

다중 운영체제 플랫폼 소프트웨어 스트리밍 서비스의 구현과 성능평가

정회원 오 창 훈*, 종신회원 전 용 희**, 정회원 송 동 호***

The Implementation and Performance Evaluation of a Multi-OS Platform Software Streaming Services

Chang-Hun O* *Regular Members*, Yong-Hee Jeon** *Lifelong Members*,

Dong-Ho Song*** *Regular Members*

요 약

네트워크 스트리밍 기술은 네트워크에서 스트리밍을 통하여 여러 가지의 애플리케이션 소프트웨어를 지원할 수 있는 서비스 방법이다. 멀티미디어 스트리밍 기술이 개발되어 광범위하게 사용되고 있지만, 소프트웨어 스트리밍 기술에 대한 연구는 활발하게 진행되지 않고 있다. 본 논문에서는, 개선된 성능을 가진 자바 언어 기반 소프트웨어 스트리밍 기술을 설계 및 구현하고, 기존 관련 시스템과 비교하고 여러 가지의 특성을 분석하고 성능 평가를 수행한다. 구현된 시스템은 윈도우 체제 이외에 다중 운영체제에서 사용될 플랫폼을 가지고 있다. 또한 클라이언트의 컴퓨터에 소프트웨어를 설치하지 않고 사용할 수 있으며, 사용자의 편리성이 극대화되는 이상적인 특징이 있다.

Key Words : Software streaming service, Multimedia streaming, Characteristic analysis, Performance evaluation

ABSTRACT

Network streaming technology is a service method which can support several application software via streaming in networks. Although multimedia streaming technology has been developed and widely used, research on the software streaming technology is not yet being progressed actively. In this paper, we design and implement a software streaming technology based on Java language with enhanced performance, compare the existing related systems with it and analyze several characteristics, and perform the performance evaluation. The implemented system has a platform to be used in multiple operating systems in addition to Windows system. It can also be used without installing a software at the client's computer, and has a novel feature that the user's convenience is maximized.

I. 서 론

컴퓨터의 급속한 보급과 인터넷 환경의 대중화에 따라 다양한 컴퓨터 시스템과 소프트웨어 사용과 보다 빠른 네트워크 성능을 가진 시스템을 요구하

고 있다. 특히 인터넷 방송이나 디지털 영상저장장치(DVR: digital video recorder) 등과 같이 멀티미디어 서비스를 위한 첨단 장비 즉, 효율적인 서버 시스템 및 관리기술 또는 고급 네트워크 기술과 장비들이 필요하다^[1]. 이런 멀티미디어 서비스의 실현

* 대구가톨릭대학교 대학원 컴퓨터정보통신공학과(fineoch@hanmail.net),

** 대구가톨릭대학교 컴퓨터정보통신공학부(yhjeon@cu.ac.kr), *** 소프트온넷(주) (dhsong@softonnet.com)

논문번호 : KICS2007-02-051, 접수일자 : 2007년 2 월 9 일, 최종논문접수일자 : 2007년 12 월 12 일

을 위하여 컴퓨터는 고성능화 되어야 하며, 이에 따른 네트워크의 부하가 증가되고 고성능 서버 장비의 추가 및 네트워크 시스템 증설을 요구하고 있다. 다양한 분야에서 고성능 컴퓨터 환경에 적합한 응용 소프트웨어들이 개발되고 업무에 활용 되면서, 개인 및 조직에서는 고가의 소프트웨어의 구입 및 관리에도 막대한 시간과 비용이 지불되고 있다.

한편 소프트웨어 개발 기업 측면에서는 개인이나 조직에서 소프트웨어의 불법 복제 및 무단 사용으로 인해 막대한 자금과 시간을 허비하게 되고, 새로운 소프트웨어 개발 의욕을 상실하게 만들기도 한다. 그러므로 소프트웨어의 불법 복제를 근본적으로 방지하고 사용자가 필요한 소프트웨어를 편리하게 사용 가능한 기술이 필요하며, 지속적인 수입원 창출을 위하여 보다 편리하고 효율 가치 있는 소프트웨어를 연구 개발할 필요가 있다. 소프트웨어 관리 측면에서도 구입비용의 적절한 편성 및 예산 절감, 체계화 된 시스템을 통한 관리 효율성을 기하고, 때 번 업그레이드되는 소프트웨어의 배포 또는 설치에 따른 시간과 인건비의 절감, 정확한 통계에 의한 과학적이고 체계적인 관리의 필요성이 요구된다.

이미 윈도우 기반의 터미널서비스 같은 기술들이 일부 사용 되고 있지만 클라이언트 자원을 활용하지 못하며 별도의 하드웨어 장비가 부착되거나 서버 부하의 문제 등 여러 가지 불편한 내용이 있어 서버 부하를 최소화 하면서도 전송을 편리하게 하는 등의 보다 편리하고 효과적인 방법에 대해 연구하게 되었다. 이러한 문제점들의 해결 방안을 위한 여러 가지 접근 방법 중 사용자 및 관리자가 네트워크나 온라인을 통해 사용하고자 하는 소프트웨어 및 정보를 활용 할 수 있는 방법을 찾고자 한다.

소프트웨어를 매번 컴퓨터에 직접 인스톨 하지 않고 서버의 자원을 활용 할 수 있다면 여러 가지 사용자 편리성을 추구 할 수 있는 동시에 관리의 효율성을 극대화 할 수 있다. 이런 방법에는 멀티미디어 스트리밍과 소프트웨어 스트리밍 방법이 있는데, 본 논문에서는 소프트웨어 스트리밍에 대하여 기술하고자 한다. 국내에 소프트웨어 스트리밍 제품이 나와 있지만, 관련 논문이 없는 실정이다²⁾. 본 논문에서는 자바 언어 기반 다중 운영체제 지원 소프트웨어 스트리밍 서비스의 설계 및 구현에 대하여 기술하고, 시스템에 대한 특성을 분석하고, 성능 평가에 대하여 기술하고자 한다.

본 논문에서는 먼저 스트리밍에 관한 개념 및 발전 배경에 대해 살펴보고, 기존의 윈도우에서 사용

하고 있는 웹 베이스의 터미널 서비스의 기능에 대해 알아보고 그 문제점을 도출하고, 그에 따른 개선 방향을 모색하며 각 기능별로 문제점에 대한 차이를 연구하여 개선된 성능을 가진 기술을 제시하고자 한다. 본 논문의 나머지 구성은 다음과 같다. 제 II장에서는 관련 연구로 스트리밍의 개념 및 발전 과정 등을 알아보고 기존 사용 방식의 사례 분석 및 다른 응용 사례를 살펴본다. 제 III장에서는 구현된 스트리밍 기술의 개발 환경 및 구현 방법을 설명하고, 제 IV장에서는 기존 방식들과의 차이점등을 비교하고 구현된 기술의 특징을 분석한다. 제 V장에서는 성능 평가 및 고찰을 수행하고, 마지막으로, 제 VI장에서는 본 논문의 연구 결과를 도출하고 향후 연구 과제를 제시한다.

II. 관련연구

2.1 개요

그림 1에서는 멀티미디어 스트리밍의 개념도를 보여준다. 그림 1에서 동영상 파일은 동시에 파일 전체를 보내는 것이 아니라, 시간의 흐름에 따라 해당 분량만큼을 조금씩 보내게 된다. 동영상 파일의 용량이 크기 때문에 한꺼번에 파일 전체를 보내주 기란 불가능하고, 이를 해결하기 위해서 조금씩 파일의 일부만(실제로 영상이 재생되는 분량만큼만) 실시간으로 전송해주는 체제이다^{3,4)}.

