

Design, Implementation and Performance Evaluation of a Novel PESS (Performance Enhanced Software Streaming) Technology

Chang Hun O*, Yong-Hee Jeon ^{0**}

* KSoft, Daegu, Rep. of Korea

** Department of Computer and Information Communications Engineering, Catholic University of Daegu,
Hayang, Gyeongsan, Rep. of Korea

ksoftceo@paran.com, yhjeon@cu.ac.kr

Abstract— In order to prevent illegal copies of software and make consumers be able to use the software more conveniently, we propose a novel PESS (Performance Enhanced Software Streaming) technology. This technology is a service method which can support several applications software via streaming in networks. In this paper, we first describe both the design and implementation aspects of the technology. The main features in the implemented system are both a platform to be used in multiple operating systems in addition to Windows system and enhanced performance.

We then compare the proposed system with some existing related systems and analyze several characteristics, and carry out the performance evaluation. Based on the performance evaluation, it is revealed that the ratio of CPU usage, memory utilization, and response ratio are greatly reduced. It is also shown that each application performance is much improved compared to that of general streaming technology.

Keywords— Software Streaming, Software Library, Streaming Technology, Performance Evaluation, Cloud Computing

I. INTRODUCTION

A wide variety of application software has been developing in order to be used in business environment as well as for individual users. As the result, individuals and corporate should spend huge amount of money and time for the purchase and maintenance of expensive software [1]. Furthermore, illegally copying and using software by individual and corporate cause the enormous waste of money and time. Therefore, illegal copies of software should be fundamentally prevented, and technologies which enable consumers to use the software conveniently are needed.

In order to solve the afore-mentioned problems, we are motivated to investigate a methodology for the efficient software usage by centralized management. In software streaming, users and managers could utilize software and information through network or online. If resources of server are available without installing software at the client's computer each time, it would be able to maximize the conveniences for users and the efficiency of maintenance. By adopting this technology, an organization may be able to save the time, resources, and expenses that otherwise should be

spent due to the installation and management of software by the previous methods.

In this paper, we propose a novel PESS technology and describe both the design and implementation aspects of the technology. We also present the performance evaluation results of the proposed system.

The rest of the paper consists of as follows. In Chapter II, we examine the evolution of computing infrastructure as the related works. In Chapter III, we describe components, algorithms, protocols, and implementation of the system. In Chapter IV, the performance evaluation results are presented. In Chapter V, some applications of the PESS system are given. Finally, we summarize this study in Chapter VI.

II. THE EVOLUTION OF COMPUTING INFRASTRUCTURE

A thin client application refers to computing environments which relies on its servers to process most or all of works. In client computer, it has minimal resources for just input and display. This is contrasted with the traditional fat client. Because this method heavily depends on the server processing via networks, high-performance servers and high-capacity networks are necessary. Accordingly, this results in the cost increase.

The implementation method of WBT (Windows Based Terminal), which was developed for the Windows operating system by Microsoft, is that each process runs on the server, but client's computer has only the function of a terminal [1]. Therefore, separate process resources for CPU, memory and disk is allocated per users. As the number of user access is increased through the networks, the server becomes heavily loaded, and as the result, the capacity of server requires to be very high. Otherwise the number of clients has to be restricted in order to get services.

There is also an inconvenience of clients having to connect to server all the time to use the data. Also, as the load cannot be distributed on server, users can only have service by connecting to a restricted or appointed server. It is hard to manage individuals, and the resulting increased maintenance work imposes the difficult scalability.

o: the corresponding author

Cloud computing is Internet-based computing in which resources and software are shared to computers on needed-basis. This is a paradigm shift for IT services based on the Internet and typically involves over-the-Internet provision of dynamically scalable and often virtualized resources [2]. Most cloud computing infrastructures consists of services performed by common centers and built on servers.

Figure 1 shows a typical cloud computing infrastructure.

