Evaluation of Metaverse Server in a Widely-Distributed Environment

Mayu MATSUBARA
Ochanomizu University
l2–1–1, Otsuka, Bunkyo-ku
Tokyo 112–8610, JAPAN
mayu@ogl.is.ocha.ac.jp

Masato OGUCHI
Ochanomizu University
l2–1–1, Otsuka, Bunkyo-ku
Tokyo 112–8610, JAPAN
oguchi@computer.org

***Abstract*—”Metaverse” is a new service connecting to three dimensional virtual space constructed as electronic data through the Internet, which is expected to be spread in the future. Although metaverse is expected as a promising platform for information exchange and communication among Internet users, it has several shortcomings such as requiring relatively high-performance client terminals and its long response time in some cases. We have analyzed the problem of the response time from a viewpoint of load of the server in order to make an efficient and high-performance server.**

OpenSim server that provides the metaverse service with open source software has been constructed and evaluated. In this paper, we have made OpenSim server in grid mode to analyze distributed metaverse service. We have evaluated metaverse server performance during the phase of user login to metaverse service and teleport from one region to another. That is to say, we have focused on a server side of metaverse system, analyzed its behavior, and clarified the cause of its long response time.

***Keywords*-Metaverse; OpenSim; Second Life; Distributed Server; Centralized Server**

I. Introduction

Recent years, the Internet culture led by user is pervading because broadband networks at home are spread and general PC performance is improved. Thus, it has been strongly desired to evolve a communication and information service among users over a network. ”Metaverse” discussed in this study is three dimensional space like real world constructed as electronic data through the Internet. By connecting to the space, we can do more real and more easier communication.

Users are not passive but becoming active creator to informartion service. Therefore Metaverse service has been attracting a lot of attention. One of the most famous Metaverse service is Second Life provided by LINDEN RESERCH laboratory in US[1]. In Japan, CyberAgent started to provide ”Ameba pigg” that is becoming popular Metaverse service[2]. Other Metaverse services are also known such as IMVU[3], Moshi Monsters[4], and Meet-Me[5].

On the other hands, it needs a high spec client PC, especially high-performance graphics, in using Metaverse service. [1] Until now, although they are not spread more than expected in this world, major companies are going into Metaverse business. Thus, they are becoming popular little by little. According to a LINDEN RESEARCH report in March 2010, one million users login Second Life in the past one month. Region scale in Second Life and the amount of original currency in circulation are increasing. In addition, we expect that the problem of low spec of client PC will be solved before long because performance of PC is advancing rapidly.

II. Metaverse

A. An Overwiew of Metaverse

Metaverse means virtual world of three dimentional space in the internet. Users can communicate with others by operating another human being that is called ”avatar” in this space. It also provides a number of services to users by using original money available within Second Life. For example, we can purchase or sell items(clothes, vihicle, and house), communicate within the company, and so on. Metaverse is similar to online game, although prepared goal (bring down enemies) is not provided by the service, and users play according to their own purpose.

1) Second Life: Second Life is one of the largest Metaverse service. As the name suggests, it provides the space to enjoy ”Second Life”. It provides visually beautiful graphics, it is possible to make objects with smooth interface, we can purchase and sell objects we made, and change LINDEN dollar which is valid in Second Life to US dollar. However, there are differences between real world and virtual world: There is no physical constraint. Avatar can fly the sky freely, and access region around the world in a moment. LINDEN RESEARCH laboratory which provides Second Life has not made the source code of Server Software of Second Life open. However they started to open the code of client viewer.

2) OpenSim: OpenSim is open-software of Metaverse server. In other words, this is platform for constructing a virtual world[3]. Since OpenSim Server is based on the client viewer of Second Life, it is possible to access from the client viewer of Second Life to OpenSim Server. Compatibility is guaranteed with Second Life Server. We can

[1] Recommended hardware requirements for Second Life client viewer is that CPU is 1.5GHz or more, Main Memory is 1GB or more, and Graphiccard is Nvidia is 6700 or more and ATI is X800 or more

construct our own server and customize it easily, because this is open source software.

