pISSN 1976-913X
eISSN 2092-805X

J Inf Process Syst, Vol.9, No.3, September 2013 http://dx.doi.org/10.3745/JIPS.2013.9.3.489

A Simulation Model of Object Movement for Evaluating the Communication Load in Networked Virtual Environments

Mingyu Lim* and Yunjin Lee**

Abstract—In this paper, we propose a common simulation model that can be reused for different performance evaluations of networked virtual environments. To this end, we analyzed the common features of NVEs, in which multiple regions compose a shared space, and where a user has his/her own interest area. Communication architecture can be client-server or peer-server models. In usual simulations, users move around the world while the number of users varies with the system. Our model provides various simulation parameters to customize the region configuration and user movement pattern. Furthermore, our model introduces a way to mimic a lot of users in a minimal experiment environment. The proposed model is integrated with our network framework, which supports various scalability approaches. We specifically applied our model to the interest management and load distribution schemes to evaluate communication overhead. With the proposed simulation model, a new simulation can be easily designed in a large-scale environment.

Keywords—Networked Virtual Environments, Simulation Model, Load Distribution, Interest Management

1. INTRODUCTION

A networked virtual environment (NVE) is a software system in which dispersed users interact with each other by sharing their context [1]. There are various application areas that belong to an NVE, such as multi-user online games, virtual communities, collaborative design systems, military simulation systems, and so on. One of important research issues in NVEs is how to support the type of scalability that provides a user with an interactive performance, even if the number of users increases. There have been many efforts to reduce the requirement of the network bandwidth and the communication overhead by interest management using multicast [2-4] and load distribution approaches among multiple servers [5-7]. These researches verify the performance NVE systems by simulating a shared environment in which numerous users exist.

However, they separately developed the different schemes in order to compare the performance of different NVE systems. It is a very time consuming task; although it should be noted that some simulation components can be reused for different similar experiments. Another problem is that it is difficult to measure the scalability with the actual user.

Manuscript received June 20, 2012; accepted July 24, 2012.
Corresponding Author: Yunjin Lee
* Dept. of Internet & Multimedia, Konkuk University, Seoul, Korea (mlim@konkuk.ac.kr)
** Division of Digital Media, Ajou University, Suwon, Korea (yunjin@ajou.ac.kr)

In this paper, we propose a common simulation model that can be reused for different performance evaluations. To this end, we analyze common features of NVEs in which a world is divided into regions, and a user has his/her own interest area. In usual simulations, users move around the world, while new users remain logged into the system. Our model provides various simulation parameters to customize a specific environment. Furthermore, our model designs a way to mimic a lot of users in a simple experiment environment. The proposed model is integrated with our network framework, which supports various scalability approaches. We specifically applied our model to the interest management and load distribution schemes. With the proposed simulation model, a new simulation can be easily designed in a large-scale environment.

2. INTEREST MANAGEMENT

Interest management means that a user receives messages only from those whom he/she is interested in communicating with [2]. Users' interests are not only represented by the spatial distance between them, but also by social groups such as those who have the same hobbies or those who want to buy the same items in a shopping mall, for example. The interest management can be classified as follows: region-based management divides a virtual world into several geographical regions, and a user specifies the regions that he/she is interested in. Aura-based management localizes the spatial area of interest of an individual user in the small surrounding area [3]. Class-based management represents a user's interest according to his/her logical context. For example, a user can be interested in only a specific type of object regardless of their location. Hybrid management combines the aforementioned schemes together.

Even though the existing recent bodies of work have addressed fine-grained filtering mechanisms, they still incur other drawbacks in terms of the limitation of available multicast addresses or the management overhead of multicast groups. This makes the system less scalable as the number of users increases. Our previous research proposed a new interest management to solve the problem [4]. It can, not only reduce the number of messages, but also the number of multicast addresses without significant computational overhead. We assume that a virtual world is divided into rectangular regions, and that each user has a spatial interest area (IA) that is similar to the aura concept. Interaction among users with the same interest occurs only in a user's IA. Users with the same interest dynamically form a group when they move closer to each other. Whenever a group member updates his/her state, he/she sends update messages to other group members via multicast. On the other hand, when a group is within the interest area of a user who has a different interest, a representative user of the group sends him/her the update information of the group at a low frequency. The member user with the lowest number of ID value or the longest duration time of interaction among the other members of the group is elected as a representative user.

