

소프트웨어 온디맨드 스트리밍 시스템을 위한 성능평가 도구 설계

김영만^{0*}, 허성진^{**}, 최완^{**}, 한왕원^{*}

*국민대학교 컴퓨터학부, **한국전자통신연구원 디지털홈 연구단

{ymkim⁰, wwhan}@kookmin.ac.kr, **{sjheo, wchoi}@etri.re.kr

Design of Performance Analyzing Tool for Software On-Demand Streaming System

Young Man Kim*, Seongjin Heo**, Wan Choi**, Wang Won Han*

*School of Computer Science, Kookmin University, **Digital Home Division, ETRI

요 약

소프트웨어 스트리밍[1]은 프로그램 설치 및 실행을 위하여 서버로부터 데이터 전송이 진행 중인 동안에도 PC, PDA, 휴대용 개인 컴퓨터, 휴대 전화 등과 같은 데스크 탑 또는 이동 연산장치 상에서 미설치 소프트웨어의 실행이 즉각적으로 이루어지도록 하는 기능이다. 즉, 소프트웨어 스트리밍 기술을 사용하면 사용자가 다운로드, 압축해제, 인스톨과 시스템 재구성이라는 일련의 과정이 완료될 때까지 기다릴 필요 없이 최소한의 다운로드 후에 해당 소프트웨어가 곧바로 실행될 수 있도록 해준다. 응용프로그램을 실행하는데 필요한 첫번째 실행 블록이 메모리에 적재되고 기본적인 환경설정을 마치자마자 나머지 블록들이 다운로드되고 설치되기도 전에 실행될 수 있기 때문에 스트리밍 시스템은 응용 프로그램의 실행준비 시간을 대폭 줄일 수 있게 해준다. 게다가, 응용프로그램 실행시 실제로 사용되지 않는 대부분의 블록들은 서버로부터 다운로드 받지 않아도 된다. 그 결과, 메모리와 대역폭 같은 리소스의 활용이 절약된다. 이러한 스트리밍 시스템을 사용하면 사용자는 다양한 공개 또는 상업용 응용프로그램을 광범위하게 지원하는 사용자 투명성을 가진 가상 소프트웨어 컴퓨팅 환경을 만들 수 있다. 본 논문에서는 리눅스 환경에서 설계 구현된 Software On-Demand(SOD) 스트리밍 시스템을 소개하고, SOD 스트리밍 시스템의 성능측정을 위한 소프트웨어를 설계한다.

1. 서 론

소프트웨어 온디맨드 서비스/시스템 시장은 현재 55 억달러에 이르고 있으며 2006 년도에는 200 억달러에 이르리라고 전망되는 최신기술로 자리매김하고 있다. 아직까지는 웹 기반 ASP 방식이 시장의 주류를 형성하고 있으며 그 다음으로 서버 기반 ASP 방식이 사용되고 있으나 기술상의 몇가지 문제점들로 인하여 스트리밍 기반의 온디맨드 기술이 새로 등장하고 있다.

소프트웨어 및 콘텐츠를 동시에 서비스하는 시장만을 보아도 고부가 가치 시장이며, 단단계 형태의 소프트웨어 유통시장까지 고려했을 때 방대한 규모를 가지고 있는데 온디맨드 기술의 차세대판인 스트리밍 방식이 시장을 주도할 것으로 전망되고 있다.

스트리밍 기술을 확보한 국내의 대표적인 업체들로 소프트온넷과 한컴을 들 수 있는데, 소프트온넷은 현

재 국내 100 여개 대학과 30 여개 기업에서 내부인트라넷을 통하여 윈도우즈 응용프로그램들에 대하여 스트리밍 서비스를 제공하고 있으나 B2C 형태의 서비스를 위하여 기술축적을 해가고 있는 상태이며 한컴의 경우 2003 년말 미국의 ThinkFree 사를 인수하여 자바형 오피스를 ASP 할 수 있는 체제를 구축하고 있다. 그 외의 국내 온디맨드 스트리밍 회사는 외국 제품을 수입하여 영업하는 형태를 취하고 있다[1].

