

A Policy on Efficient Load-Balancing Using Contents-Based Game Servers

Won-Shig Myung

Department of Computer Engineering
Dongguk University, Seoul, Korea

ABSTRACT

This paper proposes a policy on efficient load balancing that can reduce the network game server load burdened by surging number of users. The study adopted a contents-based clustering technique. Recently, skyrocketing number of clients in on-line games causes overloads on specific game servers, and the consequent server-instability brings the worst situation: a server failure. To cope with this problem, one can install more high-powered servers or be equipped with back-up servers, which is often inefficient in terms of cost performance. To solve this problem, the present study examined the technology enhancing the performance and efficiency of game servers by reducing the loads of specific game servers. In doing this, this study used the clustering technology to compose game servers classified by their contents and carried out appropriate load balancing to numerous clients with load balancers in each region.

Keywords: *game servers, load balancing, RR method and contents, overloads.*

1. INTRODUCTION

In on-line games, gamers play, not with a computer, but with other humans with the computer as a medium. So, the games are not conducted in a fixed pattern. Rather, each person has unique features in playing their games. The concept of games has undergone a major transition: what people played together face to face has been changed into something they play together geographically apart from each other using terminals connected to a network.

On-line games are a pastime that multiple players simultaneously connected to the Internet enjoy in an interactive way using computers. Unlike earlier games conducted based on a predetermined scenario, these are the wide variety of games that assume hypothetical conditions and exchange data for a specific period of time via the terminals connected with others on a network [1].

The on-line games in the past were played by only two persons exchanging data based on one-on-one connections, whereas recent ones (e.g. MMORPG: Massively Multiplayer Online Role-playing Game) enable tens of thousands of people to be connected simultaneously. Specifically, Korea has established an excellent network infrastructure that can't be found anywhere in the world. Almost every household has a high-speed Internet access. What made this possible was, in part, high density of population that has accelerated the formation of good Internet infrastructure.

However, this rapid increase in the use of on-line games may lead to surging traffics exceeding the limited Internet communication capacity so that the connection to the games is unstable or the server fails. Expanding the servers though this measure is very costly could a solution for a problem.

To deal with this problem, the present study proposes the load balancing technology that: connects in the form of local clustering the game servers divided by their contents used in each on-line game; reduces the loads of specific servers using the load balancer, and; enhances performance of server for their efficient operation.

2. RELATED WORKS

On-line games are a real-time virtual game which multiple persons play exchanging their data between their terminals based on a network. This section explores the basic structures of on-line game systems as well as the load balancing in the previous on-line games.

2.1 Basic structure of on-line game systems

The game servers are favored in on-line games mostly because they can deal with more gamers in a given period of time. Contemporary on-line games are able to accommodate tremendous number of gamers so as to be called as a society. Single server or peer-to-peer networks [2] can never accommodate tens of thousand people connected at the same time. The capability to distribute and deal with gamers

* Corresponding author. E-mail : wsmyoung@dgu.edu
Manuscript received Jan. 25, 2007 ; accepted Mar. 12, 2007

is required to the servers. (Fig. 1) illustrates the basic structure of on-line game systems.

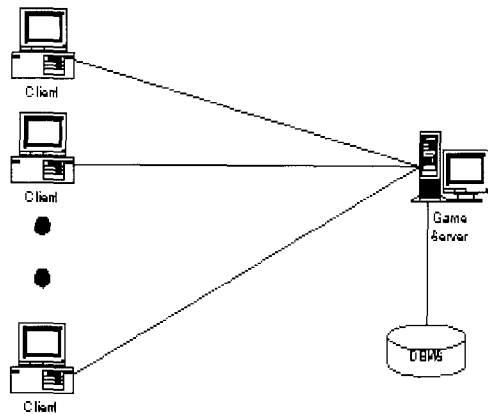


Fig. 1. The basic structure of on-line games.

On-line games consist of two parts: client and server. The gamers connect to the remote game server via the Internet by activating the client program, and then take part in the real-time games, in which the amount of data transmitted to the network or to be processed by the server is characteristically determined according to the number of users in any given point of time. Therefore, as the number of users increases, the load of the network and server rises exponentially resulting in the limitation in the number of users that the server is able to accommodate [3]. To cope with this problem, gamers should be divided and managed through game server expansion.

The structure proposed in (Fig. 1, 2) can't deal with the exponential increase in the number of clients because all the clients are connected to one server in this structure. In this situation, even server failure may occur. Thus, general extension techniques are used to solve this problem as illustrated in (Fig. 3).

