

HIERARCHICAL LOAD TESTING ARCHITECTURE USING LARGE SCALE VIRTUAL CLIENTS

Bum Hyun Lim, Jin Ryong Kim and Kwang Hyun Shim

Digital Content Research Division
Electronics and Telecommunications Research Institute
Daejeon, Korea
{ibh63427, jessekim, shimkh}@etri.re.kr

ABSTRACT

In this work, we develop a hierarchical load testing architecture using large scale virtual clients to reduce the testing time and ensure the stability of the server for distributed applications. It explicitly secures the stability of the servers for networked virtual environment and at the same time, it elaborately generates actual loads for testing the performance of the servers. Our agent based load testing architecture provides variety of interactions of virtual entities in the virtual worlds to perform realistic simulations. Simulation results illustrate that our proposed architecture ensures the stability and capacity of the servers for both massively multiplayer online games and peer-to-peer network games.

1. INTRODUCTION

Advances in computer technology in the entertainment industry are enabling people to capture their activities and experiences as greater volumes of ever-richer media. Furthermore, advances in broadband network technologies, home networking, and wireless mobile technologies have opened up a new era of ubiquitous. With the advances in computing, communication, and multimedia technologies, the issue of testing client-server system has been attracting interests from the researchers due to growing use of distributed applications [1].

Generally speaking, it is not easy to test and measure the performance of client-server based system due to its complexity. In order to solve this complex problem and make rigid distributed applications, we propose a load testing architecture for its simulation. Wangerin et al. [2] proposed event based simulator structure for client-server model. This structure installs simulation module around the centre core and the centre core executes the manager role, which manages the simulation. The advantage of this structure is that the various types of simulations can be generally done. However, it is not scalable to simulate large

scaled concurrent users with interactions in networked virtual environment.

Applications like online games are one of the representatives for large scale of virtual environments. In online games, large scales of users enter single virtual world concurrently and interact with each other to provide the new game interest and keep explosive development trend. Online games like Lineage series [3] and World of Warcraft [4] are the major game titles for online games. However, in proposition to their success, the stability and performance of game servers become major issues in online games. To meet such requirements for game servers, large scale of game users actually participate the game to test its performance before deploying the game. In order to reduce such testing time, it is recommended simulating its environment. The industry-standard software application load testing solutions prevents costly performance problems in production by generating thousands of loads. It is helpful to detect dead lock, memory leak, thread leak, and normal synchronization problems caused by fault coding routines and timing conditions. However, such application load testing tools are available for only web or system applications [5]. In this work, we develop a load testing architecture and application for networked virtual environments focused on client-server based online games [6].

2. HIERARCHICAL LOAD TESTING ARCHITECTURE

A load testing architecture for networked virtual environment should be able to create hundreds to thousands loads to ensure the stability of its servers. To achieve highly loading environment, it should trace the programming bugs of the server and measure service quality of the clients.

We present a hierarchical load testing architecture and it has following characteristics.

- 1. To create hundreds of thousands of loads, we propose a hierarchical load testing architecture. We put one host as a master to manage agent hosts.*

2. There exist several agent hosts which actually generate virtual clients.
3. The virtual clients can connect with servers and create actual loads and they are controlled by a master host via their agent host.
4. The number of agent hosts can be increased constantly in parallel.
5. The master host provides realtime hardware performance and service speed to measure service quality.
6. The agent hosts can create as many virtual clients as up to their hardware performance.
7. We design IP network emulator to test with more realistic network environment.
8. We design analysis to analyze simulation result using GUI and various graphs.

