Advanced Software On-demand based on Functional Streaming

Jeong Min Shim[1], Won Young Kim[1], Wan Choi[1],

[1] Electronics and Telecommunications Research Institute (ETRI), Daejeon, South Korea
{jmshim, wykim, wchoi}@etri.re.kr

Abstract. Streaming is a technology that enables either real-time or on-demand distribution of multimedia contents over network. Recently streaming technology has been applied onto applications, and many deployment tools for enterprise applications have been developed. Software streaming is a technology to provide software whichever users need on-demand in real-time by using streaming technology without downloading and installing a full package in advance before its use. Software streaming technology has many issues that are application load time, network fault-tolerant and etc. In this paper, we discuss issues for software streaming technology. Then, we propose a new SOD system based on functional streaming called *Advanced Software On-Demand (ASOD)* system. Also, we present schemes to solve issues that are application load time and network fault-tolerant.

1 Introduction

Streaming is the process of playing application while it is still downloading [1] [2]. Streaming has been mostly found on streaming media that lets users listen to or view the digitized contents such as sound, animation and video, as it is being downloaded.

Recently streaming technology has been applied onto applications, and many companies have developed deployment tools for enterprise applications such as AppStream's AppStream.NOW platform [3], Softricity's SoftGrid platform [4], Stream Theory's AppExpress platform [5] and SoftOnNet's Z!Stream [6]. Software streaming is a technology to provide software whichever users need on-demand in real-time by using streaming technology without downloading and installing a full package in advance before its use.

Software streaming technology still has many issues that are application load time, network fault-tolerant and etc. To solve these problems, we will define software by differentiating environment for launching, basic function and additional functions. Basic function is first and certainly necessary contents to launch an application. Additional functions are contents for each component (a set of the menus).

In this paper, we discuss issues for software streaming technology, and present advanced Software On-Demand (ASOD) system based on functional streaming using basic function and additional functions. Also, we present schemes to solve issues that are application load time and network fault-tolerant.

2 Issues in existing Software On-Demand (SOD) service

In existing SOD system, the client requests page contents to the streaming server (only, when the page contents are not found in the local cache) when an application tries to process a function of the application streamed to execute. To get contents, client will send one or more request messages to the streaming server. Operation of an application is suspended until the client receives all required contents.

We performed experiments to measure necessary contents to launch an application. Table 1 show results of *Application Launching Size* (ALS) for Linux Application. As shown in Table 1, in all application, a lot of contents are required when the application is launched. This fact implies that a client must send a lot of page requests. Also, users must wait for a long time after they request service.

Network is one of the important issues in SOD service. If a client loses connection with a streaming server, it is not able to request a page to the streaming server. If users try to use the function which is not stored in a local cache, the application will be destroyed or the client system may be crashed. Consequently, if network connection fails, the service has to be stopped although users can use a function which is stored pages in a local cache.

Table 1. Application Launching Size (ALS) versus total size of an application

Application	Application Launching Size	Total size
CBtracker	3.1 MB, of total size 100%	3.1 MB
Bubble Shooter	6.03 MB, of total size 99%	6.1 MB
Abiword	4.6 MB, of total size 15.7%	29.4 MB
OpenOffice	98.3 MB, of total size 41.5%	236.3 MB

3 Advanced Software On-Demand based on Functional Streaming

3.1 Architecture of the ASOD system based on functional streaming

To provide SOD service based on functional streaming, preliminary work that analyzes an application is needed. First, we define new transmission unit between a client and a streaming server, named Functional Unit (FUint). The FUnit consists of contents for one or more menus. There are two kinds of FUnit: (1) basic FUnit and (2) extra FUnit. Basic FUnit is first and certainly necessary contents to launch an application. Extra FUnit is contents for each component (a set of the menus). We analyze software execution and extract necessary information for functional streaming. We should be able to extract this information by extracting statistical data from simulations.

Figure 3 is architecture of client and streaming server based on functional streaming. A client system in SOD system consists of the following components: (1)

Streaming Application (SA), (2) Event Hooker (EH), (3) Application Streaming File System (ASFS) and (4) Streaming Data Manager (SDM). SA is an application running through SOD service. EH intercepts functionality, is selected by user, of an application when network fault occurs. ASFS communicates with the streaming server to get FUnits. SDM stores FUnits and information for FUnits streamed from the streaming server, and maintains information for all FUnits of an application. SDM also maintains relationship between the FUnits and functionalities.