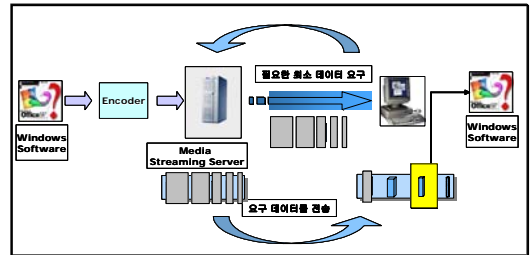


그림 1. 멀티미디어 스트리밍 개념도

표 1. 미디어와 소프트웨어 스트리밍의 차이

	미디어 스트리밍	소프트웨어 스트리밍
데이터의 안정성	데이터가 네트워크상에서 손실되더라도 재생 가능	데이터가 네트워크상에서 손실되면 사용 불가능
구조	순차적 구조	논리적 구조
전송 방식	Push 방식	동적인 Pull 방식

인터넷에서 동영상을 다운로드가 아닌 실시간 전송과, 구현이라는 개념을 가능케 한 것이 바로 스트리밍 기술이다. 서버에서 파일을 작게 분할하여 계속 보내면 받는 PC에서는 분할된 파일 조각들이 들어오는 대로 바로 재결합되어 원래의 파일로 재생하는 방식을 말한다.

스트리밍 관련 기술은 1970년대 후반부터 관련 기술이 꾸준히 발전하여 1989년 다운로드 미디어 기반(downloading media-based) 재생방식이 출현된 이후, 1994년 초 Real Media사로부터 오디오 스트리밍 시스템(audio streaming system)을 세계 최초로 개발하여 오디오 관련 시장에서 상용화 되었고, 1995년 MPEG사가 비디오 스트리밍(video streaming) 기술을 개발하여 CD롬을 통한 영화 보급 및 동영상 전송 방식 관련 기술의 선도 자리를 차지하고 있다. 그러나 1997년 마이크로소프트사가 윈도우 미디어 기술을 선보이며 인터넷을 통한 동영상보급이 활발히 시장에 출시되면서 멀티미디어 환경에 적합한 윈도우 방식으로 가장 널리 보급되게 되었다. 이와 같이 멀티미디어 스트리밍 기술은 음성 및 동영상을 실시간으로 받아 볼 수 있도록 하는 기술을 말하며 물 흐르듯이 데이터를 흘러 보내주기 때문에 하나의 동영상 파일이 완전히 이용자의 컴퓨터로 전송되지 않아도 영상 및 사운드를 보거나 들을 수 있다⁵⁾.

소프트웨어 스트리밍 기술은 분산 가상 페이징(DVP: distributed virtual paging) 기반 스트리밍 기술이며, 자세한 내용은 3.2절에서 설명하도록 한다. 여기서는 차이점에 대해서만 알아본다. 데이터의 안정성 측면에서 미디어 스트리밍은 데이터가 네트워크상에서 손실되더라도 재생이 가능 한 반면, 소프트웨어 스트리밍은 데이터가 네트워크상에서 손실되면 사용이 불가능하고, 구조적 측면에서 미디어 스트리밍은 순차적 구조를 가지고 있는 반면에, 소프트웨어 스트리밍은 논리적 구조로 되어 있다⁶⁻¹⁰⁾. 전송 방식에서는 미디어 스트리밍은 푸시(push) 방식으로 영화의 필름이 돌아가듯 순차적으로 전송 되는 것인 반면, 소프트웨어 스트리밍은 동적인 풀(pull) 방식으로 임의로 필요한 부분의 데이터만 전송되어진다. 이를 비교표로 살펴보면 표1과 같다¹¹⁻¹³⁾.

2.2 윈도우 기반 단말 서비스

마이크로소프트사가 윈도우 운영체제에서 개발한 윈도우 기반 단말(WBT: Window Based Terminal) 서비스 기술의 구현 방식은 그림 2에서 보는 것과

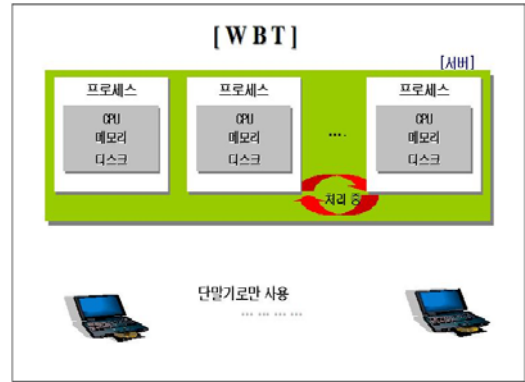


그림 2. 윈도우 기반 단말 서비스 구현방법

같이 서버 상에서 각각의 프로세스들이 돌아가고 있고 클라이언트의 컴퓨터는 단지 단말기의 기능밖에 없다. 따라서 사용자별 CPU와 메모리 및 디스크가 별도의 프로세스 자원을 할당 받게 됨으로 사용자 증가 시, 바로 서버에 상당한 부하를 주게 되며 결과적으로 서버의 성능이 아주 고성능이 되어야 하거나 클라이언트 접속자 수가 아주 제한적으로 접속하여 서비스를 받을 수밖에 없게 된다. 모든 클라이언트들은 서버를 통한 자원을 사용하게 되며, 동영상 재생 시 또는 서버로부터 화면과 소리를 따로 받아 클라이언트에서 합성(mixing)하여 재생하는 방식이고, 프린터나 스캐너, 스피커 등 주변 기기를 사용 시 서버에서 연결하여 출력을 재전송하여 사용하게 됨에 따라 클라이언트들이 사용하기에 상당히 불편한 점이 있다.

또한 클라이언트 컴퓨터에서 작업한 데이터를 저장해야 할 경우는 서버 상에 파일이 생성되며 서버에 직접 저장하게 된다. 따라서 클라이언트가 일일이 서버와 접속하지 않고는 사용 할 수 없어 불편한 점이 있다. 아울러 자원 연결 시 인식이 불가능한 경우가 많아 작업 지연 등이 발생 할 수 있고 사용자나 관리자의 편의성에 한계가 있다.

또한 서버에 부하 분산 자체가 불가능함에 따라 제한 또는 지정된 서버에만 사용자가 연결하여 서비스를 받을 수 있고, 서버 관리 라이선스 및 사용자 관리 정책의 일관성을 두기가 어렵다. 프로그램 설치 업그레이드 시 전체 서버를 관리해야 하고 사용자 증가 시 별도의 서버를 증설해야 하고 개별 관리가 힘들어 관리 업무가 증가되어 확장성이 결여된다. 네트워크 부하 측면에서는, 서버로부터 모든 변경된 내용을 화면으로 전송 받는 구조로 많은 화면 변화가 발생 되는 경우 프로그램은 더 많은

네트워크 트래픽을 유발하게 되고, 사용자 증가와 네트워크 트래픽은 정비례 관계를 가지며 네트워크 부하를 가중 시키게 된다. 따라서 이 방식의 기술 구현은 상당한 고성능 서버장비 및 고가의 네트워크 시스템의 사양을 요구하게 되어 사용자 및 관리자의 예산 절감이 힘들게 된다.

III장에서는 이와 같은 문제를 해결하고자 구현한 소프트웨어 스트리밍 기술의 설치 및 구성에 대하여 기술한다.

2.3 소프트웨어 스트리밍 제품 현황

최근에 여러 업체에서 소프트웨어 스트리밍 기술을 이용한 제품을 연구하거나 출시하고 있다. ETRI에서는 네트워크 고장 감내 소프트웨어 스트리밍 기술을 구현한 에버그린을 개발하였다^[14].