Interest management is also applied when a user interacts with others in different regions. The idea is that the filtering on inter-region interaction is used to enable users to interact mostly with adjacent users, instead of with distant users in the neighboring regions. Our scheme uses a sub-region concept to select only a subset of users from neighboring regions whose members have high possibility of interaction with users in the current region. This subset of users forms another multicast group. This enables users in the region to avoid receiving all of the update messages from neighboring regions. The size of the sub-region scope is dynamically changed according to

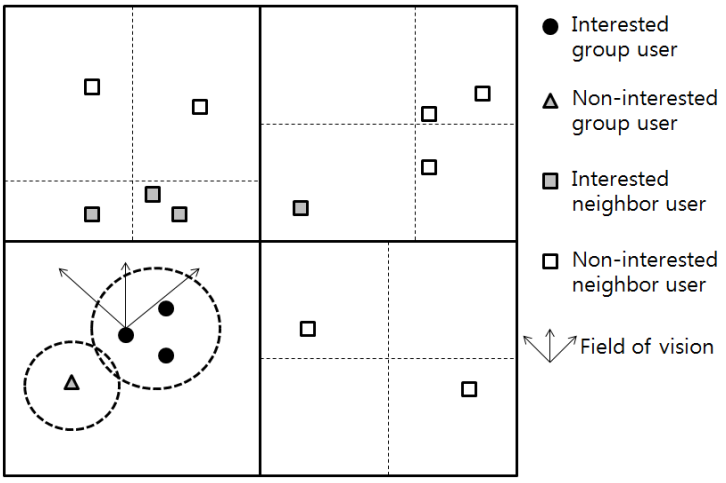

Fig. 1. Interest management

the distribution of uses in the neighboring regions so that the inter-region interaction does not deteriorate the interactive performance regardless of not only the number of users, but also of their distribution. In addition, only sub-regions in the neighboring regions, which are in the field of vision of a user, are selected for further filtering. Fig. 1 shows the overall concept of the interest management.

The various interest management approaches require performance evaluation in terms of communication and computational overhead, but different simulation is performed by a different experiment code, which burdens the simulation task.

3. LOAD DISTRIBUTION

Load distribution in multi-server-based NVE systems is another effective solution for achieving scalability. In the client-server NVEs, heavily loaded servers divide the user or region information that they are managing, and transfer them to lightly loaded or idle servers. Load distribution schemes can be classified into two approaches according to how many servers are involved in the balancing task. These two approaches are the global and local schemes. The global approach finds an optimal solution, as all servers are involved in the load distribution [5]. However, this does not fit for NVEs, which require proper interactive performance, because it takes so much time to calculate an optimal solution and to balance the workloads among all of the servers. To sustain the required interactive performance, the local approach is the most suitable, because it considers only the neighboring servers [6]. However, regardless of the number of servers, load distribution, even between only two servers, must pay a considerably high migration overhead. The state transition of users or regions takes a long time, because it requires synchronization between servers and clients. During the state transition, the system must enter into a holding state, because not only users who belonged to the previous server but also their information should be transferred to the new server in order to avoid inconsistency. If migration frequently occurs and transmits a large amount of state information, the migration overhead causes

a significant impact on the overall performance of a system, and users feel uncomfortable to interact with each other due to repeated interference from the load migration task.