In addition, OpenSim Server can be operated at two kinds of mode; standalone mode and grid mode. With standalone mode, it is possible to construct virtual world easily. OpenSim uses common data format with Second Life. Therefore, when a lot of designers create a region of virtual space, they preview its preproduction in OpenSim operated at standalone mode. By utilizing the characteristics of standalone mode, it can be used as three dimentional presentation tool, and guide at hospital and amusement park. Although standalone mode is easy for configuration, it does not have scalability.

On the other hands, grid mode is operation mode to create virtual world such as Second Life. At grid mode, by making distributed OpenSim Servers interconnect, it is possible to construct a large seamless virtual world which consists of some regions. At this mode, we can move to neighbor region and teleport to distant region.

B. Related Research

An evaluation of client viewer in Second Life is performed in reference[4]. It says that client has much load while Second Life is running. On the other hands, it also shows a result of simulation and numerical analysis as analysis on server side. In this research, we analyze server utilization on the basis of observation and measured results on server side while Metaverse is runnning.

III. Evaluation of Centrilized and Distributed Metaverse Server

Metaverse is a type of client-server system.

A. Experimental Environment of Centrilized Metaverse Server

Centrilized Server is a system in which servers who receive a lot of requests from clients are connected at one place. Users access to one server because data is gathered on it. Centrilized Server of OpenSim is able to construct with both standalone mode and grid mode. We have constructed a server with installing OpenSim. Clients access to the Server.

B. Experimental Environment of Distributed Metaverse Server

Distributed Server is a system in which information is connected among distributed servers. Web Server is typical example of distributed server. Distributed Server in Metaverse such as OpenSim is linked among servers. Client accesses somewhere in OpenSim Server. That is, the data client wants is somewhere in OpenSim Server. When Metaverse Server is constructed at grid mode using OpenSim0.6.8, it consists of four server modules, User Server, ROBUST Server, Messaging Server, and Region Server. These Server modules are able to run either on a single machine or multiple machines. User Server is a server which manages user(avatar). ROBUST Server is a server which manages objects, inventory of avatar, position of grid, and so on. Messaging Server is a server which handles message. Region Server is a server which manages region (meaning land, size of 256m x 256m).

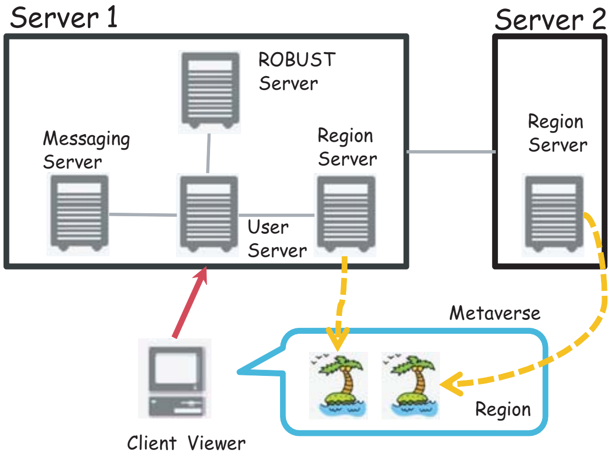


Figure 1. 2 Distributed Metaverse Servers

Figure 1 and 2 show experimental environment of this reserach work. Figure 1 uses two server machines, one machine is constructed with User Server, ROBUST Server, Messaging Server, and Region Server. The other is constructed with only Region Server which links to other server. Servers have spec as follows; CPU is Intel Xeon 3.6GHz and 2.4GHz, Main Memory is 4GB and 512MB, OS is Fedora 12(Linux2.6.31.5), and OpenSim0.6.8. On the other hand, Figure 2 shows the case in which each server module is set on five server machines.

We have used a relatively high spec client PC. The details are CPU is Intel Pentium 4 3.0GHz, Main Memory is 2.5GB, OS is Windows XP, and graphic is Intel 82945G Express Chipset Family.

IV. Login Time on Centrilized And Distributed Server

A. An Overview of Experiment

In this research work, we have measured user login time after constructing OpenSim Server with Centrilized and Distributed Server. Login time means from the time user pushes login button until OpenSim fully operates.