AppStream은 응용 프로그램의 배포, 관리 및 서비스를 지원하기 위하여 스트리밍 기술을 이용한다^[15]. 그 이외에 Softricity^[16], Stream Theory^[17] 사 등이 소프트웨어 스트리밍 관련 제품을 제공하고 있다.

III. 설치 및 구성 예

본 시스템의 개발 도구는 JDK1.3.1, IBM VisualAge for Java (EJB), Visual Studio 6.0 (ActiveX), PHP를 이용 하였고, 운영환경 중 서버 환경은 HTTP 서버, JDK 1.3.1, Linux 계열, Solaris 계열, 윈도우 서버 계열, 클라이언트 환경은 마이크로 소프트 익스플로어, 윈도우즈 계열을 이용 하였다.

3.1 설치

소프트웨어 스트리밍 방식은 그림 3에서 보는 것 같이 먼저 클라이언트 컴퓨터의 운영체제에 스트리밍이라는 별도의 윈도우 소프트웨어 플레이어(WSP: windows software player)를 가상레지스터 시스템(VFS: virtual registry system)과 가상파일시스템(VFS: virtual file system)을 구축하여 기본 환경을 만들어 놓고 실행 대기 상태를 만들게 된다. 사용자가 원하는 소프트웨어를 실행하면, 클라이언트의 기존의 운영체제의 레지스터나 파일시스템과 상관없이 이미 만들어진 가상의 소프트웨어 플레이어 즉, 가상 레지스터 시스템과 가상 파일 시스템을 통해 실행되며 네트워크를 통해 서버상의 하드 장치내의 구성환경 중 필수 요소만 순간 스트리밍을 통해 클라이언트 컴퓨터 메모리(RAM)에 적재 한 후 사용자 요구(on-demand)에 따라 서버 하드(HDD)의 프로그램 파일의 필수

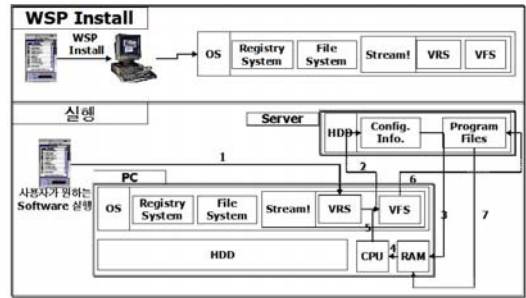


그림 3. 소프트웨어 스트리밍 설치 및 실행절차

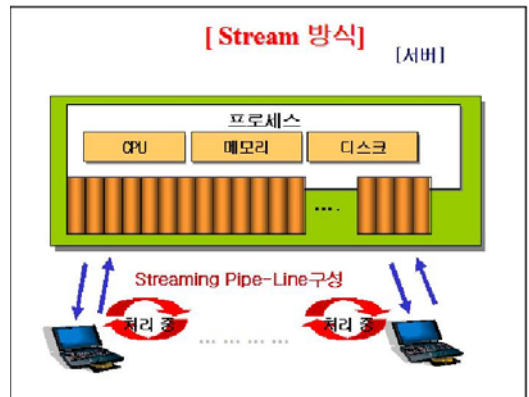


그림 4. 스트리밍 구현 방법

팩만 아주 작은 단위로 쪼개어 순간 스트리밍 방식으로 네트워크를 통해 클라이언트 컴퓨터의 메모리(RAM)에 로딩한 후 클라이언트 컴퓨터의 중앙처리장치(CPU)를 통해 실행하게 된다.

그림 4에서 보여 주듯이 스트리밍 방식은 기본적으로 서버로부터 자원을 받아 클라이언트 컴퓨터에서 프로세싱이 됨으로, 사용자 증가 시 동일한 프로세서 내에서 인스턴스(child-node)만 생성하게 된다. 즉, 사용자 증가 시에도 시스템의 부하를 최소화 할 수 있다. 따라서 디스크 할당이 발생치 않음을 알 수 있다.

기존의 윈도우 기반 단말 서비스 방식에서처럼 별도의 프로세서 자원을 할 당 받는 것이 아니기 때문에 사용자 증가 시에도 서버의 막대한 부하는 초래 하지 않게 된다. 따라서 서버 장비 및 네트워크장비 등이 고가의 고성능을 요구하지 않는다.

3.2 구성

이 절에서는 소프트웨어 스트리밍 기술의 구성 예제를 보여주기 위하여 멀티미디어 스트리밍의 구성과 비교하여 기술한다.

3.2.1 서버 프로세스

그림 5의 위부분과 같이 서버 프로세스에서 일반적으로 멀티미디어 스트리밍은 각각의 텔레사인, 테이프, 브로드캐스트, 카메라, 파일 등의 소스들을 비디오 캡처 카드, 디지털 디스크 리코더, 오디오 캡처 카드의 Acquisition을 통해 파일 인코딩, Live Broadcasting, Multi-Channel Audio 등으로 인코딩하고 Web Authoring Embedding화 한 후 윈도우 미디어 서버에 탑재 한다. 반면 소프트웨어 스트리밍 방식은 그림 5의 아래 부분처럼, 윈도우 소프트웨어를 바로 파일 별로 인코딩 한 후 윈도우 소프트웨어 서버에 탑재 한다.

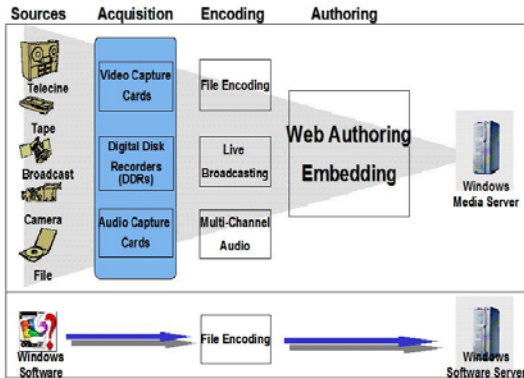


그림 5. 서버 프로세스에서의 구성 예

3.2.2 클라이언트 프로세스

그림 6에서는 클라이언트 프로세스에서의 구성 예를 보여준다. 윈도우 미디어 플레이어의 Splitter는 오디오와 비디오를 분리해서 각각의 오디오 코덱과 비디오 코덱을 통해 오디오 Renderer와 비디오

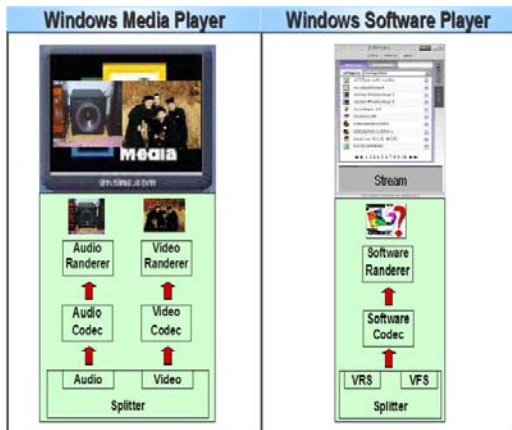


그림 6. 클라이언트 프로세스 구성 예

Renderer로 각각 프로세싱 되는 반면, 윈도우 소프트웨어 플레이어는 Splitter에서 SRS와 VFS를 통해서 소프트웨어 코덱 처리를 통해 Software Rendering 된다. 따라서 기존의 윈도우 기반 단말 서비스 방식 등의 음성과 영상이 따로 전달되는 문제점을 개선할 수 있다.