The following two processes can be employed for the cases in (Fig. 3). First, both the contents and environments of the game server adopt the same process. This mode, which simply divides the connections according to the regions, gives the server flexibility concerning the number of client connections to it, while having the problem that all the servers should be reconstructed if game environments change. Second, servers are divided into a number of different cyber spaces and several servers are made to operate each space independently. This process causes the problem that the character a gamer constructed in a certain server can't be moved across freely from one server to another with different backgrounds. So, the problem is that they should make a new character for each game server. In addition, because it is difficult to balance clients when gamers are concentrated in one popular server, the performance expected with an increase in the number of servers is hard to achieve [4]. The mode as illustrated in (Fig. 3) has an advantage that the loads added to all the game servers are almost evenly balanced. After all, the average loads in each server decrease so that it is able to accommodate as many users as the number of servers that have become available. However, the problem in this mode is its potential failure in synchronization between game servers, which is important because gamers should be

provided with the game environments with identical map and game data regardless of which server he/she is connected to.

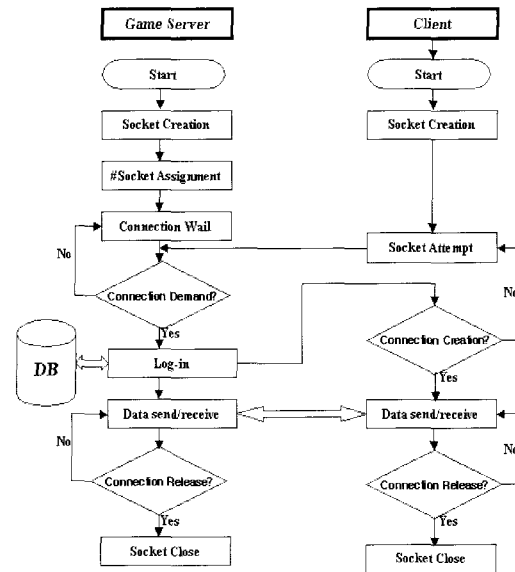


Fig. 2. On-Line game system basis operation.

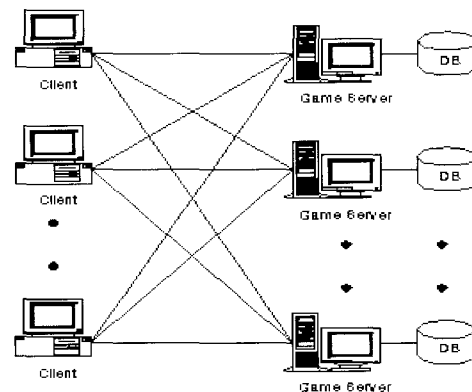


Fig. 3. General expansion structure in on-line game systems.

2.2 Load Balancing in on-line games

The process of load balancing in the previous on-line games is to have just one server to operate initially when not many users are connected, and then to combine the servers for the same game into an internal network to play the same roles as the user connection increases. This is especially convenient because the capacity of the servers is augmented. Load balancing is a technology that appropriately distributes loads to multiple servers based on the connections each server has. If excessive loads should be concentrated in one specific server [1].

In load balancing, gamers who completed the login process by punching their ID and password are distributed evenly to each server so that no individual server becomes overwhelmed by too much traffic. In this process, the user never knows which server he/she will be connected to. (Fig. 4) illustrates the

structure of previous load balancing processes, in which a communication server either is present in the game server or can be generated independently to be used. The role of a communication server is, by connecting to the previous client, to protect the game server so it is able to focus only on the process of the games to reduce its loads. DB server may be also called as login server, or two separate servers may play these two roles. DB server stores the information of a gamer (ID, password, etc.), as well as halted game situations. Therefore, this server plays a role in having the gamers to be logged in by confirming their ID/password when they first connect to the server. Having all the information about the gamer as well as about the game characters, it shows and processes the same information even when the gamer is connected to a different IP or uses another computer.

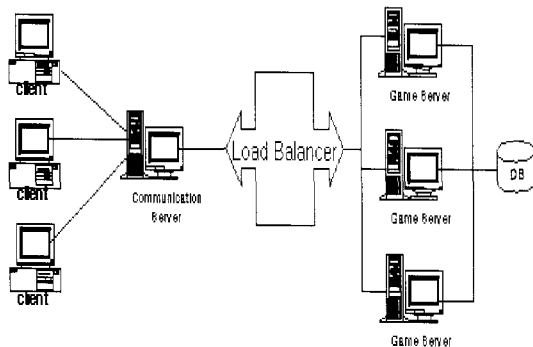


Fig. 4. Previous load balancing processes.

On-line games are often designed with a clustered structure to prepare for a potential expansion in the future. (Fig. 5) schematizes the process of load balancing in a clustered structure. In this process, each server has a separate topographical background and a shared time background.