B. Result of a Measurement and Discussion

Figure 3 shows a result of a measurement. Login time of servers with different construction ways shows that it takes about 16 seconds at standalone mode, about 18 seconds at

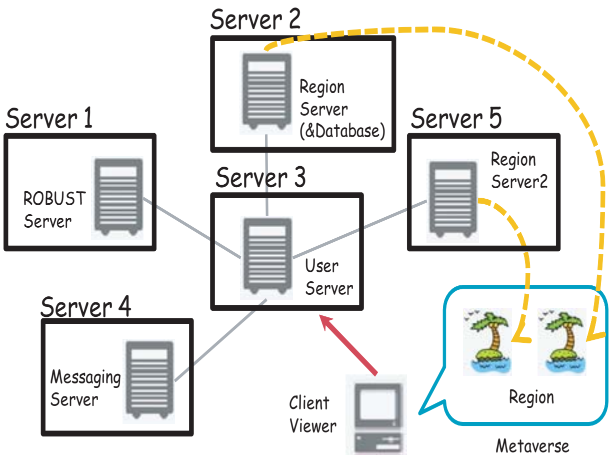


Figure 2. 5 Distributed Metaverse Servers

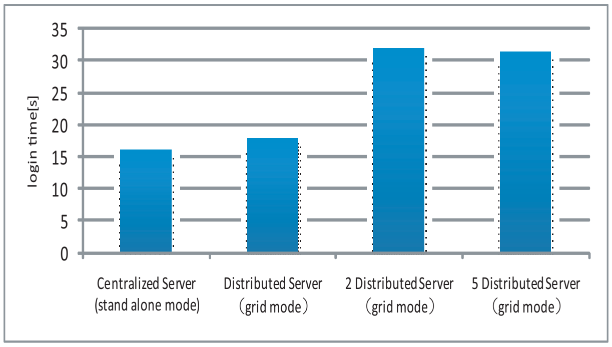


Figure 3. Login Time of Servers with Different Construction Ways

grid mode, about 32 seconds at two distributed servers, about 31 seconds at five distributed servers.

Comparing a case of standalone mode with that of grid mode, grid mode takes a little longer time to login than standalone mode. There is not much difference between the cases of two distributed servers and the case of five distributed servers. However comparing the former two(centrilized servers) with the latter two(distributed servers), they have a feature that the difference of login time is about double. These experimental results indicate that there is the difference of login time between centrilized server and distributed server. However it takes too much login time in each case comparing to the case of Web. In order to make Metaverse popular, the improvement of response time is necessary. Therefore, we have analyzed the response of Metaverse server in the rest of this paper.

V. Log Analysis at User Login on Distributed Server

A. An Overview of Experiment

In this papaer, we have analzed server log in distributed Metaverse Server when user login Metaverse, because we expect Metaverse becomes popular like Web system. That is to say, we have profiled server at user login after constructing distributed server by using OpenSim Server with grid mode. The details of the experiment is, when users login OpenSim, processing time of each server is calculated after getting logs from each server. Next, execution of all servers is sorted by time to analyze what kind of execution and how long it takes by color cording.

B. Process of Each Server Module

Figure 4 shows a process for accessing to server in five distributed servers(grid mode). This analysis is led by color cording of server logs after getting the logs from each server and sorting the log by time as shown in Figure 5, in reference to [8].

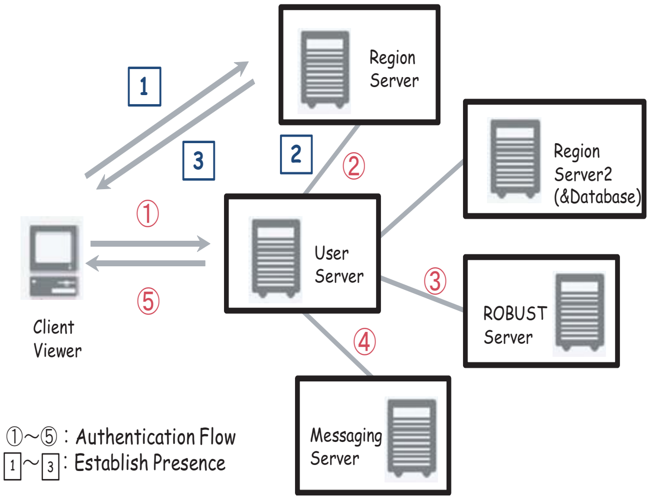
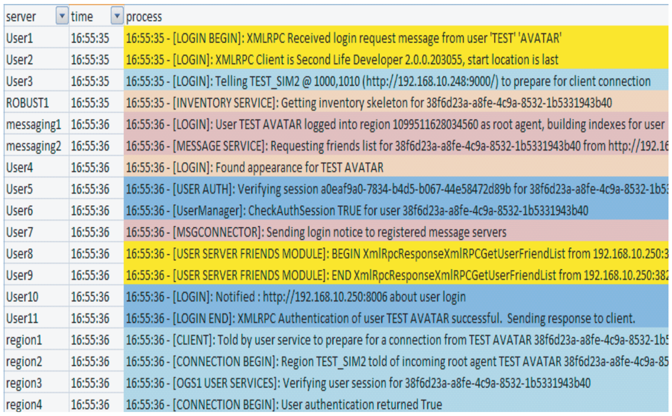