한편 대표적인 국내 연구기관인 ETRI 에서 주도하여 개발한 리눅스 SOD 엔진은 비효율적인 구조상의 문제점을 안고 있어 성능 개선의 여지가 많이 남아있다. 따라서 SOD 엔진의 문제점들을 구체적으로 도출하고 높은 성능향상을 가져오기 위하여 SOD 엔진에 대한 정성적이고 분석적인 성능평가 도구가 필요한 상황이다.

SOD 성능측정 소프트웨어를 개발하기 위한 환경으로 SOD 스트리밍 시스템을 지원하는 스트림 서버인 Z!Stream [2]서버를 사용한다. Z!Stream 서버는 마운

팅, 인증, 감사작업, 접속 관리, 세션 관리, 부하 조절 등과 같은 스트리밍과 관련된 풍부한 함수 집합을 제공한다.

본 논문의 목표는 SOD 엔진의 성능평가를 위한 도구를 설계하는 것이며, 차후에 설계한 성능평가 도구를 구현하고 SOD 엔진을 실제 평가 분석함으로써 성능저하 원인을 찾아내어 SOD 시스템의 성능향상 방안들을 제안하는데 있다.

2. Software On-Demand (SOD) Streaming System 설계 및 구현

이상적인 응용프로그램 컴퓨팅 환경에서는 사용자가 다운로드, 인스톨 등과 같은 부가적인 업무에 시간과 노력을 투입하지 않아도 본 업무를 처리할 수 있도록 부수적인 작업을 자동으로 처리하는 기능을 지원해준다. 사용자가 문제를 해결하기 위해 특정 응용 프로그램을 실행하고자 할 때 데스크 탑 상에서 해당 응용 프로그램을 상징하는 아이콘을 클릭하면 프로그램 초기 실행에 필요한 모듈들이 메모리에 로딩되자마자 프로그램의 UI 윈도우가 화면에 나타나게 되며 이 프로그램이 최초로 호출되는 경우에는 추가적으로 프로그램 다운로드와 인스톨 절차가 백그라운드에서 진행된다.

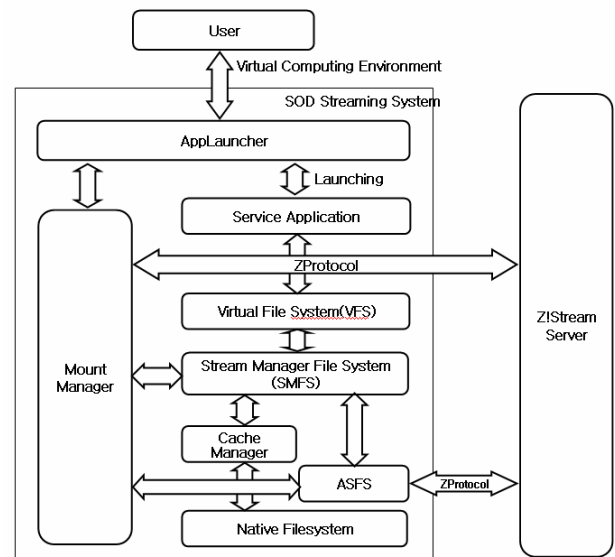
이번 장에서는 Software On-Demand(SOD) Streaming System이라 불리는 스트리밍 시스템을 제안한다. 제안된 SOD 시스템은 다음과 같은 특징을 가진다. 첫째로, SOD 시스템은 복잡한 인스톨 작업으로부터 사용자의 수고를 덜어준다. 두 번째로, 소프트웨어 개발자에게 새로운 제품의 광고와 유통을 위한 강력한 수단을 제공한다. 세 번째로 사용자는 새로운 프로그램을 실행하는 경우에 UI 윈도우의 빠른 팝업을 경험하게 된다. 네 번째로 SOD 스트리밍은 프로그램이 사용하는 데이터 파일을 모두 다운로드하지 않고도 응용 프로그램이 데이터 파일을 액세스할 수 있도록 해준다. 마지막으로 SOD 스트리밍은 서버에서 직접 스트리밍되기 때문에 소프트웨어 배포와 업데이트를 용이하게 한다.

2.1 SOD Streaming System Architecture

[그림 1]은 SOD 스트리밍 시스템의 구조를 보여주고 있으며 가상 컴퓨팅 환경을 만들어 다운로드와 인스톨 절차를 사용자로부터 은폐시킨다. 가상 컴퓨팅 환경의 주된 목적은 각 응용 프로그램 소프트웨어를 위해 응용 프로그램에 종속적인 프로그램 레지스트리, 환경 변수, 구성 파일, 관련 컴포넌트에 대한 컴퓨팅 환경을 사용자가 인식하지 않는 상태에서 자동적으로 설정·구축하는데 있다.

