International Journal of u- and e- Service, Science and Technology
Vol.10, No.6 (2017), pp.93-102
http://dx.doi.org/10.14257/ijunesst.2017.10.6.09

Design and Implementation of Smart Phone Games for Linux Clustering

Jun-Ho Huh*, Seok-Hwa Jung**, Kyungryong Seo**[1]

Department of Software, Catholic University of Pusan, Busan, Republic of Korea
*Department of Computer Engineering, Pukyong National University at Daeyeon, Busan, Republic of Korea***
72networks@cup.ac.kr, tsw3186@naver.com, krseo@pknu.ac.kr

Abstract

Following the increase in the popularity of smart phone games, the number of participating gamers is also rapidly increasing imposing a heavy traffic on a game server(s). This slows down the game speed and sometimes leads to interruption of game proceedings. Accordingly, the game service providers are adding more servers for their games to allow better performance but increasing the number of servers means more expenditure for the service providers. Meanwhile, the microprocessors often used for the games have become faster and cheaper and the studies on the cluster systems which enable more efficient game performance are being carried out widely by the game companies instead of just adding more servers. The problem in using a cluster system is that the performance efficiency and the effects of the utilized cluster system can only be guaranteed through a program that has used a specific parallel programing method. Without this programing technique, the benefits of using cluster systems in games are minimal. An Eco-Server that fully allows a large number of gamer accesses has been implemented in this study to surmount the problem. For the experimental model, a new platform technology for the smart phone game which uses a Linux-based network server has been designed and implemented.

***Keywords**: Games, Smart Phone Games, Linux Network Server, Linux, Clustering*

1. Introduction

The network technology, which is the fundamental element technology in the on-line gaming industry, is being studied widely. Currently, the on-line game industry is growing so fast that it is expected that this industry will be the focus of future digital economy. However, there are many technical problems to be solved in on-line games and the priority is to identify the flaws in the past gaming processes and find the solutions to improve their performances. Most of the problems in the on-line games are related to the quality of the network performance, the server, or both. Therefore, constructing a systematically robust server and efficient network are required prior to creating some high-quality/performance games. The on-line game server should be able to handle huge number of access attempts by the gamers and at the same time, the gaming network needs to provide a flawless and fast gaming actions to them. The cluster systems have been considered and studied widely as an alternative solution of maintaining the costs of running the games low without increasing the number of game servers. Together with continuous price decrease of microprocessors and performance-enhanced networks, the cluster system can be a central technology for the on-line games. However, game developers are required to have in-depth knowledge and skill sets to perform parallel programming when constructing a cluster system for the on-line games. Thus, in order to

[1] Corresponding Author

create the games that can handle a great number of multiple access, a Linux-based network server (Eco-server) has been designed and implemented for a smart phone game as an element of a platform technology.

2. Related Research

Currently, the most generally used method is the 2-tier method which maintains one-to-one relationship between a client and a server. This method is also called a centralized structure or a single server structure. Also, there is another form called a distributed server structure where neither any particular server nor client that assumes a specific role exists, like in LAN. Here, the server can be client and vice versa [1-6].

Different from the centralized sever structure, the distributed server structure is needed when there are more than one server. The systems that are operated with just a single server have reached their limits as the number of network game users increased rapidly. The distributed server system usually has more than one server and the servers are divided into two types: one(s) that can share data and one(s) that cannot. The former is called replicated server and the latter, non-replicated server [5-10].

The replicated server structure is one that lets the client to increases the number of users who can access to the server by connecting more than two systems that have the same server structures. The disadvantage of this structure is that the client connected to each different server will not be able to share data with other client(s) of other server. The non-replicated server structure can share data through the database between the servers and the connection can be switched to other server. However, this could result in clock synchronization, deadlock between servers, and integrity and simultaneity of game data. Again, this structure can share data with other server(s) but sometimes the clients can be concentrated on a certain server so that the loads will not be distributed evenly [11-20].