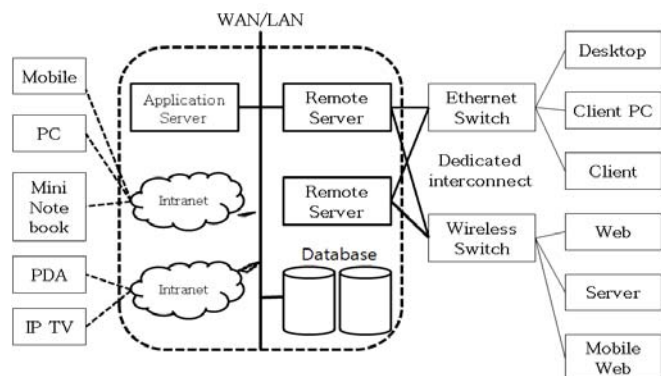


Figure 1 A Typical Cloud Computing Configuration

Table 1 summarizes the comparison of the related technologies.

Table 1 The comparison of the related technologies

Description	WBT service	Thin Client	Cloud Computing
Where software Installed	Server	Server	Server
Client	Simple Terminal	Simple Terminal	Low-performance Computer
Network Speed Required	Fast	Fast	Fast
Network Traffic Load	proportional to the number of connections		
Server Load	High	High	High
Data Storage	Server	Server	Server/Terminal

As shown in the table 1, the existing related technologies including the cloud computing share almost the same property.

The streaming technology has been developed since the late 1970s [2-5]. In 1997, Microsoft introduced the Windows media technology, and movies were supplied through Internet. As such, multimedia streaming technology made it possible to receive audios and videos in near real time.

Software streaming technology is based on DVP (distributed virtual paging). In terms of safety of data, media streaming can play the file even if the data is lost on the network, while software streaming cannot execute the file if data is lost on the network. With regards to the transmission method, media streaming sends in order in a push method, while software streaming sends necessary parts of data arbitrarily in dynamic pull method. Figure 2 shows a conceptual diagram of multimedia streaming.

Using the principles of multimedia streaming technology, the software streaming technology had been developed for the efficient usage and management of software. This technology

uses the virtual paging technique and runs a part of application software existing in the server at the client, as illustrated in the Figure 3. The client receives only necessary part of software for execution instead of downloading and directly installing the total software on its storage devices.

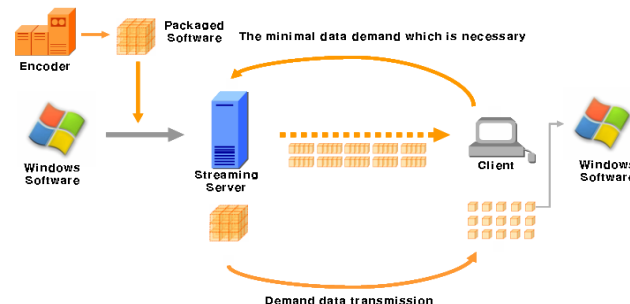


Figure 2 Conceptual Diagram of Multimedia Streaming

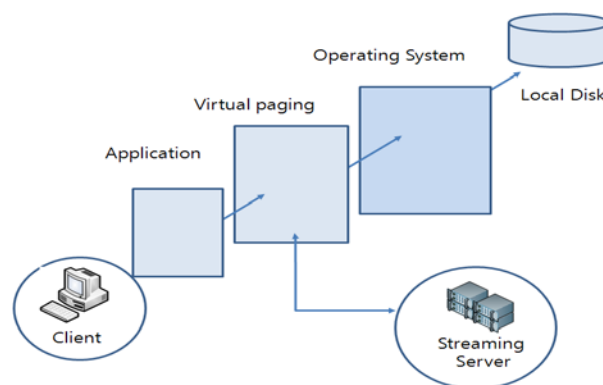


Figure 3 Software Streaming Structure Diagram

Major technologies used by the PESS system include the following:

- VFS(Virtual File System), Virtual Paging, System-related device driver technology
- The standard Internet Real-Time Protocol (RTP), Real-time Streaming Protocol (RTSP), Service Discovery Protocol (SDP), MMS etc.
- Execution file format techniques such as EXE, Byte Code etc.
- Software packaging format techniques such as JAR, Zip etc.
- Virtual run environment processing techniques such as OS, CLI, JVM, WIPI etc.
- Encryption techniques such as RSA public key encryption algorithm and DES (Data Encryption Standard)

III. DESIGN AND IMPLEMENTATION

At first, the initial basic environment for software streaming is provided by building separate WSP (windows software player), VRS (virtual registry system) and VFS (virtual file system) to client operating system as in Figure 4. After construction of the basic environment, the system creates the performance ready status. If the user demand software is

The diagram is divided into two main sections: 'WSP Install' and 'Run'.