사용자가 새로운 응용 프로그램을 최초로 선택하면 SOD 시스템은 환경 관련 데이터를 스트리밍 서버로부터 수신한다. 환경 설정이 끝나게 되면 SOD 시스템은

서버에 프로그램 실행을 위한 최소한의 바이너리 실행 페이지들을 요청한다.



[그림 1] SOD 스트리밍 시스템

SOD 사용자 인터페이스 프로그램(AppLauncher)은 사용자에게 SOD를 통하여 실행가능한 소프트웨어 아이콘들을 보여준다. 사용자가 AppLauncher 윈도우에서 임의의 아이콘을 선택하면 SOD 시스템 컴포넌트 중의 하나인 Mount Manager(MM)은 응용 프로그램 환경 데이터를 다운로드하고 인스톨하는 작업을 수행한다. MM은 이러한 작업을 위해 Z 프로토콜을 이용하여 해당 데이터를 스트림 서버에 요청한다. 응용 프로그램 컴퓨팅 환경에서 응용프로그램을 위한 초기 설정이 끝나면 AppLauncher는 응용 프로세스를 발진하기 위하여 실행 이미지 블록들을 OS에게 요청한다. 이때 프로그램 이미지와 데이터 파일을 위한 I/O가 발생하고 해당 요청은 Linux Virtual File System(VFS) 모듈을 경유하여 SOD 시스템의 세번째 컴포넌트인 Stream Manager File System (SMFS)에 도착한다. SMFS는 우선 SOD 시스템의 네번째 컴포넌트인 Cache Manager(CM)를 경유하여 로컬 디스크 캐시를 검색한다. 만일 페이지가 로컬 캐시에서 발견되지 않으면 SMFS는 스트리밍 서버에게 해당 페이지 요구 메시지를 보내기 위해 다섯번째 컴포넌트인 Application Streaming File System(ASFS)를 호출하며 ASFS는 Z 프로토콜을 통해 서버로부터 해당 페이지를 전송받는다. ASFS를 통하여 입수된 페이지는 재사용을 위해 로컬 디스크 캐시에 저장되고 동일한 내용이 CM에 생성된 후 프로그램 실행이 중지된 위치에서 재개된다. 현재 SOD 시스템은 리눅스 상에서 구현 완료된 상태에 있다.

2.2 SOD System Modules

이 절에서는 본 시스템을 구성하는 SOD 각 모듈에 대하여 자세히 설명한다.

2.2.1 AppLauncher

AppLauncher는 사용자가 원하는 응용 프로그램을 손쉽게 찾아내고 선택할 수 있도록 웹 브라우저와 유사한 유저 인터페이스를 제공한다. 사용자가 실행하고자 하는 응용 소프트웨어를 상징하는 아이콘을 클릭할 때, AppLauncher는 Mount Manager에게 응용 소프트웨어 실행을 위한 가상 컴퓨팅 환경을 확립하도록 지시한다. 그 후 fork()와 exec() 시스템 호출에 의해 응용 프로그램 프로세스를 개시한다.

2.2.2 Mount Manager (MM)

리눅스는 다중 사용자용 실행 환경을 제공한다. 그러므로, Mount Manager 역시 다중 AppLauncher 프로세스 정보를 유지해야 한다. 각 AppLauncher가 개별 응용 프로그램 컴퓨팅 환경을 생성하도록 MM에게 지시하면 MM은 로컬 캐시로부터 상응하는 환경 데이터를 검색한다. 만일 데이터가 캐시에서 발견되지 않으면, MM은 스트리밍 서버에게 환경 데이터 요구 메시지를 보낸다. MM은 도착된 환경 데이터를 사용하여 가상 컴퓨팅 환경을 구축하고 관리한다.