IV. 특성 분석

4.1 장치 구성도

그림 7의 장치 구성도와 같이 소프트웨어 스트리밍 방식의 장치 구성은 서버의 자원을 순간 전송 방식을 통해 클라이언트 컴퓨터로 스트리밍 받고 각각의 프로세싱이 클라이언트 컴퓨터에서 이루어지기 때문에 데이터를 클라이언트 컴퓨터에 직접 저장 가능하고, 프린터, 스캐너, 플로피, 카메라 등 주변 기기를 클라이언트 컴퓨터에서 직접사용 할 수 있는 것이 가장 큰 장점이며, 멀티미디어 응용프로그램 지원을 거의 완벽하게 지원 받을 수 있다. 또한 LAN/WAN 환경의 영향을 받지 않고 별도의 셋팅 또는 변경 없이 편리하게 사용 할 수 있다. 따라서 서버와 터미널 간의 통신에 높은 대역폭이 소요되거나, 동시 사용자가 증가 할 경우 사용이 어려워지는 기존 방식에 비해 훨씬 사용이 편리하게 된다.

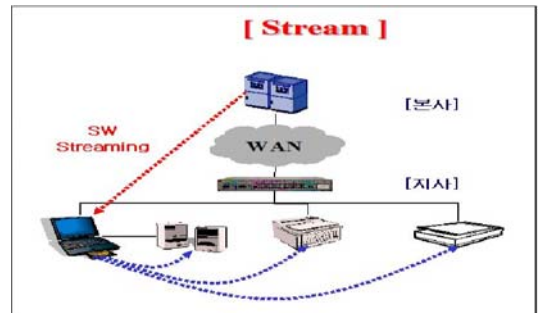


그림 7. 장치 구성도

4.2 관리 확장성

그림 8에서 보여주는 바와 같이, 소프트웨어 스트리밍 방식의 관리 확장성 문제는 관리자가 별도 또는 메인 컨트롤 서버를 두고 단일 지점 관리만 하고, 각각 필요 요소에 따라 여러 군데 컨테이너 서버를 분산 해 두어 관리하고 부하의 정도에 따라 유동적으로 연결되고, 각각의 컨테이너 서버에는 여러 가지 다양한 응용 프로그램을 탑재하고 사용 할 수 있다.

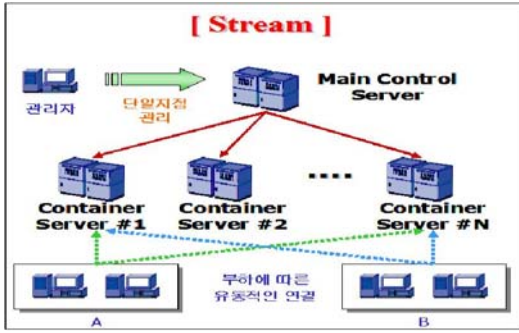


그림 8. 관리 확장성 개념도

따라서 단일 지점 중앙 관리가 가능하며, 라이선스 관리 또는 프로그램 배포, 사용자관리, 그룹 관리가 메인 컨트롤 서버에서 완벽 하게 지원되기 때문에 프로그램 설치 및 업그레이드 시 메인 서버만 관리하면 된다. 따라서 사용자 증가 시 별도의 서버를 증설 할 필요가 없어 기존 방식의 관리 방식의 부재 및 관리 업무의 증가와는 대조적으로 한결 수월한 관리 도구이다.

4.3 네트워크 부하 및 사용량

제안된 방식의 또 다른 특징 중 하나는 기존방식의 네트워크의 부하 및 서버부하를 최소화 한 것이라 할 수 있다. 기존 윈도우 기반 단말 서비스방식은 하나의 서버에 여러PC가 접속해, 화면 전체의 이미지100% 그대로 클라이언트 PC로 불러와 데이터를 처리하기 때문에 네트워크 트래픽이 사용자 수 증가에 따라 정비례하게 증가하게 되고, 그래픽 관련 프로그램을 사용할 경우 평균 한 서버 클라이언트 사용자 수 20~30 유저를 초과하면 심각한 네트워크 트래픽 및 서버 부하가 발생하게 된다.

반면 제안된 방식은 서버의 자원을 참조하여 필요한 시점에 프로그램을 아주 작은 단위로 슬라이스(쪼개어)해 필요한 시점에 클라이언트 PC로 불러(On-demand방식)와 데이터 처리는 개별 클라이언트 PC에서 처리되고 개별 클라이언트 PC하드의 데이터를 활용하고 저장함으로써 네트워크 부하에 별 영향을 주지 않는 것이 특징이며 사용자 편리성이 제고된다 할 수 있다.

4.4 서버구성 및 부하 균형

소프트웨어 스트리밍 방식의 서버 구성 및 부하 균형은 그림 9와 같이 클라이언트 사용자가 웹을 통해 서비스를 신청하면 라우터 및 방화벽을 통해 서비스 페이지를 통해 로그인 한 후, 사용 할 응용

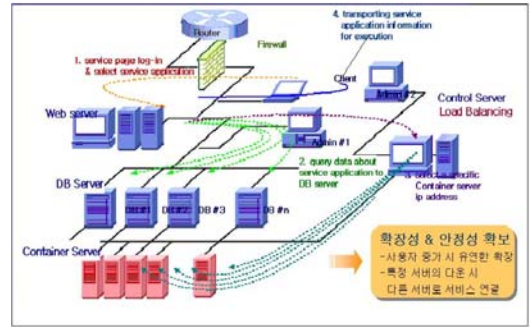


그림 9. 서버 구성 및 부하 균형

소프트웨어를 선택 하게 되면 메인 컨트롤 서버의 Admin에서 각각의 데이터 베이스서버로부터 서비스 될 프로그램에 질문 데이터를 받아서 컨테이너 서버의 IP를 통해서 선택된 특별한 응용 프로그램 요청한 클라이언트의 컴퓨터로 전송되어 서비스를 시작 할 수 있다.

특히 사용자 증가 시 에 따른 유연한 확장이 가능하고, 특정 서버가 다운 시 다른 서버로 자동 접속되어 연결된다.

4.5 기존 방식과의 차이점 분석

4.5.1 주문형 방식

그림 10에서 보듯이 기존 방식과의 차이점은 사용자 요구에 따른 주문형(on-damand) 방식이라는 점이다. 즉, SOD(software on demand) 서버에 다양한 응용 소프트웨어를 단 한번 설치해 두고 개별 컴퓨터에 프로그램을 설치하지 않은 상태에서 누구든지, 언제든지, 어디서든지 소프트웨어를 개별 인증을 받아서 사용 할 수 있다는 점이다. 따라서 기업 및 학교 등지의 관리 효율성 극대화 및 비용 절감 효과가 획기적으로 일어날 수 있다는 점이다.

반면에, 그림 11에서와 같이 기존 클라이언트-서버 기반 컴퓨터는 클라이언트의 모든 동작이 서버로 전달, 서버에서 연산 후 클라이언트로 전송되어 클라이언트 컴퓨터는 단말기로만 사용되는데 반해

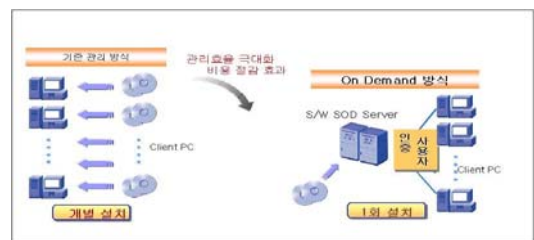


그림 10. 주문형 소프트웨어 방식 개념도

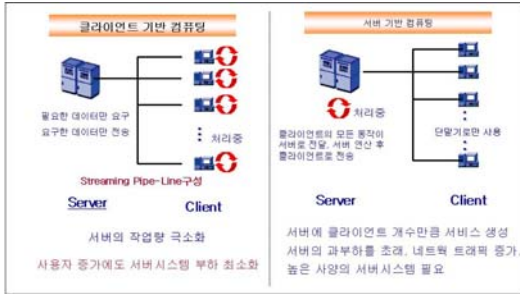


그림 11. 기존 클라이언트-서버 방식의 특징

클라이언트 컴퓨팅은 필요한 데이터만 요구하고 요구한 데이터만 스트리밍 파이프라인을 통해 전송받아 클라이언트에서 프로세싱 되는 것이 가장 특징적이다. 따라서 서버에 클라이언트 개수 만큼 서비스가 생성되어 서버의 과부하를 초래하고 네트워크 트래픽이 증가 되어 높은 고성능의 서버시스템이 필요하게 되는 기존 방식과 대조된다.