WSP Install: Shows a 'WSP Install' icon (a document with a computer) pointing to a desktop computer icon. This leads to a horizontal bar representing the system components: OS, Registry System, File System, ZIStream, ZVRS*, and ZVFS*.

Run: Shows a 'WSP' icon pointing to a large box labeled 'PC'. Inside the 'PC' box are components: OS, Registry System, File System, ZIStream, ZVRS, and ZVFS. Below this is a 'HDD' box. To the right of the 'PC' box is a 'Server' box containing 'HDD', 'Config. Info.', and 'Program Files'. Below the 'Server' box are 'CPU' and 'RAM' boxes. Numbered arrows indicate the flow of data and execution:

- 1: From PC to Server
- 2: From Server to PC
- 3: From Server to RAM
- 4: From RAM to CPU
- 5: From CPU to ZVRS
- 6: From Server to ZVFS
- 7: From ZVFS to RAM

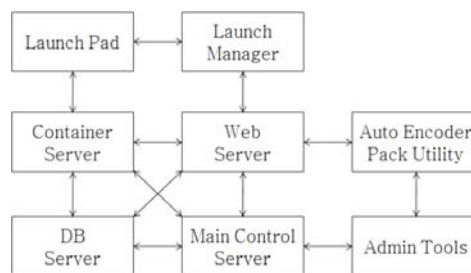
Legend:

- WSP : Windows Software Player
- ZVRS : ZIStream™ Virtual Registry System
- ZVFS : ZIStream™ Virtual File System

Those computers installed with the PESS client modules establish the connection with the PESS server modules. Thus both client and server recognize each other as if they are in the same computer. After a client is authorized and selects necessary software in the server, the initial required module is instantly streamed to the client and jobs are being progressed. Then the next sequences of software files are transmitted by streaming method according to the user's demand. These streamlined processes eliminate the need to install software on the user's computer.

[illegible]

The main components of PESS system provide efficient services by a variety of functions linked together. Figure 7 shows the system architecture.



The meaning of each important link is as follows:

- Launch Pad - Container Server: Service application data requested by the Launch Pad is delivered by the Container Server. The server stores the user's billing information.
- Launch Pad – Launch Manager – Web Server: When a user chooses a service application via Web interface, the Web server delivers the execution information of selected service application to the Launch Manager. Then the Manager produces the necessary parameters for execution and delivers them to the Pad.
- Container Server – Main Control Server: The main server registers/deletes/modifies the necessary packs for service at the Container Server.
- DB Server – Container Server: The user and pack information stored in the DB is transferred to the Container Server. The log data produced by the Container stores in the DB

- Main Control Server – Admin Tools: The Admin Tools provide interface to the Main Control Server for the configuration and management functions required for the system service.
- Auto Encoder – Admin Tools – Main Control Server: The service application is encoded to be suitable for the system service and the resulting generated pack is registered at the Main Control Server.

They are categorized into two elements: service element and management element. They are as follows:

- 1) Service components
 - Container server: role as actual sever, data delivery and processing
 - Launch manager: version control, upgrade
 - Launch pad: user application service
 - Web desk-top: user role interface
- 2) Management components
 - Central management server: role as central manager
 - Manager: server configuration, management, user book management
 - Pack encoder: creates contents(packs) to be used
 - Pack utility: perform internal data update

Figure 8 shows the manager information management protocol, as an example of several protocols implemented for the PESS.

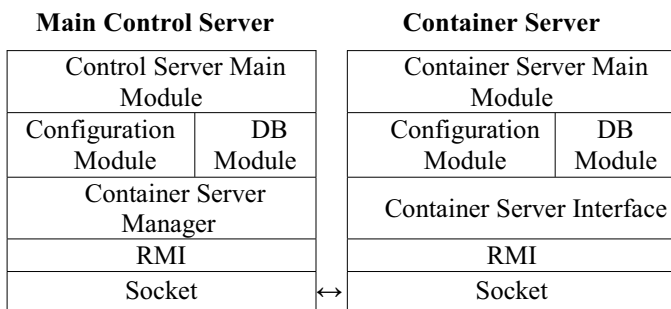


Figure 8 The manager information management protocol

Figure 9 represents the basic service flow for the PESS system.

