data flow. Each locale has its own coordinate system and the entire virtual world is constituted by numbers of Locales. Only the clients within the same Locale will then receive each other's messages. PARADISE[16] proposed a "projection aggregation" approach that consists of organization aggregation and grid aggregation. This approach provides a high-level method to establish AOI based on the general spatial area and entity class.

Interleaved Squaring method[1] is another approach of the spatial culling method for the data flow management. The Interleave squaring method partitions the virtual world into interleaved squares. This approach aims to adopt the advantage of uniformly updating in AOI change provided by the hexagonal approach[13], while without its complexity in computing the resident cell. Hence, in this project, the Interleaved Squaring method is adopted to compute the Area-Of-Interest of an avatar.

4. The Scalable Server Cluster

4.1 The Scalable Architecture

In order to support multiple distinct scenes and multiple simultaneous participants at the same time, a hierarchy of server cluster is designed as shown in Figure 1. The proposed server cluster contains three types of servers, which are the gateway server, the master server and the client server. The gateway server is responsible for authorizing the user who wants to enter a virtual world. Furthermore, the gateway server also contains the information of the master server which manages that specific virtual world. The master server is the main server that manages a virtual world. It will record all the information of the virtual world, including the size of the virtual world, the number of the regions that were partitioned and the information of the slave server that manages a region. The major function of the master server is to coordinate the process of transferring an avatar from one slave server into another. Finally, the slave server is the one that actually controls the data distribution among the clients that are resides in the region that it manages.

The relationship between the master server and slave server can be illustrated by Figure 2. As shown in Figure 2, each slave server is responsible for the clients with their corresponding avatars located within a particular region of the virtual world. The data exchange

among the slave servers can be performed with the peer-to-peer method, while the master server is responsible for coordinating the establishment of the communication channel among the slave servers.

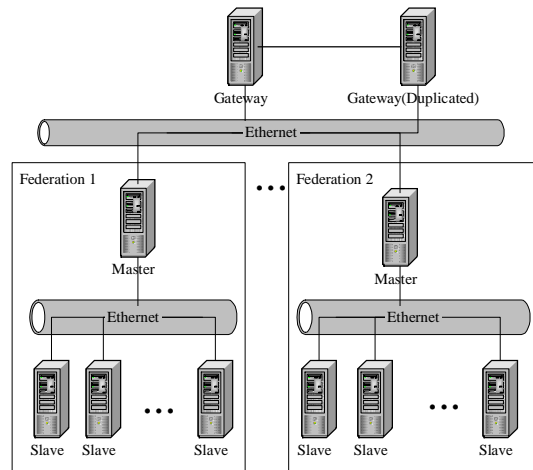


Figure 1. The hierarchy of multiple server architecture

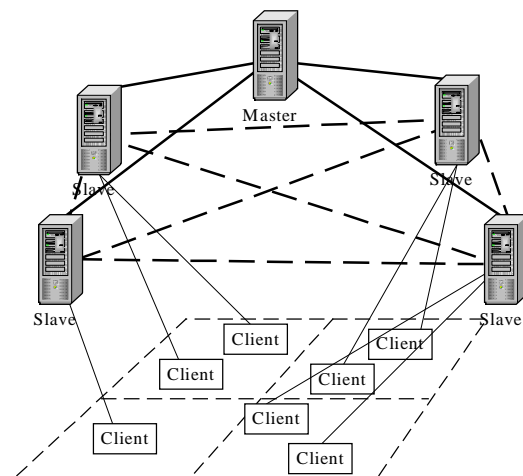


Figure 2. Partitioning a virtual world across multiple slave servers

In this paper, the character, or avatar, manipulated by each client site is modeled as the Local Object as illustrated in Figure 3. The status information of this avatar will then be duplicated in the server site. We model this image on the server site as the Object Proxy of the Local Object on the client site. The contents of the Object Proxy will be further forwarded to other clients so that other clients can embody its image. We call this embodiment as the Clone Object of a Local Object on a separate site. That is, each client site contains a Local Object that is controlled by the user and Clone Objects of other remote users. On the other hand, the slave server will create an Object Proxy for each client within its district. The Object Proxy is updated by the information sent by the Local Object on

the client site. It will then in turn forward the update data to all Clone objects that are located within its AOI.