The reason for selecting Linux for the system is that it is possible to install the operation system kernels with low cost, re-organization of all elements is easy, works fine on a low-performance cheap hardware platform, and strong enough to completely utilize most of the hardware elements. Also, other advantages are: Linux abides by standards for the quality of source code, its kernels are small-sized, it is compatible with other operation systems, and updating and patching are easier compared to other operation systems. The volume of internet traffic has significantly increased following the rapid increase in the number of smartphone-based games [21]. Additionally, due to the wide spread of smartphones around the world, social network games and cloud computing have developed greatly. Users are now can store data or software’s in a highly reliable server(s) and use them on their smartphones or PCs through network-based computing [22-23]. Accordingly, the traffic volume centering around the server storages has increased much more than the past so that the Solid State Drives (SSD) which can perform I/O processes efficiently are becoming popular. Although SSDs have the merits of high-performance, low-power consumption, low-noise, low-heating and other properties, their major demerit is the price, which is often higher than the hard disks. Considering these properties, SSDs are used as a Cache device in the large-capacity main storages [24-27].

When adopting a SSD as a Cache device, 'writ-back' or 'write-through' type writing policy can be used. The 'write-through' method records the data on both the SSD Cache and the main storage while the 'writ-back' method records the data on the SSD Cache first and normally inputs the same data into the main storage later when the system gets a spare time in processing. As a result, the possibility of improvement in the writing performance in the 'write-through' method is rather small whereas much performance enhancement can be expected for the 'write-back' method. However, once the 'write-back' method has been used, the data will be lost when SSD malfunctions [24]. Various studies that would prevent data loss by improving SSD's performance by using it as a 'write-back' Cache have been proposed [5-7]. One of those ideas is that by configuring a multiple number of SSDs as a RAID form to use them as a 'write-back' Cache [27]. Several physical disks are connected as a stripe form in this RAID technology and the data including the additional information which can be used to recover the lost data will be stored so that the lost data can be recovered with the help of this information even if one disk fails in such a striped connection [28].

3. A Design of Smart Phone Games for Linux Clustering

As most of smart phone games are played bi-directionally, a server is required to assume an intermediary role. In this study, the game was played between the two clients so that the design included a single server and two clients [Figure 1]. Since both the server and the clients perform the same hosting process in their gaming terminals (*i.e.*, PCs or other digital devices), they can communicate each other using SOCKET_STREAM (TCP protocol) on AF_UNIX (UNIX domain socket).

To implement a server that can accept simultaneous accesses, each client's access will be processed by each children server continuously generated with fork.

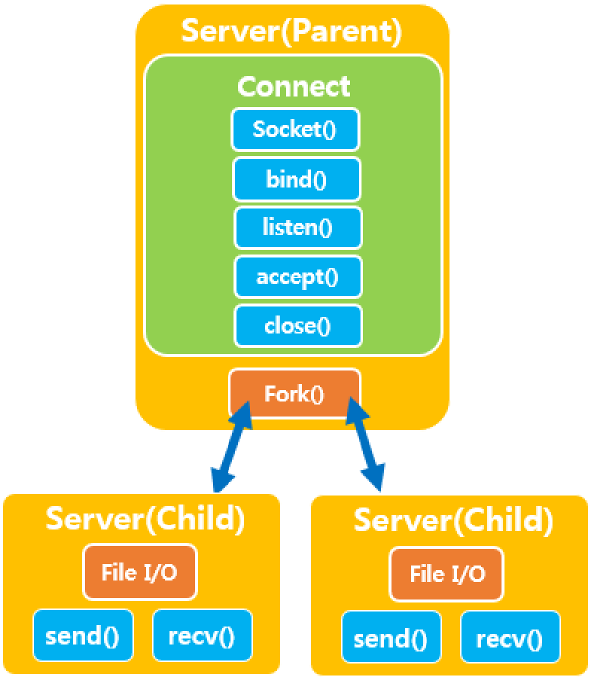

Figure 1.