Our load distribution research proposed an efficient load distribution mechanism that distributes the load in a more fine-grained manner [7]. A group of servers takes charge of the same portion of the world to avoid information transfer overhead, while the existing approaches allow each server to manage the disjointed regions in a virtual world. At run time, our mechanism classifies messages into three task types according to the occurrence history of messages and the time taken for processing them. These three task types are as follows: CPU intensive tasks, network I/O intensive tasks, and normal tasks. Task distribution is handled differently according to the task type. When a server becomes overloaded, it distributes CPU intensive tasks to neighboring servers while its clients do network I/O intensive tasks to these servers. By sharing region information and only distributing tasks among servers, an overloaded server neither reorganizes region information, nor does it synchronize the state transition procedure, which remarkably reduces the distribution overhead. Fig. 2 shows the concept of task-based load distribution.

Even though different load distribution schemes have common factors for performance evaluation, such as regions, user movements, and the client-server model, current simulation is conducted as separate programs. If these common elements are reused in a different simulation, it makes the development of NVE simulation more efficient than before.

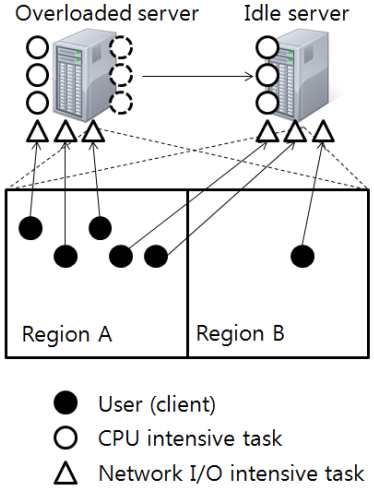

Fig. 2. Task-based load distribution

4. NVE NETWORK FRAMEWORK

The network framework that we developed is focus on providing various scalability approaches for NVE systems [8]. As a software library, the framework provides NVE application developers with APIs that support different application-level communication functionalities, such as communication architecture (client-server or peer-server models), the membership management of users (multiple sessions and regions), message definition, and transmission. Particu-

larly, the framework includes the interest management and load distribution mechanisms that an operator can set in a configuration file.

In the framework, a user belongs to a region, and he/she is interested in other users in the current and neighboring regions. Users in the same and neighboring regions can interact with each other. A user is represented by a user object, and a region is managed by a region manager module in the network framework. A region manager maintains the list of current users. There can be separate groups of regions as well in the framework. A session is defined as the independent interaction space in which multiple regions exist. A session manager module in the framework manages a session, and a session manager manages the list of region managers. The top-level interaction manager module of the framework is called a multi-session manager, which manages the list of session managers.

Using the network framework, we implemented various scalability approaches. However, for performance evaluation, we had to develop separate simulation programs to compare different schemes. Therefore, we tried to reduce the effort of simulation development by analyzing the common factors of scalability approaches.

In the next section, we describe a common simulation model and how it is integrated with the network framework.

5. COMMON SIMULATION MODEL

In this section, we describe the core modules, region management and the non-playable character (NPC) modules, for emulating NVE communication performance. Most NVE systems realize a shared space with the region and aura concepts for the interest management and load distribution schemes. In our simulation model, the shared space is assumed to be the collection of 2D rectangular regions. The information of a region includes a string name, a multicast address, and the size and position information. The multicast address is used for the region-based multicast transmission. The size and position information is maintained by the upper left and lower right points of a region. Simulation designers can set the region information in a server-side configuration file, as shown in Fig. 3.

Our current network framework supports a user model for NVEs by providing the CAtlasUser class, which includes user information such as a user name, host information, position, orienta-

```
#region configuration
REGION_NUM                          4

REGION_NAME1                        r1
REGION_SERVER_ADDR1                 220.69.185.101
REGION_SERVER_PORT1                 8001
REGION_SERVER_UDP_PORT1             9001
REGION_ADDR1                        224.1.1.2
REGION_PORT1                        7001
REGION_TLX1                             -250
REGION_TLY1                             0
REGION_TLZ1                             0
REGION_BRX1                             0
REGION_BRY1                             0
REGION_BRZ1                             250

...|
```