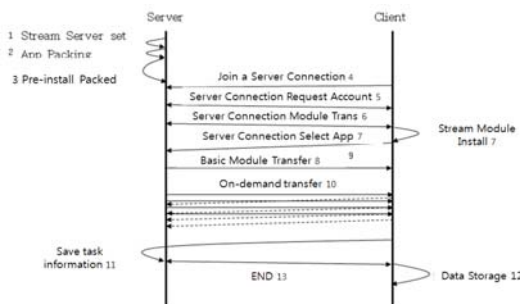


Figure 9 The Basic Service Protocol

It is important to define the system control flow for the system implementation. The following explanation is given to

show some operation sequences when an end-user tries to use a service application.

- The end-user chooses a service application to use via the system web desk-top.
- The system web desk-top retrieves the information for the requested service application from data base.
- The desk-top delivers the retrieved information to the launch manager so that the launch pad can finally be used.
- The launch pad interprets the required information for parameters and system services given from the manager.
- With the information, the pad accesses the container server and requests data required for the execution of service application.
- The launch pad initializes a virtual file system (VFS) required for asking the needed data to the container server.
- The launch pad runs the service application from the VFS.
- During the execution, the necessary file data is processed from the VFS. If the data exists in the cache, it is processed in the local system while it requests to the container server if not in the cache.

Figure 10 shows the service execution protocol, as an example of several protocols implemented in the PESS system.

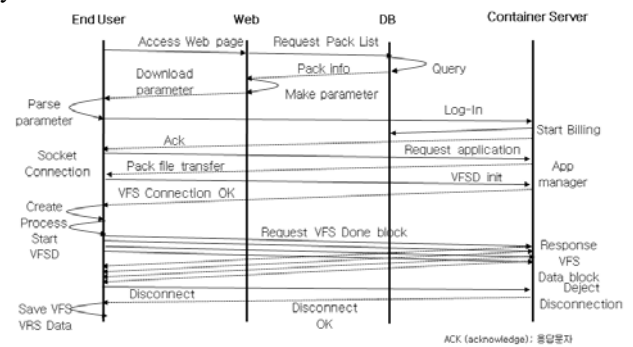


Figure 10 Service execution protocol

Figure 11 illustrates an example of the execution screen for the system.



Figure 11 An example of the system execution screen

IV. PERFORMANCE EVALUATION

For the performance evaluation we made the performance test bed as shown in Figure 12. We used an L2 switch, Windows 2000 computers as Client PC, and Compaq ML 370s with both Windows 2000 and Redhat Linux 7.2 as servers.

For the performance evaluation of the PESS system, we made five test scenarios depending upon the driving applications and the number of user created, with the total user number of 1,000. The created users by applications have been carried out by having a small time delay (about five seconds).

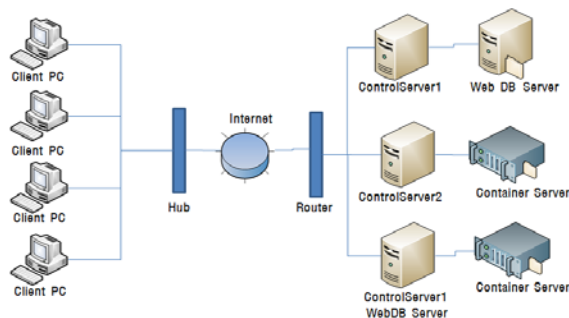


Figure 12 The performance test bed

Figure 13 shows the ratio of CPU usage of control and container servers.