Figure 4. Login Process in Distributed Servers

First, OpenSim Server executes the following procedure to authenticate an avatar.

1) Access from client to User Server (sending user name and password).
2) Access from User Server to Region Server (sending user ID to connect)
3) Access from User Server to ROBUST Server (request appearance and inventry information).
4) Access from User Server to Messaging Server (telling user login).
5) Access from User Server to Client (affirimation of session again)



server	time	process
User1	16:55:35	16:55:35 - [LOGIN BEGIN]: XMLRPC Received login request message from user 'TEST' 'AVATAR'
User2	16:55:35	16:55:35 - [LOGIN]: XMLRPC Client is Second Life Developer 2.0.0.203055, start location is last
User3	16:55:35	16:55:35 - [LOGIN]: Telling TEST_SIM2 @ 1000,1010 (http://192.168.10.248:9000/) to prepare for client connection
ROBUST1	16:55:35	16:55:35 - [INVENTORY SERVICE]: Getting inventory skeleton for 38f6d23a-a8fe-4c9a-8532-1b5331943b40
messaging1	16:55:36	16:55:36 - [LOGIN]: User TEST AVATAR logged into region 1099511628034560 as root agent, building indexes for user
messaging2	16:55:36	16:55:36 - [MESSAGE SERVICE]: Requesting friends list for 38f6d23a-a8fe-4c9a-8532-1b5331943b40 from http://192.16
User4	16:55:36	16:55:36 - [LOGIN]: Found appearance for TEST AVATAR
User5	16:55:36	16:55:36 - [USER AUTH]: Verifying session a0eaf9a0-7834-b4d5-b067-44e58472d89b for 38f6d23a-a8fe-4c9a-8532-1b5
User6	16:55:36	16:55:36 - [UserManager]: CheckAuthSession TRUE for user 38f6d23a-a8fe-4c9a-8532-1b5331943b40
User7	16:55:36	16:55:36 - [MSGCONNECTOR]: Sending login notice to registered message servers
User8	16:55:36	16:55:36 - [USER SERVER FRIENDS MODULE]: BEGIN XmlRpcResponseXmlRPCGetUserFriendList from 192.168.10.250:3
User9	16:55:36	16:55:36 - [USER SERVER FRIENDS MODULE]: END XmlRpcResponseXmlRPCGetUserFriendList from 192.168.10.250:382
User10	16:55:36	16:55:36 - [LOGIN]: Notified : http://192.168.10.250:8006 about user login
User11	16:55:36	16:55:36 - [LOGIN END]: XMLRPC Authentication of user TEST AVATAR successful. Sending response to client.
region1	16:55:36	16:55:36 - [CLIENT]: Told by user service to prepare for a connection from TEST AVATAR 38f6d23a-a8fe-4c9a-8532-1b5
region2	16:55:36	16:55:36 - [CONNECTION BEGIN]: Region TEST_SIM2 told of incoming root agent TEST AVATAR 38f6d23a-a8fe-4c9a-85
region3	16:55:36	16:55:36 - [OGS1 USER SERVICES]: Verifying user session for 38f6d23a-a8fe-4c9a-8532-1b5331943b40
region4	16:55:36	16:55:36 - [CONNECTION BEGIN]: User authentication returned True

Figure 5. Log of All Servers at Grid Mode

With this procedure, client is authenticated and login process finishes. 22 steps are executed at this time, and it takes about 7 seconds.