Fig. 3. Region configuration

tion, aura range, and so on. A user object is created whenever a new user logs into the system, and is controlled by an actual user. The communication performance of a server and a client is measured by the operation of a lot of users, but it is difficult to evaluate with real users due to the lack of experiment participants. That is the reason why we devised a non-playable character (NPC) object. A NPC object has the role of a normal user, and is controlled by the network framework according to the simulation configuration. The fundamental operation of a NPC is to join a shared space by following the same login steps and to generate events. A NPC usually sends a position update event while it moves around the space. Additionally, we added a new class (CAtlasNPCManager) to manage multiple NPCs and to define their different actions. The NPC manager class creates each NPC object as a separate thread because it conducts an independent task from other NPCs. When NPCs are created, they are included in a region and just transmit their local events. As a NPC only has the role of generating the network traffic of an additional user, it does not receive any event from other users or NPCs so that the network framework consumes useless network bandwidth. The network framework is redesigned such that it does not deliver events to NPC objects. Despite the lack of a receiving function for a NPC object, other users and servers cannot distinguish it from normal users.

In a region, a NPC object behaves in the same manner as a normal user. When a NPC generates a new event, it sends the event to the multicast address assigned to a region to which it belongs, if the simulation is conducted in the peer-server model. If the client-server model is used, then a NPC sends an event to a server. A NPC also checks the region boundary whenever it moves, and changes the current region to a new region if it is required.

When the simulation starts, the network framework creates NPC objects. Our simulation model can decide the number of NPCs and creation interval by the configuration file. Therefore, the proposed model realizes the increase of the number of connected users at a different creation rate. When a NPC is created, it starts to randomly move in a region. For the random movement, the NPC manager module can set the number of destination points. If the number is greater than zero, a NPC randomly creates a list of the fixed number of destination points in the shared space, and repeatedly moves toward them in order. The list of destination points is written in a separate file, and a NPC reads the information from the file. If the destination number is -1, a NPC chooses another random destination whenever it arrives at the previous one. The destination point can be set in the same region or different region based on the simulation configuration as well. When a NPC reaches the destination, it continues to move to the next destination. The configuration file also determines the NPC speed and movement interval because a NPC can have a different speed and sometimes it can stay in one position. NPC speed means how many position units it moves in one step. The movement interval is defined as the time in milliseconds during which a NPC does not move.

The NVE simulation should also decide when the NPC operation is finished. To this end, the NPC manager module defines the simulation duration value in milliseconds. If the duration value is -1, which is the default value, the simulation never stops until it receives an explicit termination event. If the duration is greater than zero, the simulation finishes after the designated time.

The overall interaction of simulation modules is shown in Fig. 4. The NPC manager module interacts with the existing multi-session manager module in the network framework in order to obtain region information. The NPC object module interacts with the existing region manager module in the network framework in order to join/leave a region in the shared space.

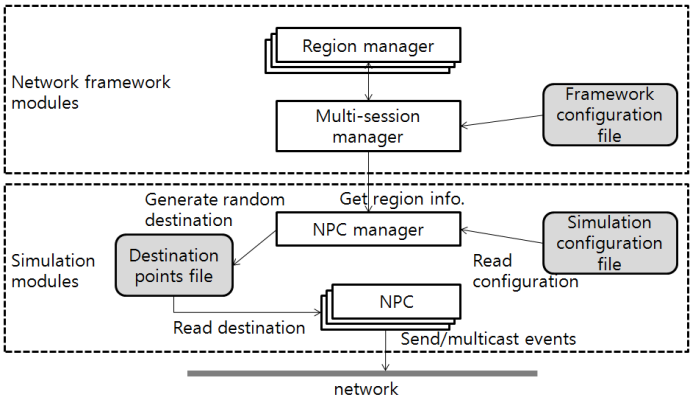

Fig. 4. NPC modules interaction

6. Simulation Results

The common simulation model is integrated with our network framework, and is simulated with the simple experimentation environment. As shown in Fig. 5, we used the following five machines: one gateway server for the users' login process, two region servers managing regions, one client that has the role of a real user, and one NPC generator that creates and manages remote users. Under the simulation environment, we conducted the evaluation of the communication overhead of the interest management and load distribution schemes. In the simulation scenario, a user and NPCs enters a region and start to generate events according to specified schemes.