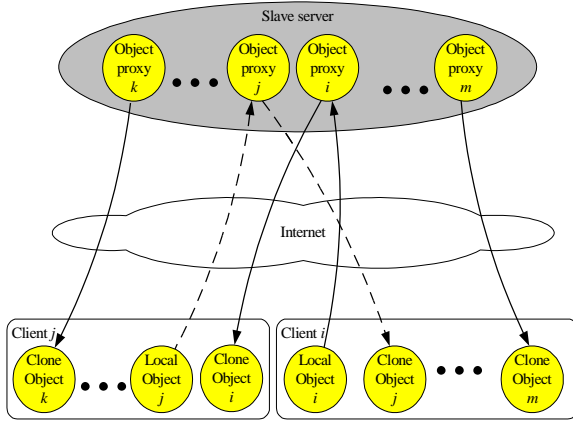


Figure 3. The Local Object, Object Proxy, and Clone Object

4.2 The Mathematical Model

As indicated by Singhal and Zyda[15], there are four issues that will affect the scalability and performance of the network virtual environment. Among them, controlling visibility of data and changing the net-VE network software architecture are two issues related to the server site. In addition, the Area-of-Interest is the technique that is often used to solve the issue of “controlling visibility of data”. We can model the Area-Of-Interest technique as follows.

Let U be the virtual world that is managed by a server and U is partitioned into n cells. Let S_i represents a fixed-sized cell of U with index i and $S_i \cap S_j = \phi, \forall i, j$. That is, all of the cells that comprise the virtual world are mutually disjointed. The virtual world U then can be expressed as the union of all these cells, i.e. $U = \bigcup_{i=1}^n S_i$. Hence, We can model the AOI by the following definition.

[Definition]

Let the operator \oplus define the operation to compute the adjacency relationship among these cells, $S_i, \forall i$. That is, Let $S_i \oplus S_j = 1$ if cell S_j is adjacent to cell S_i , otherwise $S_i \oplus S_j = 0$. The operator \oplus has the following properties:

1. $S_i \oplus S_i = 1 \forall i$ (Reflective)
2. If $S_i \oplus S_j = 1$, then $S_j \oplus S_i = 1$ (Symmetric)
3. If $S_i \oplus S_j = 1$ and $S_j \oplus S_k = 1$, it does not imply $S_i \oplus S_k = 1$ (Non-Transitive)

Let AOI_i be the set of cells that constitute the AOI centered on cell S_i . Then, we can express as $AOI_i = \{ S_j | \forall S_j \text{ such that } S_i \oplus S_j = 1 \}$.

That is, AOI_i is the union of S_i and all the cells that are adjacent to cell S_i . ■

With this definition, we can formalize the AOI technique as follows. When the Slave server receives the status update from client a , it will forward this message to all clients whose Object Proxies are within the AOI for client a . When the avatar of client a moves from a cell, says i , into another, says j , its AOI is then changed from AOI_i to AOI_j . The Slave server must inform clients whose object proxies are within cells of $(AOI_i \cup AOI_j) - AOI_j$ for the event of client a 's departure. The Slave server also needs to inform clients whose object proxies are within cells of $(AOI_i \cup AOI_j) - AOI_i$ for the event of client a 's arrival.

Significantly, we can use this model to estimate the efficiency of the AOI technique. Assume that there are totally m avatars within a virtual world and they are evenly distributed. Without the AOI technique, each status update from a client will cause the Slave server to send out $m-1$ message to all of the other clients. Let $|AOI_i|$ denotes the number of cells contained in the AOI. Hence, for the AOI technique, each status update from the Local Object to the Object Proxy will cause the Slave server, on the average, to send out $(m-1) * \left(\frac{|AOI_i|}{n} \right)$ messages to other clients that are within its AOI. Hence, on the average, compared to the conventional broadcasting method that requires the Slave server to send out $m-1$ messages for every status update, the AOI approach can reduce approximately $(1 - \frac{|AOI_i|}{n})$ percentage of the message flow inside the network.

4.3 The Shadow Object

Even though the above approach can significantly reduce the number of the messages flows, however, care must be taken when this technique is applied. Since the entire virtual world is partitioned into different regions and managed by respective Slave servers, the migration of an avatar from one server into another is an important issue for the multiple server approach of the networked virtual environment. The simplest solution is to discharge the avatar from the original slave server first and then login to another slave server. However, this approach has a serious problem of abrupting visual effect on the client site. That is,

the user will suddenly perceive an empty virtual world followed by a gradual emersion of other avatars.