To prevent game characters to be concentrated into any one server, identical spatial backgrounds are implemented in multiple servers and they are connected using the clustering technique to distribute traffics by load balancing.

However, the structures in (Fig. 4 or 5) may give so much tasks to a load balancer server that the traffic in itself can increase, causing delayed connection for a client. To cope with this problem, the present study proposes an efficient load balancing technique to reduce traffic in load balancer servers and to minimize the time spent to initiate a connection.

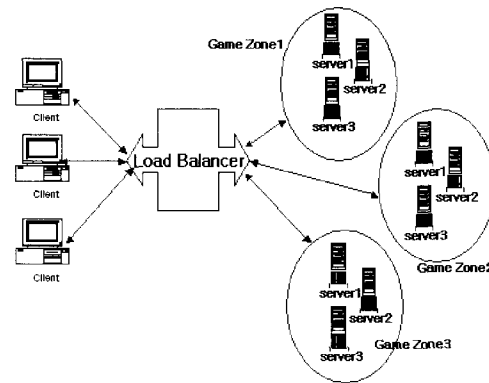


Fig. 5. Load Balancing-using clustering.

3. THE LOAD BALANCING TECHNIQUE PROPOSED

(Fig. 6) describes an efficient load balancing technique proposed in this paper.

The current on-line games adopt the game-server load balancing in a clustered structure to cope with an explosive increase in the number of clients. However, there are many problems in managing all the game servers and evenly distributing the loads with one central load balancer server. In this context, this study shows that combining multiple servers with topographical backgrounds into one cluster and then having a load balancer server together can lighten the burden of load balancers and distribute gamers more evenly to the servers than previous processes could do.

The structure of the load balancing system proposed for on-line games should support the following seven functions in general [4]:

- ① Log-in function for the connections to the client as well as for user identification and authentication
- ② Database management function that deals with personal information of the gamer and the features of his/her characters
- ③ Communication function for transmitting and receiving packets with clients
- ④ Download function that activates automatic transmission of client programs or changed data
- ⑤ Expansion function that supports map expansions and allows the movements of characters between maps
- ⑥ Game Processing function that interprets the orders of gamers according to the game rules and transmits them to the client
- ⑦ Load balancing function evenly distributing traffic lest clients should be concentrated into any one server

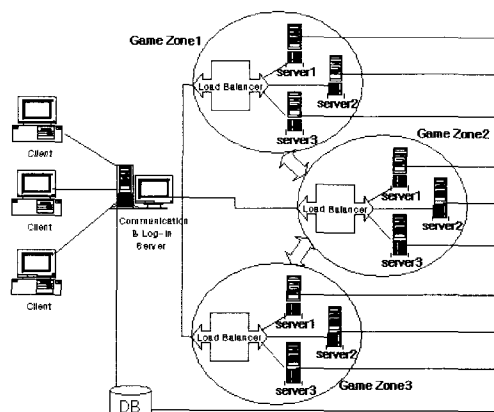


Fig. 6. The Efficient load balancing technique proposed.

The load balancing system proposed in this study has four independent servers that share these seven functions. In other words, this system combined the following four servers: 1) the DB server is responsible for integrative database management; 2) the game servers are in charge of processing the games only; 3) communication--login server is responsible for login and identification--authentication of clients as well as for downloads. They connect clients into a desired game server, and also transmit or receive packets; 4) load balancer server evenly allots clients to the game servers composed based on clustering when clients attempt to connect to a desired game server. The proposed load balancing system monitors the information about the game servers it manages.

The proposed load balancing system is activated in the following steps(Fig. 7):

- ① A gamer activates a client program in his/her terminal to play games and gain access to the login/communication server.
- ② The login/communication (LC) server connects to the DB server to get authentication for the connected client and to have automatic downloads.
- ③ After step 2 is completed, the LC server transmit the information about the user obtained from the DB server to the client.
- ④ After step 3, the LC server connects to the load balancer server in the game zone the client wants.
- ⑤ The load balancer server connects the client to the least loaded one among all the game servers it manages.
- ⑥ When the game is discontinued, the game server stores on the DB server all the information about the character designated by the client.
- ⑦ When the game background is to be changed, the LC server connects the client to the desired game zone after receiving the DB information offered by the game server, and then go back to step 5.