4.5.2 콤포넌트 특징

표 2를 통해 비교 해 볼 수 있듯이 WMT (windows media streaming technology)는 제작도구에 여러 가지 윈도우 미디어 인코더 (Encoder), Windows Media Author, Windows Media On-demand producer, Vid to ASF, Wav to ASF들이 필요한 반면 WSST (windows software streaming technology)는 Windows

표 2. 윈도우미디어 기술과 윈도우 소프트웨어 스트림 기술의 비교

	WMT	WSST
제작 도구	Windows Media Encoder Windows Media Author Windows Media On-demand producer Vid to ASF Wav to ASF	Windows Software Streaming Encoder (Builder)
편집 도구	Windows Media ASF indexer ASFChop ASFCheck	
서비스	Windows Media Unicast Service Windows Media Station Service Windows Media Program Service Windows Media Monitor Service	Windows Software Streaming Administrator Windows Software Streaming Monitor
Player	Windows Media Player	Windows Software Player

Software Streaming Encoder(Builder)만 필요하다. 편집도구로 WMT는 Windows Media ASF indexer, ASFChop, ASFCheck이 필요한 반면 WSST는 별도의 편집 도구가 필요하지 않는다는 점이다. 서비스는 Windows Media Unicast Service, Windows Media Station Service, Windows Media Program Service, Windows Media Monitor Service가 필요한 반면, WSST는 Windows Software Streaming Administrator,와 Windows Software Streaming Monitor만 필요하다. 플레이어는 각각의 미디어 플레이어와 소프트웨어 플레이어가 필요하다.

표 3. 윈도우 기반 단말 서비스와 스트리밍 방식의 특징 비교

항목	윈도우 방식 (카드형포함)	스트리밍방식	비교
구현방식	화면전송 방식	S/W 스트리밍	편리한 사용법
서비스	OA 일부	OA, 그래픽, 프로그래밍언어, 유틸리티, 게임 등 다양	다양한 서비스
멀티미디어 지원	제한적	3D, 동영상,사운드, 투르퀄러 등 모든 멀티미디어 환경 지원	멀티 미디어 지원
클라이언트 OS	윈도우 95,96,ME, 2000,XP	윈도우 95,96,ME,2000, XP	다양한 운영 환경 지원
서버 OS	윈도우2000, XP 서버	윈도우 2000/XP 서버, 리눅스, 솔라리스, 기타 유닉스	기존 유휴장비의 다양한 활용방안 지원
서버 사양	고가형 서버	저가형 서버	구축비용절감
확장성	지원불가.사용자 증가시 별도 서버관리	로드 밸런싱 클러스팅	중앙집중적인 관리와 유연한 확장성 지원
1서버 동시사용자수 (동일사양)	20-30유저 프로그램 따라 가변적	200유저 프로그램 종류에 관계없음	월등한 성능
적용분야	소규모 LAN구간, 단말기 사용 부하가 적은 소규모 프로그램	규모와 무관, LAN/WAN 모두 가능 3D, 동영상 그래픽 프로그램	사용의 편리성

4.5.3 특징 비교

표 3의 특징 비교와 같이 구현 방식에서 윈도우 기반 단말 서비스는 화면 전송방식, 스트리밍 방식은 소프트웨어 스트리밍 방식을 통해 사용자 편리성을 추구 하였고, 서비스 되는 응용 소프트웨어는 윈도우 기반 단말 서비스 단순 OA 일부에 국한 되나, 스트리밍은 OA, 그래픽, 프로그래밍언어, 유틸리티, 게임 등 다양한 응용 소프트웨어의 서비스를 제공 할 수 있다.

특히 멀티미디어의 지원에서는 윈도우 베이스 터미널 서비스는 영상과 음성의 일치가 힘들고 극히 제한적으로 활용 가능하나, 스트리밍 방식은 3D, 동영상, 사운드, 트루컬러, 등 거의 모든 멀티미디어 환경에서 완벽 지원이 가능하다. 사용자(클라이언트) OS에서는 윈도우 기반 단말 서비스와 스트리밍 방식이 모두 Win95, 98, ME, 2000, XP에서 지원이 가능한 반면 Server OS에서는 윈도우 기반 단말 서비스는 Windows 2000와 XP Server에 국한되어 제한되게 사용 되고 있지만 스트리밍 방식은 Windows 2000/XP Server, Linux, Solaris, 기타 Unix의 서버에서도 지원이 가능하다는 점이 가장 큰 차이점으로 들 수 있고 기존 유휴 장비를 다양하게 활용 할 수 있다는 점이다.

서버의 사양에서는 윈도우 기반 단말 서비스 방식은 아주 고성능의 서버를 사용 해야만 서비스를 지원 할 수 있지만, 스트리밍 방식은 일반 저 성능 컴퓨터 서버로도 거의 모든 서비스를 지원 할 수 있어 구축비용을 현저하게 절감 할 수 있다. 확장성에서는 윈도우 베이스 터미널 서비스 방식은 사용자 증가 시 별도의 서버를 증설하여야 하나 스트리밍 방식은 부하 균형 클러스팅을 통해 보았듯이 메인 서버 외에는 소프트웨어 컨테이너만 별도로 두면 다양하게 확장 및 관리가 가능해 중앙 집중적인 관리와 유연한 확장성이 장점이다.

기존의 국내외에서 비교적 활성화되어 있는 기술인 터미널 서비스와 투스카니 방식 등과 같은 윈도우 기반 단말 서비스 방식과 본 논문에서 제안된 스트리밍 방식의 차이점에 따른 효과는 표 3에서 다양한 분야에서 사용자 편리성의 극대화에 대한 성능 개선 사항을 제시하고 있다.

적용 분야에서 윈도우 기반 단말 서비스 방식은 소규모 또는 LAN 구간 및 은행 업무처럼 DB 접속 방식인 단말기 형태에 적합하고 부하가 적은 프로그램 사용에 적합하며 별도의 터미널 라이선서가 필요한 반면, 스트리밍 방식은 규모와 무관하며,

LAN/WAN 모두 가능하며, 3D, 동영상, 그래픽 등 모든 프로그램에 적용 가능하며 별다른 하드웨어 구축 장비가 필요하지 않다.

V. 성능 평가 및 고찰

5.1 성능 시험

5.1.1 테스트 환경

성능 시험 구성도는 그림 12와 같다.



그림 12. 성능시험 구성도

그림 12의 구성도를 참고하여 성능 시험에 사용된 개별 시스템을 살펴보면 다음과 같다.