In order to smoothly transfer an avatar from one server into another, we must overlap the border of two adjacent regions that are managed by two different Slave servers. We call these two servers as the logically chained servers, or chained servers for short. When an avatar is in a border district of a server, a copy of its information must also be forwarded to its chained server so that the chained server can be aware of the existence of that avatar and forward other clients information to it. We model this copy of avatar information as the Shadow Object of the original avatar. That is, when an avatar arrives the border district of a server, its Shadow Object will be created on its chained server. For example, Figure 4 illustrates two partial regions that are managed by two chained servers. The dark shaded cells, i.e. cells $k+3$, $j+3$, and $i+3$, represent the border districts between two chained servers. The white man in Figure 4 represents the Shadow Object of the black man in the chained server.

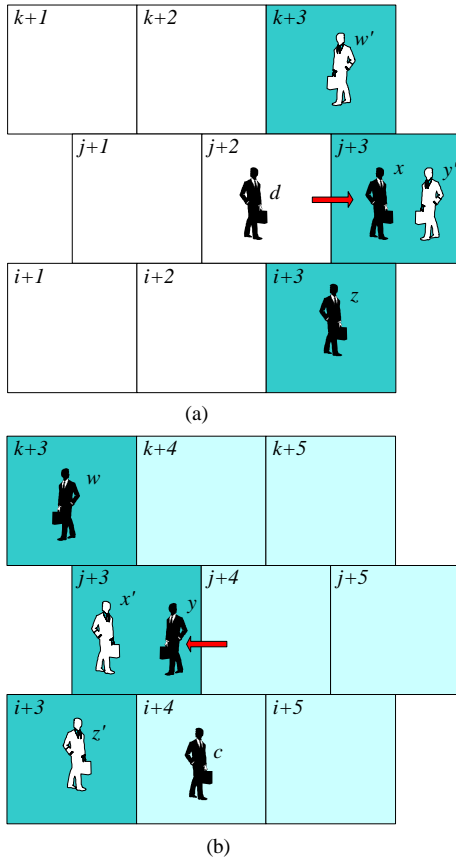


Figure 4. The Shadow Object for the avatar migration

Similar to the Object Proxy, the chained server will also calculate the AOI of the Shadow

Object and forward the information of the Shadow Object to all of the clients within its AOI. However, in order to prevent a client from receiving message from both Object Proxy and Shadow Object at the same time, the information of the Shadow Object will only forward to clients who are not in the border district. When an avatar eventually moves into its chained server, since the information of that avatar already exists in the form of the Shadow Object, the chained server only needs to switch its status from Shadow Object to Object Proxy. By this way, we can smoothly transfer an avatar from one server into another without any perceptual glitch. Hence, the algorithm for the slave server to manage the message flow is shown as follows:

Procedure Status_Update

```

BEGIN
  IF client a is moved from cell, says i, to
  another cell, says j,
  THEN (AOIi is changed to AOIj)
  BEGIN
    Notify the clients whose Object
    Proxies or Shadow Objects are
    within cells of (AOIi ∪ AOIj) -
    AOIj for the departure of client
    a;
    Notify the clients whose Object
    Proxies or Shadow Objects are
    within cells of (AOIi ∪ AOIj)
    -AOIi for the arrival of client
    a;
  END
  // Check if client a is in the border. If it does,
  then perform server to
  // server data channel connection channel
  for client a.
  IF client a is an Object Proxy in Slave
  server
  THEN
  BEGIN
    Update Object Proxy of client a;
    Forward this status change message
    to all clients whose Object
    Proxies or Shadow Objects are
    within cells of client a's AOI;
  END
  ELSE (client a is a Shadow Object in Slave
  server)
  Forward this status change
  message to all clients whose
  Object Proxies are within cells of
  client a's AOI exclude border
  cells;
END

```

END

5. Conclusion and Future Work

This paper presents a method to construct a scalable networked virtual environment. In addition, the issue of partitioning a virtual environment across multiple servers is also studied. As indicated by Singhal and Zyda[15], four essential issues to build a scalable networked virtual environment are optimizing the communication protocol, controlling the visibility of data, taking advantage of perceptual limitations, and enhancing the system architecture. This paper proposes a hierarchical server structure to manage a large-scale virtual environment by multiple servers. However, when we manage a large-scale virtual environment, care must be taken when we transfer an avatar from one server to another. In order to prevent the abrupt visual effect, a region managed by a server must be overlapped on the borderland with its adjacency region managed by different servers. With this approach, when an avatar is within the borderland, its Shadow object can ensure that other clients on the contiguous server will receive its status update messages. In this way, a smooth status transfer is achieved when this avatar migrates from one server into another. Hence, a scalable networked virtual environment can be easily designed based upon the proposed server cluster structure and avatar migration technique.