2.2.3 Stream Manager File System (SMFS)

SMFS는 리눅스 VFS와 네이티브 파일 시스템(예를 들면, ext2) 사이에 위치하는 SOD전용 가상 파일 시스템 모듈이다. 응용 프로그램 프로세스 혹은 MM이 프로그램을 시작하거나 데이터 페이지를 요구할 때 해당 호출은 리눅스 VFS를 경유하여 SMFS에 도착한다. SMFS는 우선 Cache Manager를 호출하여 로컬 캐시내에서 해당 페이지를 찾는다. CM은 페이지의 내용 혹은 미발견 오류정보를 SMFS에게 리턴한다. 후자의 경우 SMFS는 해당 페이지 데이터를 얻기 위해 ASFS를 통하여 스트리밍 서버에게 요구하며 서버로부터 새로운 페이지 데이터가 도착하면 ASFS를 통하여 SMFS에게 배달된다. 그 후 해당 페이지는 CM을 경유하여 로컬 디스크 캐시에 저장되고, 메모리에도 로딩된 후에 응용 프로그램 프로세스 혹은 MM의 실행이 재개된다.

2.2.4 Cache Manager (CM)

데이터 재사용성을 증진시키기 위해 로컬 디스크 캐시에 저장되는 데이터는 응용 프로그램 스트리밍 서비스가 보다 신속히 진행될 수 있도록 도와준다. 즉, 응용 프로그램 프로세스는 네트워크 라운드 트립 지연을 생략함으로써 사용자에게 신속한 프로그램 실행을 제공하며 귀중한 네트워크 자원을 절약한다. CM은 빠른 페이지 검색을 위해 내부적으로 인덱스 구조를 가지고 있다.

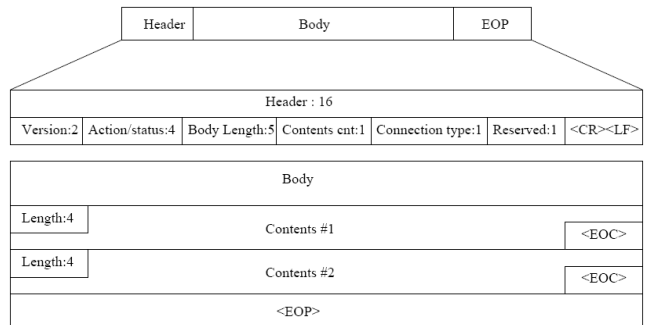
2.2.5 Application Streaming File System (ASFS)

ASFS는 데이터 페이지 요구를 Z 프로토콜 메시지에 실어서 서버로 보내주며, 서버로부터 도착한 메시지 내의 페이지 데이터를 SMFS에 전달한다.

2.3 Z Protocol

Z 프로토콜은 SOD 스트리밍을 위해 설계된 응용 프로토콜이다. 실시간 QoS 필요조건을 만족시켜야만 하는 멀티미디어 스트리밍과는 대조적으로 SOD 스트리밍은 신뢰할 수 있는 메시지 전달을 필요로 한다. 따라서 Z 프로토콜은 TCP 트랜스포트 프로토콜 상위에 구축된다. 메시지에는 제어와 데이터 메시지 2가지 종류가 있다. 제어(데이터) 메시지는 RASCP(RASP) 보조 프로토콜에서 정의되고, MM(ASFS)과 스트리밍 서버 사이에서 사용된다. MM은 RASCP를 경유하여 환경 데이터, 인증, 접속 관리, 세션 관리 등의 제어 정보를 송신한다. 반면에 ASFS는 프로그램 또는 데이터 페이지에 RASP를 사용하여 요구 메시지를 보낸다.

RASCP와 RASP 모두 공통의 요구-응답 통신 패러다임을 따르므로 공통 Z 프로토콜 헤더 포맷을 공유한다. Z 프로토콜의 포맷은 [그림 2]와 같이 구성된다.