- Switching Hub Paxcomm NEX 3024 L2 Switch
- Lanchpad (Client) 조립 PC (Windows 2000) Rational Performance Studio
- Lanchpad (Client) (Windows 2000)
- Performance Testing Tool
Zionfocus RackMount System(5)
(Windows 2000 Professional) [Virtual User Generator]
- License Server
- Control Server(1) - Compaq ML370 (Windows 2000 Server)
- Web, DB server(1) - Compaq ML370 (Windows 2000 Server)
- Container Server(1-1) - Compaq ML370(Windows 2000 Server)
- Container Server(1-2) - Compaq ML370(Windows 2000 Server)
- Control Server(2) - Compaq ML370(Redhat Linux 7.2)
- Web, DB Server(2) - Compaq ML370(Redhat Linux 7.2)
- Control Server(2) - Compaq ML370(Redhat Linux 7.2)

표 4. 시스템 사양

구분	기종	OS	IP	사양	기타
Control Server(1) Web.DB Server(1)	Compaq ML570	windows 2000 Server (Service Pack 2)	211.61.28.10	CPUGH(2개) RAM:1GB	MySQL 3.23.X PHP4.1.X IIS5.0 Agent 2002.5.20 버전
Container Server(1-1)	Compaq ML370	Windows2000 Server (Service pack 2)	211.61.28.23	CPU:1GHz RAM:2GB	Performance Studio 2002.5.20버전 Agent 2002.5.20 버전
Container Server(1-2)	Compaq ML370	Windows2000 Server (Service pack 2)	211.61.28.24	CPU:1GHz RAM:2GB	Agent 2002.5.20 버전
Control Server(1) Web.DB Server(1)	Compaq ML370	RedHat 7.2	211.61.28.22	CPU:1GHz RAM:2GB	MySQL 3.23.X PHP4.1.X IIS5.0 Agent 2002.5.20 버전
Container Server(2-1)	Compaq ML370	RedHat 7.2	211.61.28.21	CPU:1GHz RAM:2GB	Agent 2002.5.20 버전
Container Server(2-2)	Compaq ML370	RedHat 7.2	211.61.28.25	CPU:1GHz RAM:2GB	Agent 2002.5.20 버전
Virtual User Generator 1-5	Zion Lack Mount	Windows2000Server (Service pack 2)	211.61.28.28 ~ 211.61.28.30	CPU:1GHz RAM:2GB	Performance Studio 2002.5.20버전
Client 1	조립PC	Windows2000 Pro	211.61.28.44	CPU:850Mhz RAM:256MB	
Client 1	삼성PC	Windows XP HE	211.61.28.43	CPU:1.7Ghz RAM:512MB	
Client 1	조립PC	Windows ME	211.61.28.46	CPU:850Mhz RAM:512MB	
Client 1	조립PC	Windows 98	211.61.28.45	CPU:850Mhz RAM:256MB	

- Control Server(2-1) - Compaq ML370(Redhat Linux 7.2)
- Control Server(2-2) - Compaq ML370(Redhat Linux 7.2)

표 4는 시스템 사양을 보여준다.

5.1.2 스트림 서버(리눅스의) 성능 평가

표 5의 내용은 구동 애플리케이션과 생성 사용자를 나타낸 표로서 캐시데이터가 없는 상태에서 한 사용자가 애플리케이션을 하나 구동 시킨 후 종료하고 나온 과정을 총 5개의 시나리오를 만든 후, 1000명의 사용자를 표 5와 같이 동시에 생성시켰다. 애플리케이션별로 생성된 사용자들은 약간의 지연시간(약 5초)을 두어 실행 하였다.

그림 13은 컨트롤, 컨테이너 서버의 CPU 사용률을 보여준다. 여기서 컨트롤 서버는 전체 서비스에 대한 관리 임무를 담당하고, 컨테이너 서버는 사용자의 컴퓨터에 필요한 애플리케이션 코드를 전달하는 서버이다.

표 5. 구동 애플리케이션과 생성 사용자 수

구 분	생성 사용자	비 고
Winzip	300	30%
한글 97	150	15%
오피스 워드	150	15%
페인트 샵 프로	200	20%
드림위버	200	20%
총 User 수	1,000	100%

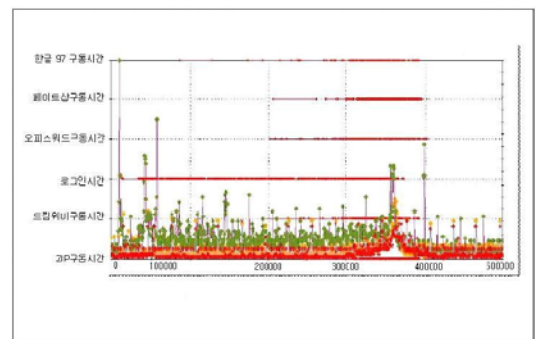


그림 13. 컨트롤, 컨테이너 서버의 CPU 사용률

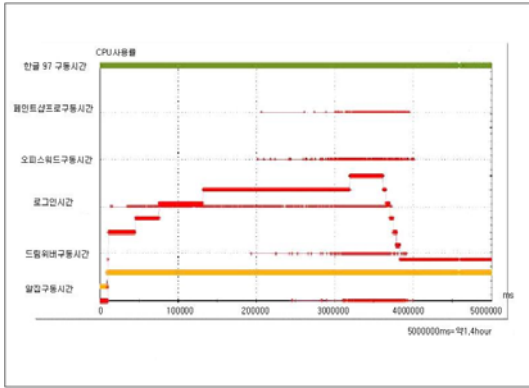


그림 14. 서버별 메모리 사용률

약 1시간 30분 동안 1000명의 사용자가 접속한 결과 컨트롤 서버, 컨테이너 서버(1), 컨테이너 서버(2)에 대한 CPU 사용률을 보여준다. 대체로 각 서버 모두 20% 이하의 CPU 사용률을 나타내고 있으며, 최고 정점 시간에서도 40%를 넘지 않고 있다. 컨트롤 서버는 사용자가 로그인 시도 할 때 CPU를 거의 25% 이하로 사용하다가 사용자 접속이 완료 된 후에는 10% 미만으로 감소하였다. 이는 컨트롤 서버에 웹서버와 데이터베이스 서버를 같이 구동 시키고 있기 때문에 나타나는 현상으로 볼 수 있다. 컨트롤 서버는 전체적으로 25% 미만의 CPU 사용률을 보이다가 접속이 가장 많이 생길 때 40% 정도 사용하다가 다시 감소하는 것으로 나타났다. 전체적으로 컨트롤 서버는 컨테이너 서버보다 약간 CPU 사용률이 높은 경향을 보이고 있다.

컨트롤 서버, 컨테이너 서버(1), 컨테이너 서버(2)에 대한 메모리 사용률은 그림 14와 같다. 대체로 컨테이너 서버의 메모리 사용률은 커다란 변동 없이 일정한 비율로 사용되고 있음을 알 수 있으며, 컨트롤 서버는 동시 접속자가 증가 될수록 메모리 사용률이 증가되는 경향을 보이고 있어 접속자 수에 따라 변동률이 의존적임을 알 수 있다. (참고로 컨테이너 서버(1)가 컨테이너 서버(2) 보다 전체적으로 메모리 사용률이 크게 나타난 것은 도구 및 기타 메모리 상주 프로그램이 같이 구동되어 있기 때문이다.)