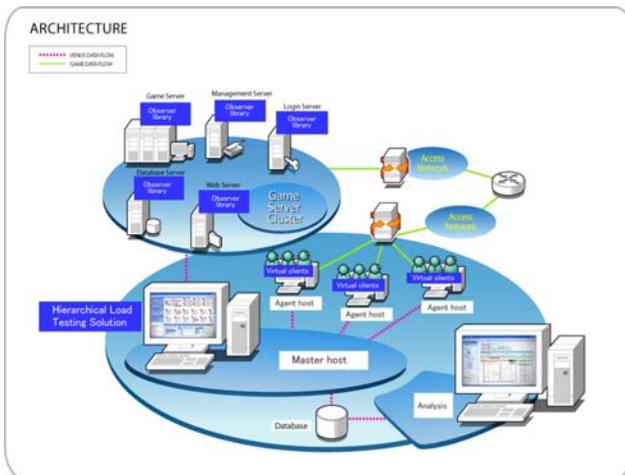


Fig. 1. Hierarchical load testing architecture

Figure 1 illustrates a hierarchical load testing architecture for networked virtual environment. It consists of master host, a number of agent hosts, virtual clients on their agent host, network emulator and analysis.

There exists a protocol between each component that is linked to the network. We design simulation protocol between the master and agent hosts, and between agent hosts and their virtual clients.

2.1. Master host

The master host controls its agent hosts and deliver event packets to agent hosts. The master host exists as an application entity and manages overall simulation. The master host is also connected to the observer component located in the server system to get the feedback information from servers such as CPU, memory and network usages.

2.2. Agent host

One or more agent hosts communicate with a master host using simulation protocols. They actually interpret received

commands from a master host and manage their virtual clients. The roles of agent hosts are to create, control, remove or monitor their virtual clients.

2.3. Virtual client

The virtual clients are the replacement of actual client and they are controlled by a master host. The virtual clients enter the virtual worlds (i.e. online game world) by connecting the server. They join the virtual worlds like actual client does and interact with each other using an application of master host.

2.4. Observer library

Observer library resides in the server system and provide performance information to the master host. Observer library simply calculates the performance index of server systems and send them to the master host. Performance index indicates resources usage ratio such as CPU, network and memory.

2.5. Network emulator

IP Network Emulator makes virtual network environment by modeling the behavior of real network environment and generates real network packets physically. By using network emulator, we can test the stability of server by modulating network packets. The roles of network emulator are to delay sending/receiving packets from NDIS driver, control packet drop probability, copy network packets, and control network bandwidth.

2.6. Analysis

Analysis is an application to analyze simulation results. It graphically displays data stored in the database and checks the value of data immediately. Because it graphically displays changes in number of virtual clients and response time, all sorts of corresponding situation during simulation can be easily analyzed.

3. LOAD TESTING SCENARIOS

3.1. Virtual client implementation

As we mention in earlier section, we can implement virtual client using our virtual client template. This template provides the networking connection with the agent host. To make a light-weight virtual client, we remove all the graphical resources from an actual client application and converge it into the template.

3.2. Types of load tests

Table 1. Basic test type

Type	Method
Login test	Execute login/logout increasing number of virtual client
Login server performance test	Test stability of login server with login/logout possibility
Movement test	Ensures stability and capacity of server with test of movement
Response time test	Measure the response time with round-trip delay from virtual client to server
Casual game test	Provides test command set for casual games
P2P test	Makes virtual network environment by modeling the behavior of real network environment and physically modulates network packets