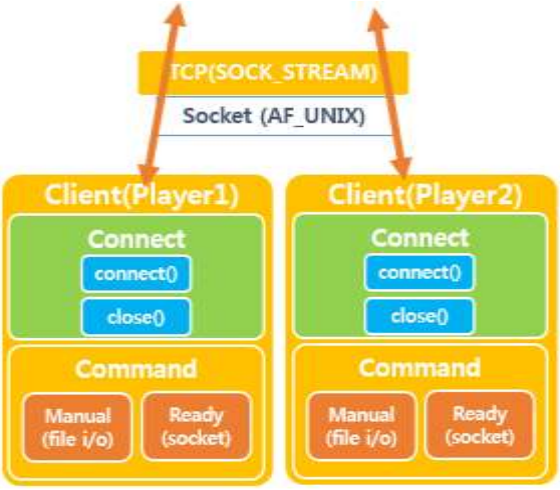

Figure 2. Whole Design Structure for the Linux Network Server [20]

As the rules of game, the server will call up the manual file to give it to the client and provide a service that shows the color of opponent's number tile when the game starts. As the clients have a mechanism of comparing each other's tiles during the game, shared memory was used between the clients. A semaphore was used to solve the problem of data corruptions or for the cases that each client cannot refer to the shared memory properly. In addition, synchronization process was adjusted by using SIGNAL because the first player has to wait for the data provided by the next player to perform calculations.

The message that includes the game results computed by the first player after comparing the values of number tiles when SIGURS1 (determined by using sigaction function) arrived was delivered through the shared memory to each player Figure 2.

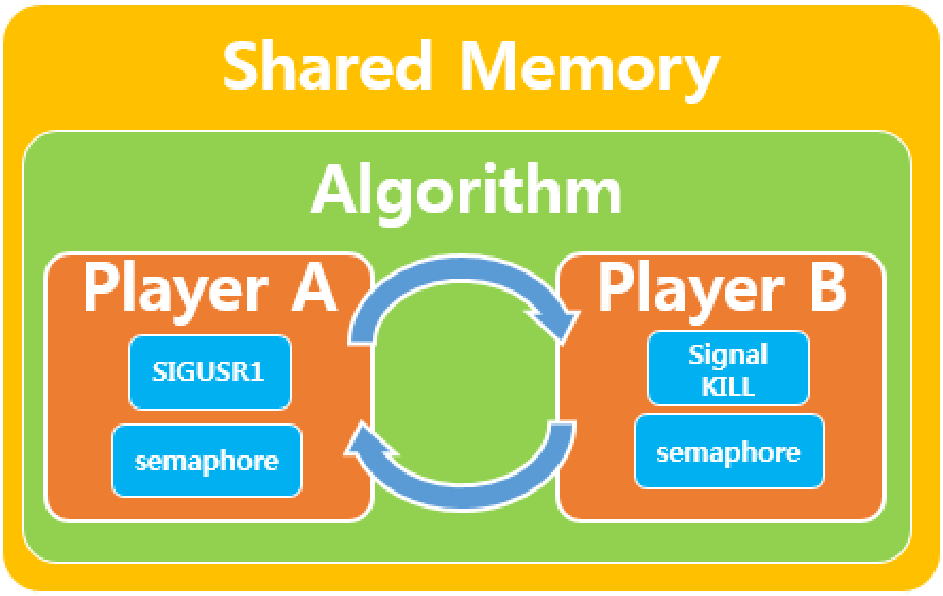

Figure 2. Detailed Presentation of Client-Shared Memory [20]

4. An Implementation of Smart Phone Games for Linux Clustering

Figure 3 shows the communication process in the Linux-based game implemented in this study. Since the connection between the server and the client should be established on a 1:n basis rather than on a 1:1 basis, the server used here is a multiserver where multiple child processes are generated by using a fork function to simultaneously link with multiple number of clients. These child processes serve each client individually.

After a server socket has been created and binding a socket file descriptor with an assigned ip address/port number with the bind function, the server waits for the client's access request through the listen function. The connection request through the connect function will be established by using an accept function.