For the interest management, we conducted various experiments, and Fig. 6 shows one of the results that applied different filtering levels, which in this case was the number of messages received by a user. If we only use the inter-region interaction management without adaptable scoping, only the sub-regions in neighboring regions filter the messages and the filtering level is low. When adaptable scoping is added to inter-region management and both inter-region and intra-region managements are combined, the filtering level increases.

For the load distribution approaches, we also conducted various experiments, and Fig. 7

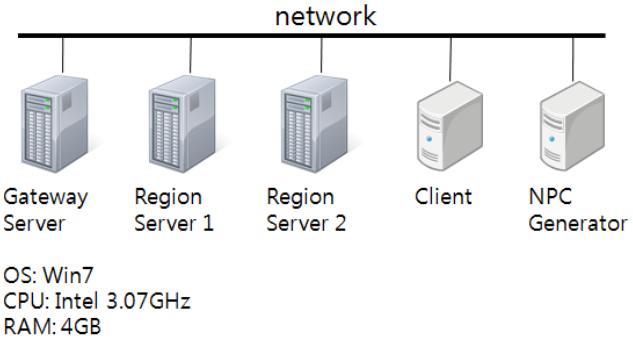

Fig. 5. Experimental environment

shows one of them, which is the time taken to complete the load distribution task. It ensures that the existing local load-balancing scheme takes much more time than the task-based mechanism. due to synchronization overhead. The overhead of the existing scheme increases exponentially due to the required steps for the load migration according to the migrated users, while that of the task-based scheme increases gradually.

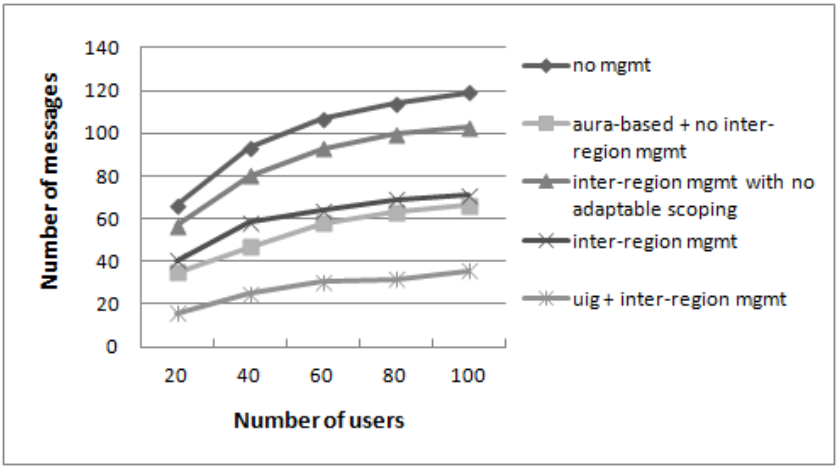

Fig. 6. Average number of messages

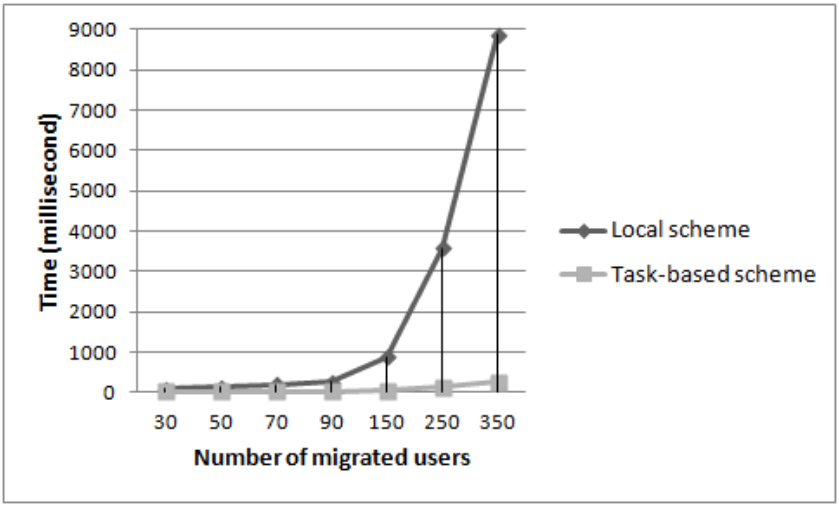

Fig. 7. Time to complete load distribution (region server 1)