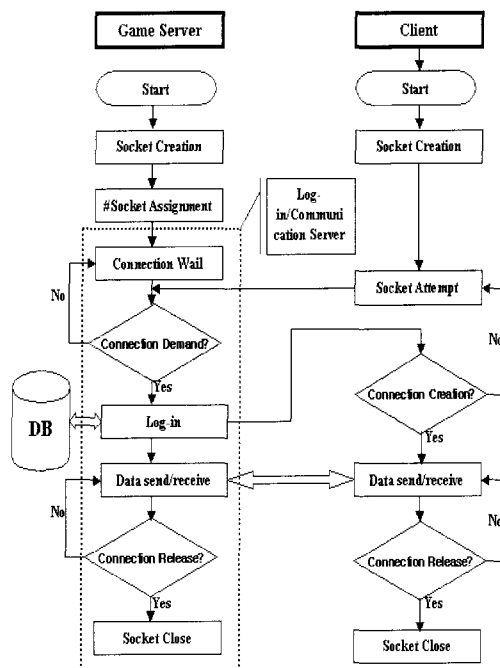


Fig. 7. Proposed new game server operation.

4. CONCLUSIONS

The structures of various servers introduced in sections 2 and 3 are compared with one another in the following table [5].

This paper proposes a policy on efficient load balancing that can reduce the network game server load burdened by surging number of users, using a contents-based clustering technique. The previous systems have one load balancer server to handle all the functions, whereas the system proposed in the study set up game servers with regionally identical contents in a clustering structure and has a load balancer server in each clustered region to efficiently distribute gamers.

The efficient load balancing system proposed is advantageous in that its combined login/communication server can shorten the potential delay in connection compared with the separate management of two independent servers; that its low cost for initial implementation facilitates easy introduction, and; that it can accommodate large-scale users.

The shortage of this system is that the entire service can be paralyzed if the load balancing system fails. So, the question for follow-up studies is to consider ways to add a complementary function that converts to a back-up system when the load balancing system is down.

Few Korean products are No.1 in the world market. However, the computer game area has a potential to become world best among all the areas in service or culture industry. Specifically, Korea already possesses world-class technologies in on-line games. It is not too much to say that how many simultaneous user can be efficiently accommodated is the critical issue in the area of on-line games.

Table 1. Merits and Demerits of Structures of Various Servers

<i>Types</i>	<i>Merits</i>	<i>Demerits</i>
• <i>basic structure of current on-line systems</i>	<ul style="list-style-type: none"> • easy implementation • operation of servers 	<ul style="list-style-type: none"> • inappropriate for expansion • unable to cope with influx of users
• <i>General expansion structure</i>	<ul style="list-style-type: none"> • overall balanced loads 	<ul style="list-style-type: none"> • unable to communicate between servers
• <i>proposed game server structure</i>	<ul style="list-style-type: none"> • balanced loads • movable to other servers • enhanced cost performance 	<ul style="list-style-type: none"> • unable to serve as a back-up in case the load balancing system fails

REFERENCE

[1] Industry technical cooperation center, "Industry research on actual state research of game industry laying stress on technological level", Dongguk Univ., 2002.2

[2] Computer Game Conference, 1999

[3] Nam Jae Lee, Duck Woo Seo, Hoon Sung-Kwak, "Distributed Server Configuration Scheme for Internet Omline Game" **In Proc. of the 2th Korea Game Society**, pp. 31-36, 2001

[4] Wen tong Cai, Percival Xavier, Stephen J. Turner, Bu-Sung Games on the Internet, 16th Workshop on Parallel and Distributed Simulation", p.60, May 12-15, 2002

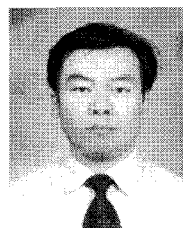
[5] Edward Chang, Hector Garcia-Molina, "Reducing Initial Latency in Media Servers", **IEEE Multimedia** Vol. 4, No. 3, pp. 50-61, July-September 1997

[6] M. Deloura, "Game Programming Gems 1", Charles River Media, 1999 [7] A. Lamothe, "Tricks of the Windows Game Programming Gurus", Macmillian USA, 2000

[7] W. R. Stevens, UNIX Network Programming 2nd Edition, Prentice Hall, 1998

[8] H. I. Bozma, J.S. Duncan, "A Game-Theoretic Approach to Integration of Modules", **IEEE TRANSACTION ON Pattern Analysis And Machine Intelligence**, Vol. 16, No. 11, pp. 1074-1086, November 1994

[9] Christoforos E. Kozyrakis, David A. Patterson, "A New Direction for Computer Architecture Research", **IEEE COMPUTER**, Vol. 31, No. 11, pp. 24-32, November 1998



Won-Shig Myung

He received the B.S., M.S in computer science from Anyang university, Korea in 1996, 1998 respectively and also received M.S., Ph.D. in computer engineering from Dongguk university, Korea in 2004. His main research interests include parallel computers, game server, web server clustering, load balancing systems, data encryption standard.