5.1.3 응답시간 비교

웹 서버에 대한 로그인 응답률은 그림 15와 같다. 그림의 표에서 보는 바와 같이 10명의 사용자가의 접속에 비해 1000명의 사용자가 약 240배 로그

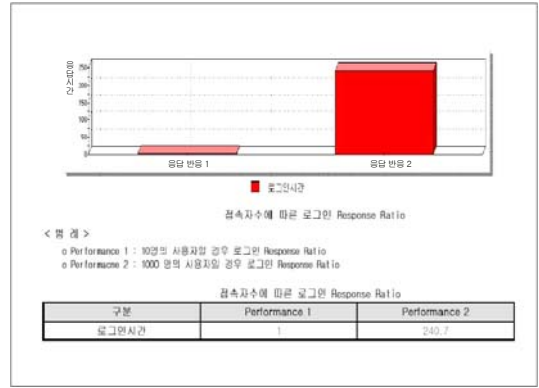


그림 15. 접속자별 응답비율

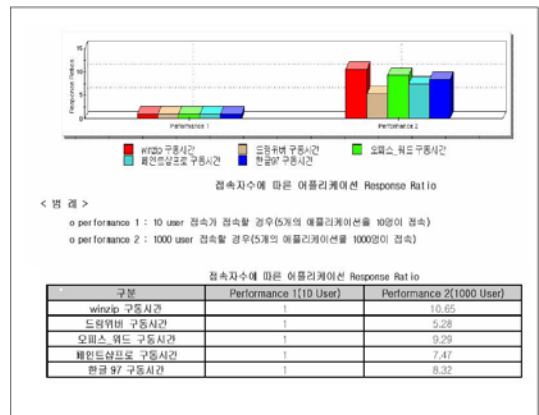


그림 16. 접속자수에 따른 애플리케이션 응답비율

인 응답 시간이 나타남을 볼 수 있다. 위 결과는 제품 자체의 결과라기보다는 웹서버(아파치서버)의 성능과 하드웨어의 성능에 더 깊이 관련 된 것으로 보인다. 따라서 하드웨어 성능을 높여서 로그인 응답 시간을 줄 일 수 있는 부분이다.

컨테이너 서버에 대한 애플리케이션 응답비율은 그림 16과 같다. 10명의 사용자가 접속하여서 각각 하나의 애플리케이션을 구동한 반응 시간은 10명이 접속한 것에 비해 그림 16의 표에서 보는 것과 같이 Winzip은 10.65배, 드림위버는 5.28배, 오피스-워드는 9.29배, 페인트샵 프로는 7.47배, 한글 97은 8.23배 반응시간이 나타난 것을 알 수 있다.

애플리케이션의 크기에 따라 반응 시간의 차이가 달리 나타났으며, Winzip이 사이즈에 비해 반응 시간이 10.65배나 나온 이유는 전체에서 Winzip을 구동한 사용자가 30%로 가장 많았기 때문이다.

5.2 고찰

본 논문에서 제안한 스트리밍 방식은 아래와 같이 사용자 편리성을 극대화할 수 있다.

- ① 조직 내 1,000명이 1,000대의 PC에서 평균 10 종류의 프로그램을 업무용으로 사용 할 때, 기존 방식은 개별 PC에 10종류의 프로그램을 각각 설치하기 위해 사용자 개인 및 전산담당자는 상당한 시간을 소요하여 인스톨해 사용했으나, 제안된 방식은 서버에만 10종의 프로그램을 단 한 번만 설치하고 네트워크가 설치된 곳에서 언제 든지 필요한 경우 즉시 네트워크로 접속하여 사용 가능하여 관리효율 및 사용편리의 극대화를 추구할 수 있다.
- ② 아울러 최근 화려한 그래픽 등의 사용으로 업무용 프로그램의 크기(용량)가 점차 커지고 있어 사용자 PC는 여러 가지 프로그램을 설치함으로 점차 고급화 사양의 H/W를 요구하게 되지만, 제안된 방식의 경우는 개별 PC에는 OS 외 사용 프로그램을 일체 인스톨 하지 않으므로 메모리 공간의 확장을 요구하지 않고, 고성능의 H/W의 잦은 업그레이드도 필요치 않아 자원의 절감을 피할 수 있어 H/W활용의 극대화가 가능하다. 반면 기존 방식은 개별 PC에 설치한 프로그램의 잦은 버전 변화에 따라 업그레이드를 위한 재설치(인스톨)에 따라 하드의 마모가 잦게 되고 결국 H/W의 손상이 자주 일어나게 된다.
- ③ 개별 PC에는 프로그램이 설치되어 있지 않는 메일 첨부 파일 등을 열기위해서는 인터넷 등에서 관련 프로그램을 다운로드 받아 설치하거나 인스톨된 PC를 찾아서 첨부 파일을 열 수 있지만 제안된 방식은 한 단번 클릭만으로 바로 파일을 열 수 있어 사용자의 업무 편리성이 극대화를 가져온다.
- ④ 기존 방식은 서버에 다양한 버전을 설치하여 사용할 수 없고 개별 PC에 설치된 프로그램과 버전이 다를 경우 충돌이 발생하지만, 제안된 방식은 서버에 다양한 프로그램 버전을 설치하여 사용자가 필요시 선택하여 사용할 수 있고, 개별 PC에 인스톨 된 프로그램 버전과의 충돌이 일어나지 않아 효율적이다.

또한 본 논문에서 제안된 스트리밍 방식은 다음과 같은 예산 절감 효과가 있다.

- ① 조직 내 1,000명의 인원이 1,000대의 PC에서 평균 10가지 업무용 소프트웨어를 사용 할 때,

기존 방식은 개별 PC에 각각 10종류의 S/W를 설치하기 위해선 1,000대×10종류의 소프트웨어를 모두 구매하여 사용하여 왔고, 부처 이기주의로 인해 특정부서에서 구매해 사용하고 다른 부서에서 같은 프로그램을 중복 구매하는 심각한 낭비를 초래하였으나, 개선방식은 S/W를 도서관의 도서개념으로 조직에서 꼭 필요한 양만큼(즉, 동시사용자 수)의 수량을 구매하면 조직 내 1,000대의 사용자가 네트워크를 통해 단 한번 클릭만으로 편리하게 모든 사용자들이 사용 가능하기 때문에 예산의 획기적 절감이 가능하다.

- ② 아울러 개별 PC의 H/W의 잦은 교체를 방지 가능함으로 H/W의 절감도 획기적이라 할 수 있다.

또한 소프트웨어 불법 사용 방지를 위한 효율적인 방안이 된다. 기존 컴퓨터 사용자는 절대적인 프로그램 사용요구에 비해 구입 예산의 부족 또는 부처 간 이기주의 등으로 인한 조직 내의 프로그램의 편중현상 또는 불법복제 및 불법사용이 공공연하게 이루어져 왔으나, 제안된 방식을 이용하면 굳이 불법을 하지 않고도 필요한 경우 서버에 접속해 자유롭게 사용 후 프로그램 종료만 하면 되므로 사용자가 편리하게 활용하여 업무효율의 획기적 증진이 가능하다. 뿐만 아니라 관리자 측면에서는 사용자별, IP별, 그룹별(부서), 프로그램별 기간별(일,월,분기,년) 등 다양한 사용현황이 과학적으로 분석 가능해 프로그램 관리의 과학적 관리가 가능하여 프로그램 관리자의 혁신적 업무 효율 증대가 가능하다.

VI. 결 론

본 논문에서는 개선된 성능을 가진 자바 언어에 기반한 소프트웨어 스트리밍 기술을 설계 및 구현하고, 기존 관련 시스템과 비교하고 여러 가지의 특성을 분석하고 성능 평가를 수행하였다. 구현된 시스템은 윈도우 체제 이외에 다중 운영체제에서 사용될 플랫폼을 가지고 있다. 또한 클라이언트의 컴퓨터에 소프트웨어를 설치하지 않고 사용할 수 있으며, 사용자의 편리성이 극대화되는 이상적인 특징이 있다. 구현된 스트리밍 기법은 응용 소프트웨어를 서버에 위치 해 놓고 필수 팩만을 스트리밍 방식을 통해 순간 전송해 클라이언트의 가상 레지스터리와 가상 파일 시스템을 통해

사용자가 필요 시 아주 작은 팩 단위로 전송 해 프로세싱 할 수 있는 기술로 서버 부하 감소 및 스트리밍 속도를 개선 할 수 있다. 요약하여 기술 하면, 구현된 기술은 다음과 같은 성능 개선 효과를 얻을 수 있다.