Finally, the proposed mechanism is currently under implementation. Further study of the server cluster will be continued to fully explore the issues of providing scalable large-scale networked virtual environment.

References

- [1] 黃俊堯, 張嘉麟, 鄭育鎔, "網際網路上三度空間多人互動系統之空間分割法技術研究", 第二屆海峽兩岸航空航天技術暨流固耦合學術研討會, 大陸北京, 87年7月, pp.43-50.
- [2] ANSI/IEEE Std. 1278-1993, "Standard for Distributed Interactive Simulation (DIS) – Application Protocols", March 1993.
- [3] J. W. Barrus, R. C. Waters, and D. B. Anderson, "Locales and Beacons: Supporting Large Multiuser Virtual Environments", IEEE Computer Graphics and Applications, November 1996, Vol. 16, No. 6, pp.50-57.
- [4] S. Benford, C. Greenhalgh, T. Rodden and J. Pycock, "Collaborative virtual environments", CACM, Volume 44, Issue 7 (2001), pp.79-85.
- [5] J. S. Dahmann, F. Kuhl, and R. Weatherly, "Standards for Simulation: As Simple As Possible But Not Simpler – The High Level Architecture for Simulation", Simulation, Vol. 71, No. 6, December 1998, pp.378-387.
- [6] T. A. Funkhouser, "RING: a client-server system for multi-user virtual environments", Proceedings of the 1995 symposium on Interactive 3D graphics, April 9 - 12, 1995, Monterey, CA USA, pp. 85-92.
- [7] C. Greenhalgh and S. Benford, "Massive, A Collaborative Virtual Environment for Teleconference", ACM Trans. on Computer Human Interfaces(TOCHI), Vol. 2, No. 3, ISSN 1073-0516, September 1995, pp.239-261.
- [8] Hagsand, "Interactive Multi-user VEs in the DIVE System", IEEE Multimedia, Vol. 3, No. 1, Spring 1996, pp. 30-39.
- [9] "HLA Technical Specifications", Available at <http://hla.dmsomil/>
- [10] Y. Honda, K. Matsuda, J. Rejimoto, and R. Lea, "Virtual society: extending the WWW to support a multi-user interactive shared 3D environment. Procs. of VRML'95, San Diego, CA. Aug 1995. Also available at <http://www.csl.sony.co.jp/person/rodger.html>
- [11] J. Y. Huang, C. T. Fang-Tson, and J. L. Chang, "A Multiple User 3D Web Browsing System", IEEE Internet Computing, Vol. 2, No. 5, Sept/Oct 1998, pp.70-80.
- [12] M.R. Macedonia et al., "NPSNET: A Multi-Player 3D Virtual Environment Over The Internet", ACM SIGGRAPH Special Issue on 1992 Symposium on Interactive 3D Graphics, (Cambridge, MA), 1995, pp. 93-94.
- [13] M.R. Macedonia, M.J. Zyda, D.R. Prat, D.P. Brutzman, and P.T. Barham, "Exploring Reality with Multicast Groups: A Network Architecture for Large-scale Virtual Environments", Proc. 1995 IEEE Virtual Reality Annual International Symposium(VRAIS95), March 1995.
- [14] G. Singh, et al., "BRICKNET : Sharing

Object Behaviours on the Net",
Proceedings of VRAIS'95, IEEE Computer
Society Press, March 1995, pp19-25.

- [15] S. Singhal and M. Zyda, "Resource Management for Scalability and Performance", Networked Virtual Environment – Design and Implementation, Addison-Wesley Pub Co., ISBN: 0201325578, (1999), pp.181-248.
- [16] S. K. Singhal, D. R. Cheriton, "Using Projection Aggregations to Support Scalability in Distributed Simulation", Proceedings of 1996 International Conference on Distributed Computing Systems, IEEE 1996.
- [17] D. J. VanHook and J. O. Calvin, "Data Distribution Management in RTI 1.3", Simulation Interoperability Workshop, Spring 1998, paper no. 206. Also available at
<http://dss.ll.mit.edu/dss.web/98S-SIW-206.html>