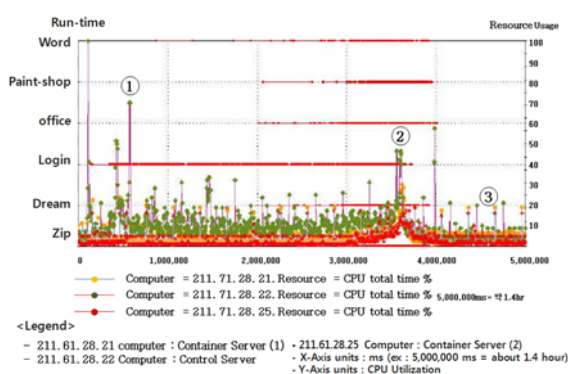


Figure 13 CPU utilization in control and container servers

After the connection duration of about one and a half hours, it shows the ratio of CPU usage in control server and container server. Mostly the ratio of CPU usage is below 20% for each server and even at the peak time, the ratio isn't over 40%. In terms of the control server, when users try to log on, they used below 25% of CPU. However, after the connection termination, it decreased to below 10%. This can occur because the web server and database server are working together on the control server. In terms of the control server, it typically shows below 25% of CPU usage. When the connection had been most frequent, the ratio had shown 40% and then it had decreased again. Overall, the control server has higher ratio of CPU usage than the container server.

The ratio of memory uses of the control server, container server and container server is shown in Figure 14.

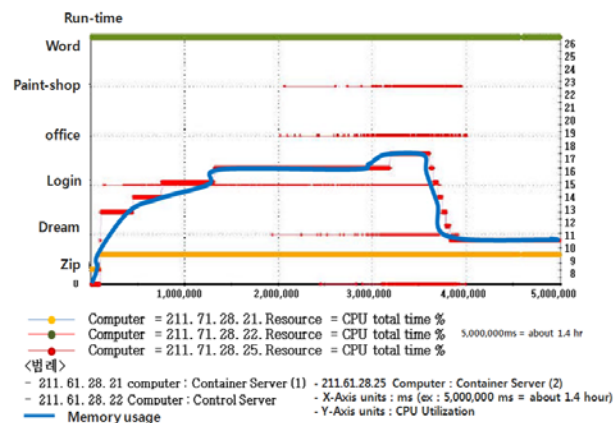


Figure 14 Memory utilization of each server

The memory usage for the container server shows a certain ratio without a big change. On the other hand, the ratio of memory usage of the control server is shown to increase, when simultaneous users were increased.

Figure 15 represents the login response ratio to the web server. As illustrated in the figure, the result of the performance 2 shows 240.7 % of that of the performance 1 in terms of login time.

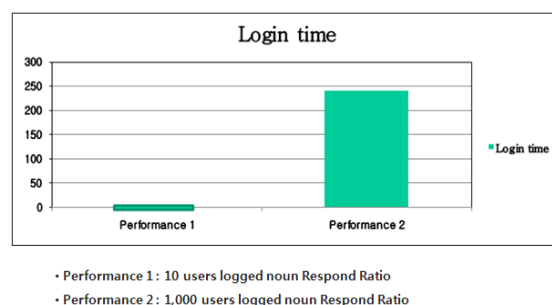


Figure 15 Response Ratio by the number of login users

V. APPLICATIONS

Figure 16 represents an example of the usage of PESS system via networks in an Intranet environment of an organization.

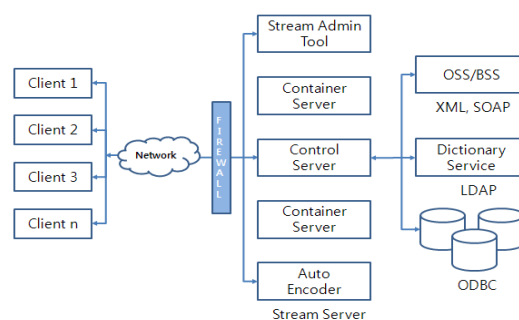


Figure 16 An example of the usage of PESS system via network in an organization

Figure 17 represents the conceptual diagram for an example of portable streaming. By installing the required applications in USB or portable HDD, it is possible to run the application without installing the application at the client computer in mobile environment.

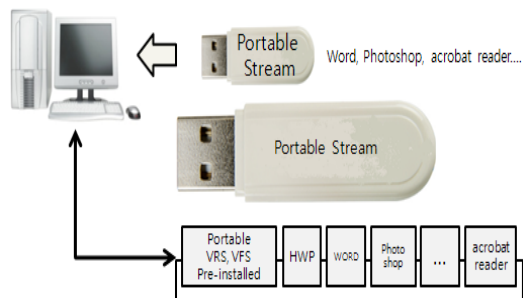


Figure 17 The conceptual diagram of portable streaming

Figure 18 illustrates a type of service configuration for high activity of server and large number of simultaneous users (more than a thousand users) through either all-in-one server or control server cluster.