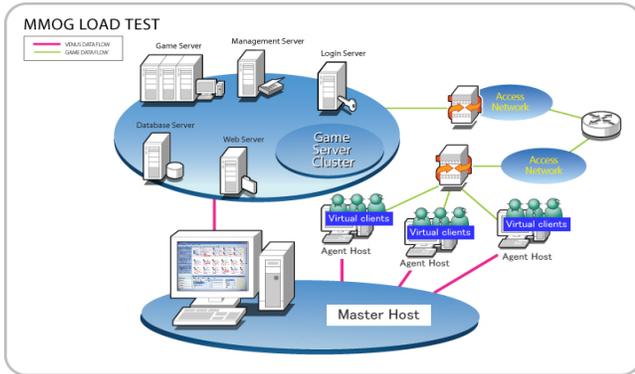


Fig. 2. MMOG load testing scenario

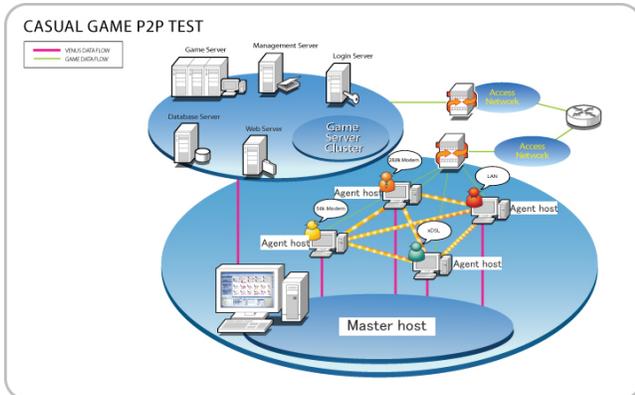


Fig. 3. P2P based network game scenario

Using our load testing architecture, we can easily perform variety of stress tests on networked virtual environment systems such as large scale login/logout, chatting messages, and population convergence tests.

Table 1 illustrates the list of basic test that our proposed architecture can perform. However, various simulations can be done under combination of method or method can be customized under user's own interest.

Figure 2 and 3 illustrate both MMOG load testing and P2P based network game scenarios, respectively.

4. EXPERIMENT

We carried out two types of load testing scenarios: MMORPG [7] and casual type game application [8].

4.1. MMOG load tests

In MMOG application, the server has a minimum unit called cell and the virtual game world consists of these cells. The server only broadcasts necessary information to the clients within the area of interests (AOI). By doing this, we could reduce network overload. For the experiment, 5,000 of virtual clients with 10 agent hosts were set up.

We implemented virtual client for this application to measure the maximum client number for the game server and tried to verify the stability of the processing logic of sever.

4.1.1. Login server test

To verify the stability of login server, the total of 5,000 virtual clients perform login/logout. The login server has 2.8 GHz of CPU, 1Gbps of network, and 10,276 MBytes of memory for the experiment. Figure 4 illustrates the increment of number of virtual clients. In Figure 4, the number of virtual clients is increased in 500. Figure 5 illustrates the result for ping time and its CPU of login server. During the load testing, it is noticed that the resources of CPU reached at 100% and ping time varies between 100 and 300 msec.

4.1.2. Game server test

Figure 6 illustrates the changes of response time when a number of virtual clients move within the same cell. The number of virtual clients is increased in 1,000 up to 5,000. It is noticed that when the number of virtual client reaches at 5,000, the response time starts to increase explosively. It is because the server has to process N^2 (where N is the number of virtual client).

Figure 7 illustrates the changes of response time when numerous virtual clients hunt for mobs. The number of virtual client is increased in 250 up to 2,500. It is noticed that when the number of virtual client reaches 1,500, the response time passed over 10,000 msec. Each test illustrates the bottlenecks and capacity of each server.

4.2. P2P based network games tests

In peer-to-peer network game tests, we set virtual network environment for each client host to get their own network environment. For networking factors, we changed packet delay, packet drop probability, packet duplication and their network bandwidths, for each four client hosts as illustrated in Table 2. It is noticed that this application can be influenced by network bandwidth when it reaches at 1350 bps.