Meanwhile, the client receives commands by using the gets function so that when the first command is delivered to the server, the game manual will be read with fopen and delivered to the client through the buffer using a send function. The client that reads the buffer through the recv function to output the manual on a display screen. The game will be initiated when the 2nd command is delivered. In this stage, the connection with opponent client is established and shared memory is created between the clients at the same time.

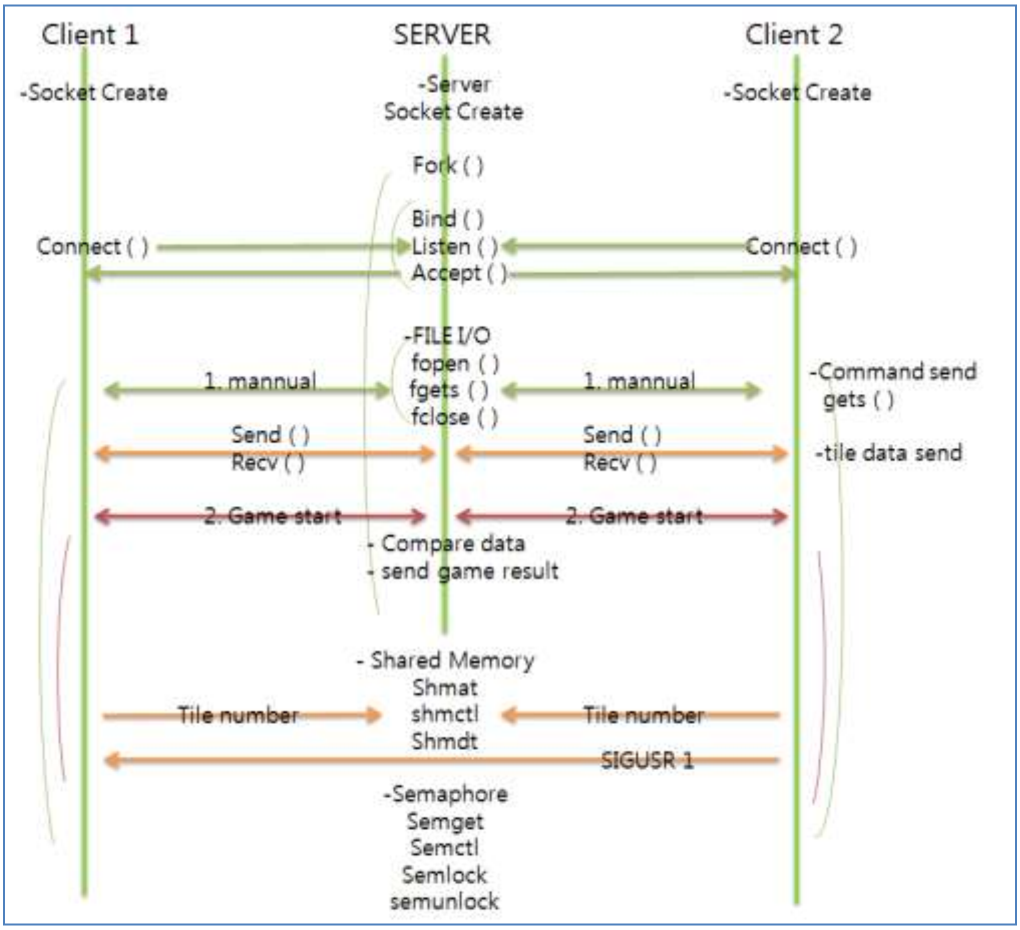

Figure 3. The Process of Communications in the Implemented Linux-based Game

The game's outcome (victory or defeat) is determined by the opponent's numerical data received through the shared memory and the order of presentation of number tiles by the clients controlled by through the shared memory as well.

Sometimes data can be corrupted when several processes attempt to use the memory simultaneously so that their sequence of operation must be determined. The most widely used method for this purpose is the one that uses Semaphore, which has been used to synchronize the clients' processes here. For example, 'Semlock' is used to block an opponent's attempt to use the shared memory when other (Client 1) is writing data to it. The shared memory will be available to the opponent through 'Semunlock' after completing the data input process.