Next, initialization of avatar is executed.

1) Access from client to Region Server (addind avatar to Region Server).
2) Access from Region Server to User Server (Getting friend list).
3) Access to Region Server to Client (update of avatar appearance)

28 steps are executed at this time, and it takes about 23 seconds.

Dividing these measurements into each server, execution time of each server is shown in Figure 6 and 7.

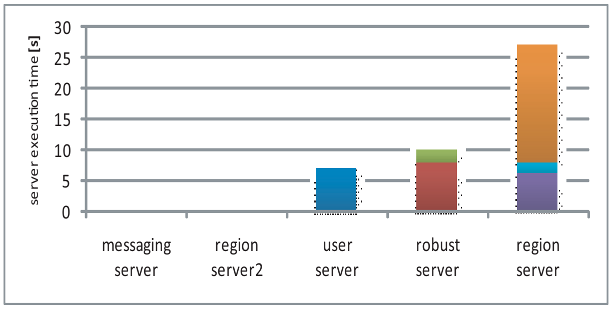


Figure 6. Execution Time of Each Server Module at Login in 2 Distributed Metaverse Servers

.

C. Result of a Measurement and Discussion

From Figure 5, 6 and 7, the difference of login time and server process are little between the case of 2 distributed servers and 5 distributed servers.

In order to reveal specific processes, we have analyzed the log, nd found that most steps take under a second and some steps take a few seconds. The step which takes long time is as follows: in specific, it takes about 6 seconds to add new queue to Region and about 19 seconds to update user appearance on Region Server. It takes about 8 seconds to get user inventory on ROBUST Server and about 7 seconds to send response for authentication to Client. That is to say, it is possible to shorten login time drastically by improving these steps performance.

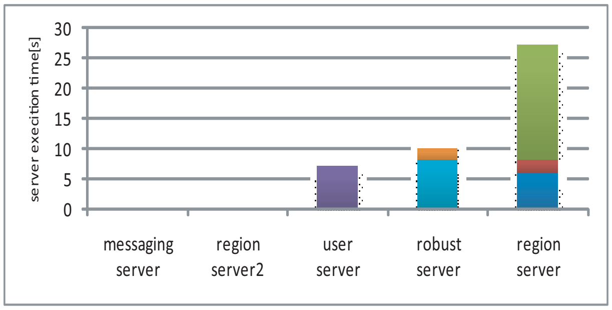


Figure 7. Execution Time of Each Server Module at Login in 5 Distributed Metaverse Servers

VI. Evaluation of User Login on Widely Distributed Server

A. An Overview of experiment

In general, distributed servers communicates in a widely distributed environment. Therefore, in this paper, Metaverse Server is not only distributed to each server module but also it is measured in a widely-distributed environment. Figure 8 shows our experiment in a widely-distributed environment. It is possible to construct by inserting a dummunet between client or one of servers, and one way delay time changes from 0ms to 100ms.

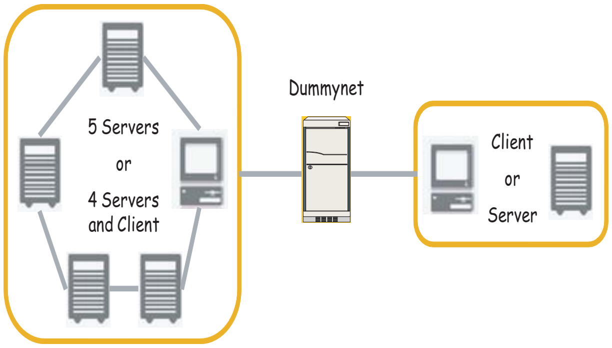


Figure 8. Widely-Distributed Environment

B. Result of a Measurement and Discussion

Figure 9 shows the result of a measurement. Figure 9 indicates that it shows little change in widely-distributed environment for client and Region Server. Howevwe User Server shows a difference about 10 seconds comparing the case of 0ms to 100ms delay. As shown by the Figure 4, it is possible to consider the reason is that User Server executes much processes.