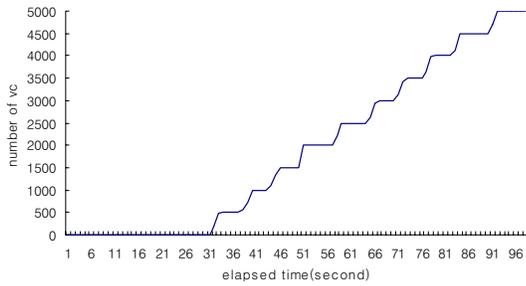


Fig. 4. Increment of number of virtual clients

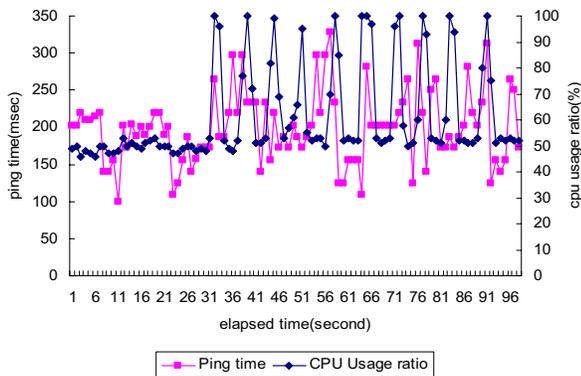


Fig. 5. Response time of login server

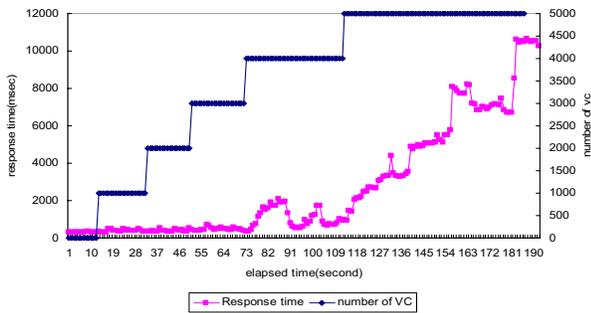


Fig. 6. Response time of movement test

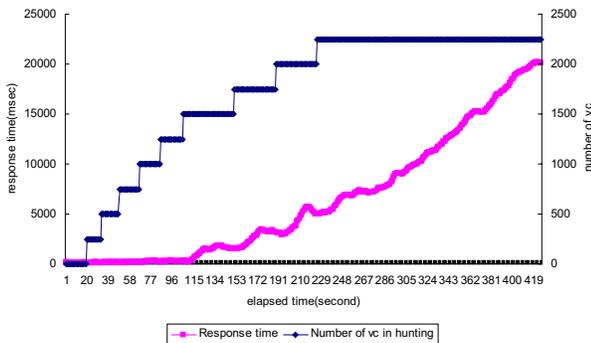


Fig. 7. Response time of battle test

Table 2. P2P test factors

Delay	Normal when delay is below 750 msec
Drop	Normal when probability if below 15%
Duplication	Not applicable
Bandwidth	1350 bps

5. CONCLUSION

In this work, we proposed a hierarchical load testing architecture to easily simulate and analyze for networked virtual environment systems. We also implemented network emulation and script components to the architecture. To verify the effectiveness of our architecture, we applied our architecture on commercial online game application. We implemented virtual client and server observer to find out the performance and ensure the stability of the servers. In result, we stably accommodated 5,000 of virtual clients in maximum.

A couple of issues should be elaborated on further and have been under our investigation. To refine the current architecture, the research on improvement of virtual client implementation has to be updated. Another issue that we concern about is its accommodation. The convenient use of our architecture and implementation must be under consideration.

6. REFERENCES

- [1] Yogesh L Deshpande, Roger Jenkins, and Simon Taylor, "Use of Simulation to Test Client-server Models", *Proceedings of the 1996 Winter Simulation Conference*, November 1996, pp. 1210-1217.
- [2] David Wangerin, Christopher DeCoro, Luis M.Campos, Hugo Coyote and Isaac D.Scherson, "A Modular Client-Server Discrete Event Simulator for Networked Computers", *Proceedings of the 35th Annual Simulation Symposium*, April 2002, pp. 125-133.
- [3] NCSOft, <http://www.lineage.com> and <http://www.lineage2.com>.
- [4] Blizzard Entertainment, <http://www.blizzard.com>.
- [5] Menasce, D.A., "Load testing of Web sites", *Internet Computing*, IEEE, July-Aug, pp. 70-74.
- [6] YungWoo Jung, et al, "VENUS: The Online Game Simulator using Massively Virtual Clients", *Lecture Notes in Computer Science*, Springer-Verlag, Volume 3398/2005, pp.589, Oct. 2004.
- [7] Dataway, <http://www.dataway.co.kr>.
- [8] ECISstudio, <http://ecistudio.com/>