Also, this program uses signals to control the order. The signal is the non-periodical signal delivered during the operation process when a certain event occurs. Here, to proceed with a game between two clients by using the user-defined signal SIGUSR1, the order of game is decided in a way that the Client 1 cannot send any additional data to the opponent Client 2 until he/she responds with his/her numerical data after Client 1's first transmission of data.

5. Conclusion and Future Work

Clients are connected to the Eco-server on an 1:1 basis and the problem here was to enable data sharing and comparison between children servers. This problem was solved by generating the shared memories between the server and the clients. Here, the data requires to be shared by the system elements (*i.e.*, server and clients) are regarded as the comparison results of number tiles of respective clients. The problem of synchronization within the gaming process was solved by using SIGNAL and Semaphore. The number of players can be increased by using the proposed design. As a result, following game [19] was implemented with Linux Figure 4, 5.

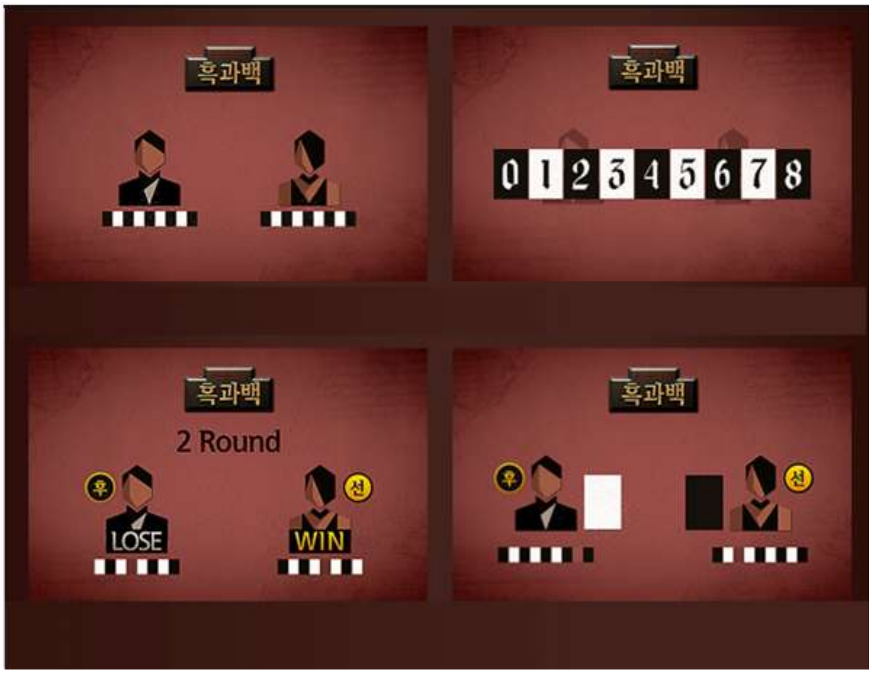

Figure4. Sample of Implemented Game (1).