A streaming server consists of (1) Streaming Data Package (SDP) and (2) Data Package Information (DPI). SDP is FUnits, are extracted through *Software Analyzer*, for applications. DPI maintains information for application packages and FUnits. DPI is used to find out FUnit which is corresponding to function required from the client.

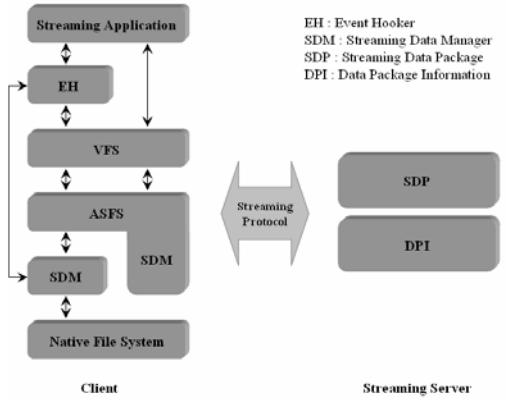

Fig. 1. Architecture of client and streaming server in the ASOD system

3.2 Techniques in the ASOD system based on functional streaming

As experimental result in section 2.1, most applications need contents, called basic FUnit, above 40% of the total software size to launch an application. An application cannot launch until basic FUnit completely are arrived. In existing SOD system, a client sends page request message of several tens and hundreds to streaming server to get basic FUnit. Therefore, whenever the client requests the page, the streaming server searches it and then transmits it to the client.

In the proposed system, application load time can be reduced by using basic FUnit. When a client requests a software streaming service to the streaming server, a streaming server sends immediately basic FUnit and FUnit information for the application to launch the application to the client without another request of the client. Accordingly, application load time must be reduced significantly.

The proposed SOD system supports network fault-tolerant. If a client loses its connection to the streaming server, the client will be notified of the situation and

client may continuously use the application with the functionality that it might have. If a menu clicked by a user is not in a local cache, process for the menu is ignored by EH. Consequently, users can continuously use the application, although network fault occurs.

In the proposed SOD system, we use a prefetching technique to reduce an application suspension time. The prefetching is a technique that sends the FUnit to be expected from the streaming server to the client without user's demand in advance. There are three different ways to apply prefetching: (1) prediction by producers, (2) prediction by static statistics, and (3) prediction by dynamic statistics. Prediction by producers is that the order of FUnits for the application is decided by producers without any statistics. A streaming server sends FUnit to a client in sequence when a service is begun by request of a user. Prediction by static statistics is that content providers or packers decide a transmission-priority by using statistics collected from each user. When an application through the SOD service is launched, the streaming server first sends basic FUnit, and extra FUnits are transmitted by decided priority after. After a priority is decided, it doesn't change. Prediction by dynamic statistics is similar to prediction by static statistics. But, prediction by dynamic statistics updates a priority of extra FUnits by using statistics continuously collected from each user that uses an application provided through SOD service.

4 Conclusion

In this paper, we proposed advanced SOD system based on functional streaming for more efficient SOD service. In the proposed SOD system, a transmission unit between a client and streaming server is FUnit. We also presented schemes to solve issues in existing SOD system. To reduce application load time, a streaming server sends basic FUnit without to wait for request message of a client when a service begins. The proposed SOD system supports network fault-tolerant that users can continuously use functions which is stored in the local cache although network fault occurs. Therefore, we introduce prefetching technique based on statistical information to reduce the application suspension time significantly.

References

1. Bitpipe, "Streaming Media Services," http://www.bitpipe.com.
2. California Software Labs (CSWL), "Basic Streaming Technology and RTSP Protocol," http://www.cswl.com.
3. AppStream Inc., "AppStream Technology," http://www.appstream.com.
4. Softricity Inc., "The SoftGrid Application Virtualization Platform," http://www.softricity.com/home/index.asp.
5. Stream Theory, "The Enterprise Software Distribution Platform," http://www.streamtheory.com.
6. SoftonNet Inc., "Z!Stream Technology," http://www.softonnet.com.