7. CONCLUSION

In this paper, we proposed a common simulation model for evaluating the communication overhead of large-scale NVE systems. We analyzed that most of the communication load stems from the number of users and their movement events. We designed a shared space with a flexi-

ble configuration of regions, and devised a NPC model to realize the real traffic of a lot of users with a few connected machines. The proposed model was integrated with the network framework and applied to the different interest management and load distribution approaches. For our future work, we plan to extend the simulation model so that it can simulate other object events such as object creation, movement, rotation, scaling, and so on.

ACKNOWLEDGEMENTS

This research was supported by the MSIP (Ministry of Science, ICT&Future Planning), Korea, under the ITRC (Information Technology Research Center) support program (NIPA-2013-H0301-13-3006) supervised by the NIPA (National IT Industry Promotion Agency).

REFERENCES

[1] S. Singhal and M. Zyda, *Networked Virtual Environments: design and implementation*, Addison-Wesley, New York, 1999.

[2] K. Morse, L. Bic and M. Dillencourt, "Interest Management in Large-scale Virtual Environments," *PRESENCE - Teleoperators and Virtual Environments*, vol. 9, no.1, 2000, pp. 52-68.

[3] C. Greenhalgh and S. Benford, "MASSIVE: A Collaborative Virtual Environment for Teleconferencing," *ACM Trans. Computer-Human Interaction*, vol. 2, no. 3, 1995, pp. 239-261.

[4] S. Han, M. Lim, D. Lee and S. Hyun, "A Scalable Interest Management Scheme for Distributed Virtual Environments," *Computer Animation and Virtual Worlds*, vol. 19, no. 2, 2008, pp. 129-149.

[5] P. Morillo, J. Orduña, M. Fernández and J. Duato, "Improving the Performance of Distributed Virtual Environment Systems," *IEEE Trans. Parallel and Distributed Systems*, vol. 16, no. 7, 2005, pp. 637-649.

[6] J. Chen, B. Wu, M. Delap, B. Knutsson, H. Lu and C. Amya, *"Locality Aware Dynamic Load Management for Massively Multiplayer Games," Proceedings of the Tenth ACM SIGPLAN Symposium on Principles and Practices of Parallel Programming*, Chicago, Illinois, USA, June 15-17, 2005, pp. 289-300.

[7] M. Lim and D. Lee, "A Task-Based Load Distribution Scheme for Multi-Server-Based Distributed Virtual Environment Systems," *PRESENCE - Teleoperators and Virtual Environments*, vol. 18, no. 1, 2009, pp. 16-38.

[8] D. Lee, M. Lim, S. Han and K. Lee, "ATLAS – A Scalable Network Framework for Distributed Virtual Environments," *PRESENCE – Teleoperators and Virtual Environments*, vol. 16, no. 2, 2007, pp. 125-156.

Mingyu Lim

He received a Ph.D. degree in Engineering from Information and Communications University (ICU) in 2006. During 2006-2008, He worked at MIRALab of University of Geneva in Switzerland as a senior researcher. He has been a professor at Konkuk University since 2009. His research interests are in the area of distributed systems, networked virtual environments, and ubiquitous computing systems.

Yunjin Lee

She received a Ph.D. degree in Computer Science and Engineering from the Pohang University of Science and Technology (POSTECH) in 2005. She has been a professor at Ajou University since 2008. Her research interests include non-photorealistic rendering, 3D mesh processing, and data compression.