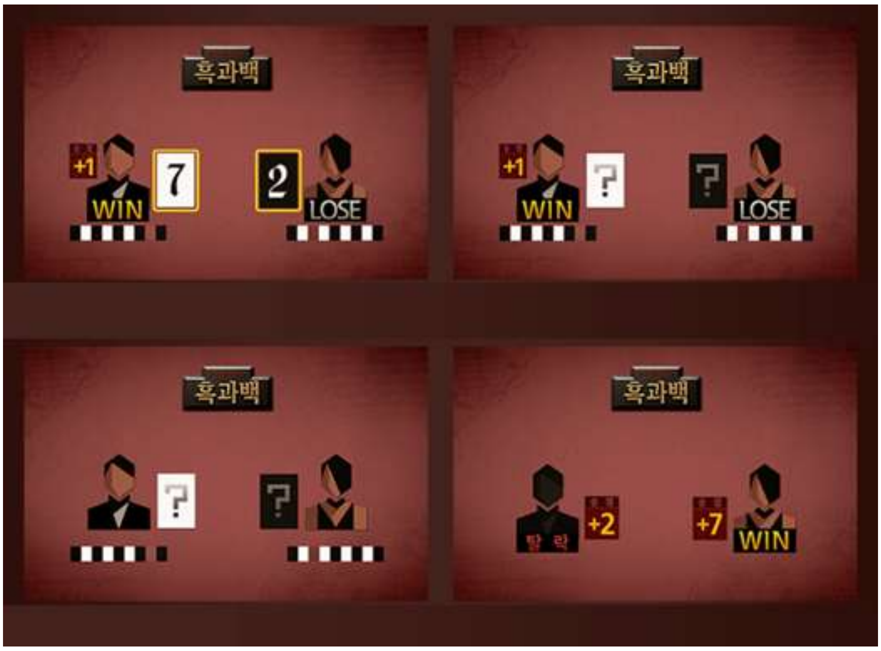

Figure 5. Sample of Implemented Game (2)

Since the majority of smart phone games are of 2-player games, the game design includes a single server assuming the role of an intermediary and two clients. This design allows easy increase in the number of players. As a future work, we plan to produce a demo-version based on this design and conduct thorough performance evaluation at the same time in the Patent.

Acknowledgements

The authors would like to thank Mr. S. J. Kim (M.S., Game Programmer, Nexon, Seoul, Republic of Korea) who has provided a valuable review on this paper.

The first draft of this part of paper [20] was presented Oral Session in The 10th 2016 International Interdisciplinary Workshop Series, JEJU, Aug 16-19 (2016). I am grateful to 2 anonymous commentators who have contributed to the enhancement of the paper's completeness with their valuable suggestions at the Workshop.

References

[1] J. H. Bae, “Design and Implementation of a Linux Web Cluster System for Load Balancer Game Server”, Journal of the Korean Society for Computer Game, vol. 1, no.1, **(2002)**, pp. 69-82.

[2] J. H. Bae, “A study of Inter-Process Communication based on Process Model for Linux Networks Game Server”, **(2007)**, pp. 5-12.

[3] J. H. Bae, “Design and Implementation of Linux Server Based Game Design Software”, no. 15, **(2008)**, pp. 55-61.

[4] J. H. Bae, “Design and development on cloud-based project management system”, Korean Society for Computer Game, vol. 26, no. 1, **(2013)**, pp. 61-67.

[5] D. Hwang, J. E. Jung., S. Park and H. T. Nguyen, “Social data visualization system for understanding diffusion patterns on Twitter: a case study on Korean enterprises”, Computing and Informatics, vol. 33, no. 3, **(2015)**, pp. 591-608.

[6] P. Maillé, B. Tuffin and J. M. Vigne, “Competition between Wireless Service Providers Sharing a Radio Resource”, Networking 2012, LNCS, Springer, Berlin Heidelberg, vol. 7290, **(2012)**, pp.355-365.

[7] G. Kesidis, I. Hamadeh, Y. Jin, S. Jiwasurat and M. Vojnović. “A model of the spread of randomly scanning Internet worms that saturate access links”, ACM Transactions on Modeling and Computer Simulation. vol. 18, no. 2, **(2008)**.

[8] J. H. Huh, T. Koh and K. Seo, “Design of a Shipboard Outside Communication Network and the Test Bed Using PLC: for the Workers' Safety Management During Ship-Building process”, Proceedings of

the 10th International Conference on Ubiquitous Information Management and Communication, ACM IMCOM 2016, no. 43, **(2016)**.

[9] E. Altman, M. K. Hanawal, R. ElAzouzi and S. Shamai, "Tradeoffs in green cellular networks", ACM Sigmetrics Performance Evaluation Review, vol. 39, no. 3, **(2011)**, pp.67-71.