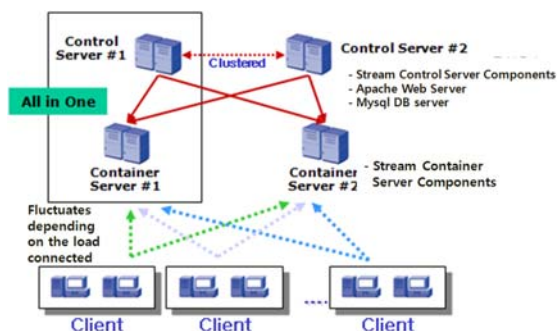


Figure 18 Configuration types of servers

Figure 19 represents a construction of wide area networks for the PESS system. By this architecture, load balancing is achievable depending on the characteristics and property of applications.

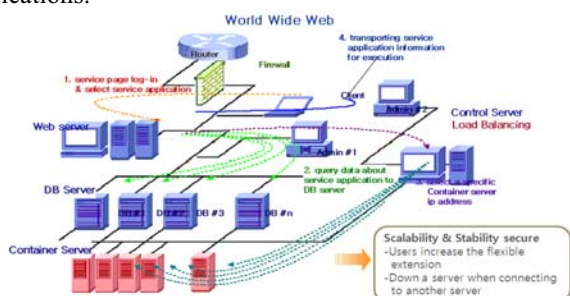


Figure 19 Server configuration and load balancing structure

VI. CONCLUSIONS

In this paper, we proposed platform independent PESS technology. We designed and implemented the technology based on Java language. The main features in the implemented system are both platforms to be used in multiple operating systems in addition to Windows system and enhanced performance. In the implemented streaming method, application software is placed on the server and only necessary packs are transmitted in an instant. By virtual file system and clients' virtual registry, if necessary, users process by transmitting a very small pack unit. Therefore, server load can be reduced and the streaming speed can also be improved.

We compared the proposed system with some existing related systems and analyzed several characteristics, and carried out the performance evaluation. (Due to the page limitation, the performance comparison results are not shown in this paper.) Based on the performance evaluation, it is revealed that the ratio of CPU usage, memory utilization, response ratio is greatly reduced. It is also shown that each application performance is much improved compared to that of general streaming technology.

The main contributions of the proposed technology can be summarized as follows [6]:

- By adopting the proposed technology, network performance may be improved and server load can be reduced as well.
- Software management works can be reduced a lot and both systematic and scientific management is possible.
- The application software isn't installed on the hard disk of the computer, so the available disk capacity may be increased. The cost of purchasing hardware equipment can therefore be reduced.
- In the use of Internet, users do not have any time or space limitations, so user's convenience is greatly increased.

REFERENCES

- [1] Yong-Hee Jeon, Chang-Hun O and Jung-Sook Jang, "Design and Implementation of Java-based Software Streaming Services with Enhanced Performance," Int. J. of Computer Science and Network Security, Vol. 8, No. 2, pp. 72-81, Feb. 2008.
- [2] H. Tan, D.L. Eager, and M.K. Vernon, "Delimiting the Range of Effectiveness of Scalable On-Demand Streaming," Proc. Performance, Rome, Italy, Sept. 2002.
- [3] Anirban Mahanti, "On-Demand Media Streaming on the Internet Trends and Issues", Department of Computer Science, University of Saskatchewan. Canada, Comprehensive Examination Paper, Dec 14, 2001.
- [4] Shim J. M., Kim W. Y., Choi W., "Design and Implementation of Network Fault-tolerant Software Streaming," Journal of Korea Contents Association, Vol. 6, No. 10, pp. 126-133, Oct. 2006.
- [5] <http://softonnet.com/korean/download/Whitepaper%20v0%209.pdf>
- [6] Chang-Hun O, Ph. D dissertation, Catholic University Of Daegu, Implementation and Evaluation of Platform Independent PESS (Performance Enhanced Software Streaming) Technology, Feb. 2010.