먼저, 단순히 네트워크 성능의 개선 또는 서버 부하 최소화 뿐 만 아니라 정확한 사용 통계를 통해 조직내의 소프트웨어의 구입을 적절하게 할 수 있어 구입예산의 획기적 절감이 가능하다. 둘째, 관리자의 소프트웨어 관리업무를 대폭 줄일 수 있으며 관리기법의 수치화 가 가능해 체계적이고 과학적인 관리가 가능해졌다. 셋째, 사용자가 구입 또는 업그레이드 하는 소프트웨어를 매번 인스톨 하는 번거로움을 덜 수 있어 사용자 편리성을 극대화 할 수 있다. 넷째, 응용 소프트웨어를 컴퓨터의 하드 디스크에 인스톨 하지 않기 때문에 디스크 용량의 증가를 줄일 수 있는 등 하드웨어 장비 구입비용을 절감 할 수 있다. 다섯째, 인터넷을 이용해 시간과 공간의 제약 없이 활용 가능해 사용자 편리성을 극대화 할 수 있다.

현재 정부에서 추진하는 전산자원의 ASP사업(렌탈 등)에 각종 하드웨어 위주로 이루어지고 있지만 앞으로는 프로그램의 ASP가 궁극적 목적이며, 이의 미들웨어로 제안된 방식이 채택되어 시범 실시되고 있다.

참 고 문 헌

[1] 백국실, 원유진, 심보옥, "Java 기반 MPEG-4 스트리밍 클라이언트 개발" *한국정보과학회 가을 학술발표 논문집(1)*, Vol. 28. NO. 2, pp.447, 2001.

[2] [http://softonnet.com/korean/download/Whitepaper %20v0%209.pdf](http://softonnet.com/korean/download/Whitepaper%20v0%209.pdf)

[3] Rebert Cohen and Radha, "Streaming file-Grained Scalable Video over packet-Based Networks," *Global Telecommunications Conference. 2000. GLOBECOM', IEEE. Vol.1.* pp.288-292, 2000.

[4] 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.

[5] T.P. Nguyen and A. Zakhor, "Distributed Video Streaming Over Internet," *Multimedia*

Computing and Networking, Jan. 2002.

[6] 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

[7] A, Dan, B. Levine, B. Lyles, H. Kassar, and D. Balsiefien, "Deployment Issues for the IP Multicast Service and Architecture", *IEEE Network*, 14(1)10-20, Jan/Feb. 2000.

[8] J.M. Almedia, D.L. Eager, and M.K. Vernon. "A Hybrid Caching Strategy for Streaming Media Files," *In Proc. MMCN '01*, San Jose, CA, Jan. 2001.

[9] D.L. Eager, M.K. Vernon, and J. Zahorjan, "Minimum Bandwidth Requirements for On-Demand Data Delivery," *IEEE Trans. on Knowledge and Engineering*, Vol. 13, No. 5, Sep./Oct. 2001.

[10] D.L. Eager, M.K. Vernon, and J. Zahorjan, "Bandwidth Skimming: A Technique for Cost-Effective Video-On-demand," *Proc MMCN '00*, San Jose, CA, Jan. 2000.

[11] D.L. Eager, M.C. Ferris, and M. K. Vernon. "Optimized Regional Caching for On-Demand Data Delivery," *In Proc. MMCN '99'*, San Jose. CA, Jan. 1999.

[12] D.L.Eager, M.K,Vernon, and J, Zahorjan. "Optimal and Efficient Merging Schedules for Video-on-demand Servers." *in Proc. ACM Multimedia '99' Orlando. FL.* Nov. 1999.

[13] R. Rejaie, M. Handley. H. Yu, and D. Estin. "Proxy Caching Mechanism for Multimedia Playback Streaming in the Internet." *In Proc. IEEE Infocom '99 New York, NY.* Mar. 1999.

[14] 심정민, 김원영, 최완, "네트워크 고장감내 소프트웨어 스트리밍 기술의 설계 및 구현", *한국콘텐츠학회논문지 제 6 권 제 10호*, pp.126-133, 2006년 10월.

[15] <http://appstream.com>

[16] <http://www.softtricity.com>

[17] <http://www.streamtheory.com>

오 창 훈 (Chang-Hun O) ??회원



1983년 3월~1990년 7월 고려대학교 행정학과
2001년 3월 ~ 2003년 2월 대구가톨릭대학교 대학원 컴퓨터 정보통신공학과 석사
2003년 3월~2005년 2월 대구가톨릭대학교 대학원 컴퓨터정보통신공학과 박사수료

1990년 3월~1995년 7월 주식회사 한성
1995년 12월~2001년 7월 한국소프트웨어개발원 부설 교육원 원장
2001년 5월~현재 한국소프트웨어개발원 원장
2001년 3월~현재 계명문화대학 전산정보학과 겸임교수
2003년 1월~현재 산업자원부 중소기업경보화 컨설턴트
2003년 7월 대구, 경북 소프트웨어기술대전 금상 수상
2003년 12월 2003 한국소프트웨어기술대전 은상 수상 (임베디드부분)
2003년 1월~현재 ISO 9000(품질) 심사원
2004년 1월~현재 ISO14000(환경) 심사원
2001년~2004년 정보통신부 IT전문인력양성교육 최우수 교육기관 3년 연속 선정기관 원장
<관심분야> 네트워크 보안, BcN QoS & Security, 웹 모델링 및 대응 기술, 통신망 성능분석

전 용 희 (Yong-Hee Jeon) ??회원



1971년 3월~1978년 2월 고려대학교 전기전자전파공학부
1985년 8월~1987년 8월 미국 플로리다공대 대학원 컴퓨터공학과
1987년 8월~1992년 12월 미국 노스캐롤라이나주립대 대학원 Elec. and Comp. Eng. MS, Ph. D.

1978년 1월~1978년 11월 삼성중공업(주)
1978년 11월~1985년 7월 한국전력기술(주)
1979년 6월~1980년 6월 벨기에 벨기토퍼사 연수
1989년 1월~1989년 6월 미국 노스캐롤라이나주립대 Dept of Elec. and Comp. Eng. TA
1989년 7월~1992년 9월 미국 노스캐롤라이나주립대 부설 CCSP (Center For Comm. & Signal Processing) RA
1992년 10월~1994년 2월 한국전자통신연구원 광대역 통신망연구부 선임연구원
1994년 3월~현재 대구가톨릭대학교 컴퓨터·정보통신공학과 교수

2001년 3월~2003년 2월 대구가톨릭대학교 공과대학 장 역임

2004년 2월~2005년 2월 한국전자통신연구원 정보보호연구단 초빙연구원
<관심분야> 네트워크 보안, BcN QoS & Security, 웹 모델링 및 대응 기술, 통신망 성능분석

송 동 호 (Dong-Ho Song) ??회원



1980년 3월~1984년 2월 경북대 전자공학과 학사
1984년 3월~1988년 2월 KAIST 전기 및 전자공학 석사
1988년 3월~1991년 3월 영국 Newcastle 대학교 전산학 박사
1988년 3월~1995년 12월 ETRI 컴퓨터 연구단 분산 멀티미디어 연구실장

1992년 1월~1993년 12월 SRI International Research Fellow
1994년 1월~1995년 12월 한국통신 중장기 과제 심사 위원
1994년 1월~현재 미국 Distributed Multimedia Systems Workshop Program Committee
1995년 1월~현재 미국 IEEE Multimedia Magazine Program Committee
1995년 1월~현재 한국항공대 컴퓨터공학과 부교수
1999년 1월~현재 소프트온넷(주) 대표이사