[10] C. Hasany, E. Altmanz and J. M. Gorcey, "A Coalition Game Approach to the Association Problem of Mobiles in Broadcast Transmission", International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, Princeton, IEEE, 236-240, **(2011)**.

[11] S. Umamaheswari and G. Radhamani, "Enhanced ANTSEC Framework with Cluster based Cooperative Caching in Mobile Ad Hoc Networks", Journal of Communications and Networks, IEEE, vol. 17, no. 1, **(2015)**, pp. 40-46.

[12] J. H. Huh and K. Seo, "Design and test bed experiments of server operation system using virtualization technology", Human-centric Computing and Information Sciences, Springer, Berlin Heidelberg, vol. 6, no. 1, **(2016)**, pp.1-21.

[13] J. Matanza, S. Kiliccote, S. Alexandres and C. R. Morcillo, "Simulation of Low-Voltage Narrow-Band Power Line Communication Networks to Propagate OpenADR Signals, Journal of Communications and Networks, IEEE, Vol.17, No.6, pp. 656-664 (2015)

[14] J. H. Huh, H. B. Kim and K. Seo, "A Design of Smart-based Education Gamification Platform Using Mobile Devices", Advanced Science and Technology Letters, SERSC, vol. 127, **(2016)**, pp. 85-90.

[15] J. H. Huh, S. Otgonchimeg and K. Seo, "Advanced metering infrastructure design and test bed experiment using intelligent agents: focusing on the PLC network base technology for Smart Grid system", The Journal of Supercomputing, Springer, US vol. 72, no. 5, **(2016)**, pp. 1862-1877.

[16] J. H. Huh and K. Seo, "Development of Competency-oriented Social Multimedia Computer Network Curriculum", Journal of Multimedia and Information System, vol. 1, no. 2, **(2014)**, pp. 133-142.

[17] H. T. Nguyen, "Computing Semantic Similarity for Vietnamese Concepts Using Wikipedia", Advances in Intelligent Systems and Computing, Springer, Switzerland, vol. 341, **(2015)**, pp. 91-106.

[18] J. H. Huh, H. B. Kim and K. Seo, "A Design of Smart-based Education Gamification Platform Using Mobile Devices", Advanced Science and Technology Letters, vol. 127, **(2016)**, pp. 85-90.

[19] TvN.: The Genius: Rule Breaker, Game Images, **(2014)**.

[20] J. H. Huh, S. H. Jung and K. Seo, "Design and Implementation of Platform Technology of Smart Phone Games for Linux Network Server", Asia-pacific Proceedings of Applied Science and Engineering for Better Human Life, vol. 4, **(2016)**, pp. 155-158.

[21] Y. H. Cho and Y. H. Oh, "Survivability in Broadband Convergence Network(BcN) of Real-time online Game", Journal of Korean Society for Computer Game, vol. 24, no. 3, **(2011)**, pp. 93-99.

[22] J. H. Bae and M. S. Kang, "Developed of Cloud based Content system on Satellite", Journal of Korean Society for Computer Game, vol. 26, no. 1, **(2013)**, pp. 7-12.

[23] J. H. Bae, "Design and development on cloud-based project management system", Journal of Korean Society for Computer Game, vol. 26, no. 1, **(2013)**, pp. 61-67.

[24] S. Byan, J. Lentini, A. Madan, L. Pabon, M. Condict, J. Kimmel, S. Kleiman, C. Small, and M. Storer, "Mercury: Host-side Flash Caching for the Data Center", in Proceedings of the 28th IEEE Symposium on Mass Storage Systems and Technologies (MSST), **(2012)**.

[25] D. Qin, A. D. Brown and A. Goel, "Reliable Writeback for Client-side Flash Caches", in Proceedings of the 2014 USENIX Annual Technical Conference (ATC), **(2014)**.

[26] R. Koller, L. Marmol, R. Rangaswami, S. Sundararaman, N. Talagala and M. Zhao, "Write Policies for Host-side Flash Caches", in Proceedings of the 11th USENIX Conference on File and Storage Technologies (FAST), **(2013)**.