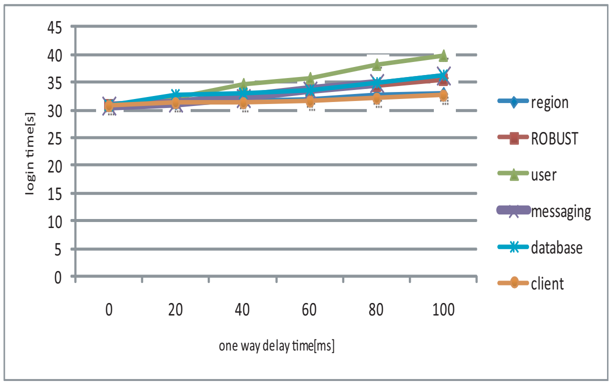


Figure 9. Login Time in Widely-Distributed Environment in Distributed Metaverse Server

VII. Evaluation of Teleport Time in a Widely-Distributed Environment with Distributed Metaverse Server

When Metaverse is distributed like web, a large Metaverse space should be constructed in the world. In such a case, it should happen that teleport will be increased among physically distributed servers. Therefore we have evaluated the teleport time on distributed Metaverse Server.

A. An Overview of Experiment

As shown by the Figure 8, dummynet is inserted between teleported region. One way delay time is changed from 0ms to 100ms, and teleport time is measured. In this mesurement, we prepared three scenarios. Teleport1 is a teleport to neighbor region in Metaverse. Teleport2 is a teleport to three distant region in Metaverse. Teleport3 is a teleport to 10 distant region in Metaverse.

B. Result of a Measurement and Discussion

Figure 10 shows the result of measurement.

Figure 10 shows that it has a difference of teleport time according to teleported region by comparing teleport1 with tereport2 and teleport3. However, by comparing teleport2 with teleport3, it does not take longer teleport time when it teleports to distant region in Metaverse . Moreover, it is clarified that teleport time is affected by physical position of server machine.

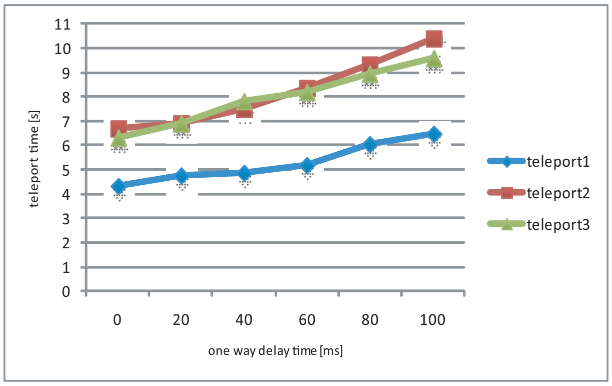


Figure 10. Teleport Time in a Widely-Distributed Environment with Distributed Metaverse Server

VIII. Conclusion and Future Works

In this paper, we have constructed centrilized server and distributed server using OpenSim, measured user login time to login Metaverse Service and profiled log of Metaverse Server. Moreover, we have measured login time and teleport time in a widely-distributed environment.

From these results, distributed server discussed in this paper takes about double login time than centrilized server. Therefore, the construction way of distributed server is changed in the experiment, and improvement of server procedure is verified by profiling server logs. After that, by measuring login time and teleport time in a widely-distributed environment, several problems are revealed. For example, we must consider the problem of position of User Server, the problem of a region position at teleport in Metaverse, and so on. In the future, to improve each server process, we will construct more larger virtual world by increasing region, and make an experiment on it.

References

[1] SecondLife, http://jp.secondlife.com/

[2] Ameba Pigg, http://pigg.ameba.jp/

[3] IMVU, http://www.imvu.com/

[4] Moshi Monsters, http://www.moshimonsters.com/

[5] Meet-Me, http://www.meet-me.jp/

[6] OpenSimulator, http://opensimulator.org/

[7] Sanjeer Kumar et.al.:"Second Life and the New Generation of Virtual Worlds",IEEE Computer,vol.41,No.9, pp.46-53 Sep. 2008.

[8] http://wiki.secondlife.com/wiki/Currentloginprotocols