[27] Y. S. Oh, J. M. Choi, D. H. Lee and S. H. Noh, "Improving performance and lifetime of the SSD RAID-based host cache through a log-structured approach", in Proceedings of the 1st Workshop on Interactions of NVM/FLASH with Operating Systems and Workloads (INFLOW), **(2013)**.

[28] D. A. Patterson, P. Chen, G. A. Gibson and R. H. Katz, "Introduction to Redundant Arrays of Inexpensive Disks (RAID)", in Proceedings of the 34th IEEE Computer Society International Conference (COMPCON), **(1989)**.

[29] E. J. Lee and D. H. Lee, "Reliability and Performance Analysis of SSD cache in game servers", Korean Society for Computer Game, vol. 28, no. 1, **(2015)**, pp. 131-136.

Authors

Jun-Ho Huh, finished the Cooperative Marine Science and Engineering Program, Texas A&M University at Galveston, United States of America in Aug. 2006. Received B.S. in Science Degree from Department of Major of Applied Marine Sciences (Marine Aquaculture, Oceanography, Marine Life Sciences), B.S. in Engineering Degree (Double Major) from Department of Major of Computer Engineering from Jeju National University at Ara, Jeju, Republic of Korea in Aug. 2007. And completion of the Secondary School (Middle and High schools) Teacher Training Curriculum in accordance with Republic of Korea Secondary Education Act. M.A. in Education Degree from Department of Major of Computer Science Education, Graduate School of Education, Pukyoug National University at Daeyeon, Busan, Republic of Korea in Aug. 2012. And completion of the Secondary School (Middle and High schools) Teacher Training Curriculum in accordance with Republic of Korea Secondary Education Act. Received the Ph.D. Degree in from Department of Major of Computer Engineering, Graduate School, Pukyoug National University at Daeyeon, Busan, Republic of Korea in Fed. 2016. He received the Best Paper Award from Korea Multimedia Society five times (Nov. 2014, May. 2015, Nov. 2015, May. 2016, Oct. 2016). And he received Best Paper Award The 10th 2016 International Interdisciplinary Workshop Series from HSST (Aug. 2016). Also he received Best Paper Award The 16th International Conference on Control, Automation and Systems (Oct. 2016), ICROS with IEEE Xplore. Senior Research Engineer of SUNCOM Co., Busan, Republic of Korea (Aug. 2015- June. 2016). And Research Professor of Dankook University at Jukjeon, Republic of Korea (July. 2016-Sep. 2016). Also Clinical Assistant Professor of E-Green Remote Continuing Education, Seoul, Republic of Korea. (Mar. 2016- Nov. 2016). Currently he is Assistant Professor of Department of Software, Catholic University of Pusan, Busan, Republic of Korea (Dec. 2016-). His research interests are Design of App, Green IT, Smart Grid, Network Security, Curriculum of Computer, High Availability Computing.

Seok-Hwa Jung, Received Bachelor of Engineering Degree from Department of Major of Computer Engineering from Pukyoug National University at Daeyeon, Busan, Republic of Korea in Feb. 2017. Hers research interests are Design of App, Game, Java, High Availability Computing.

Kyungryong Seo, received B.S. in Engineering Degree from Department of Major of Electrical Machinery Engineering from Pusan National University, Busan, Republic of Korea in Feb. 1983. M.S. in Degree in Electrical Engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Republic of Korea in Feb. 1990. Received the Ph.D. Degree in Electrical Engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Republic of Korea in Aug. 1995. Currently, he is a Full Professor (Tenure) of Computer Engineering Departments, Pukyong National University at Daeyeon, Busan, Republic of Korea. He received the Best Paper Award from Korea Multimedia Society six times (Nov. 2007, Nov. 2014, May. 2015, Nov. 2015, May. 2016, Oct. 2016). And he received Best Paper Award The 10th 2016 International Interdisciplinary Workshop Series from HSST (Aug. 2016). His research interests are High Speed Computer Network, Network Security, High Availability Computing.