[그림 2] Z 프로토콜

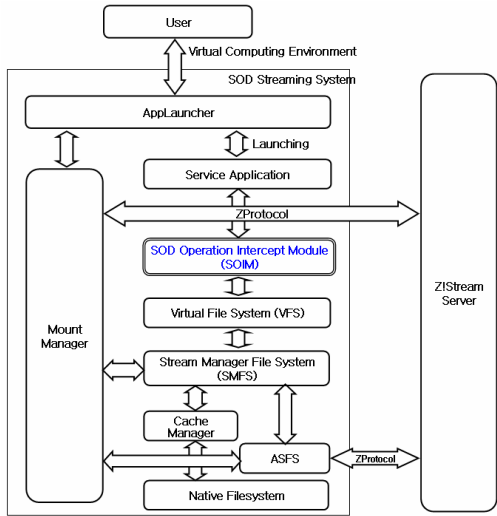
3. SOD 스트리밍 시스템을 위한 성능평가 소프트웨어 설계

이번 절에서는 앞에서 설명한 SOD시스템의 성능을 평가하는 소프트웨어를 설계하고자 한다. 우선 소프트웨어를 설계하기에 앞서 SOD 시스템에 적용할 수 있는 성능평가 지표는 다음과 같은 것들이 있을 수 있다.

- 새로운 소프트웨어 실행시 화면 Pop-up 시간
- 재부팅 후 화면 Pop-up 시간
- 가상 파일 시스템 마운트 시간
- 파일의 신규 페이지를 읽고 쓰는 시간
- 파일의 페이지를 읽고 쓰는 시간

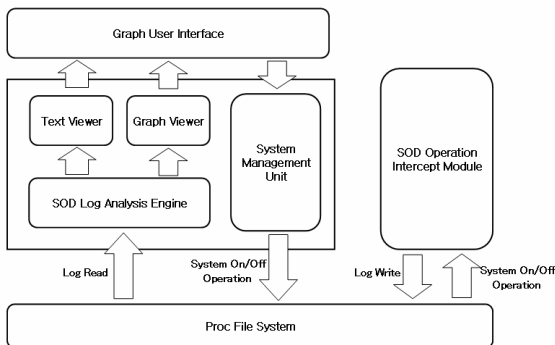
성능평가 소프트웨어는 위와 같은 지표들을 올바르게 측정하고 표현하기 위한 방법으로 다음의 시간들을 측정한다. SOD 응용프로그램의 초기실행시의 Application Load Time(ALT), SOD 응용프로그램의 재

실행시 ALT, 응용 프로그램이 현재위치에 이미 설치완료된 상태에서의 Local 실행시 ALT, Local 재실행시 ALT, Page Read-access Time(PRT), File Read-access Time(FRT)를 측정한다.



[그림 3] SOD 성능평가 모듈

SOD 성능평가 소프트웨어의 핵심부분인 SOD Operation Intercept Module(SOIM)은 SOD 응용 프로그램들이 SMFS를 사용하기 위해 호출하는 시스템 콜을 가로채고 연산에 대한 로그를 기록하는 역할을 담당한다. SOIM은 커널 모듈로 구현되어지는데 커널 모듈이란 필요에 따라 커널에 로드하거나 언로드 할 수 있는 특정한 기능을 수행하는 코드이다. 이렇게 함으로써 쉽게 SOIM을 커널의 기능으로써 확장할 수 있을 뿐만 아니라 운영체제를 다시 부팅 하지 않고도 로그 기능을 수행할 수 있도록 만들 수 있다. [그림 3]에 나타난 것과 같이 SOD과 관련된 시스템 콜들이 VFS에 진입하기 전에 SOIM 모듈이 시스템 콜을 가로챈다. 시스템 콜을 가로챈 SOIM 모듈은 연산에 대한 로그를 기록하고 SOD 연산이 제대로 실행될 수 있도록 VFS에 원래의 시스템 콜을 전달한다.



[그림 4] SOD 성능평가 소프트웨어 구조

[그림 4]는 SOD 성능평가 소프트웨어의 전체 구조를 나타내고 있다. SOD 성능평가 소프트웨어는 크게 SOD 응용 프로그램의 연산을 가로채어 Proc 파일시스템에 로그를 기록하는 SOIM, Proc 파일시스템에 기록된 SOD 연산의 로그를 분석하기 위한 SOD Log 분석 엔진, 성능평가 소프트웨어의 On/Off와 같이 시스템과

관련된 기능들을 수행하고 관리하는 System Management Unit 그리고 사용자에게 보여지는 GUI로 이루어진다.

SOD 성능평가 소프트웨어의 실행흐름은 사용자가 성능평가 소프트웨어의 GUI를 사용하여 로그기록의 시작을 요구하면 System Management Unit은 SOIM에게 로그를 기록하도록 시킨다. 로그기록을 지시받은 SOIM은 SOD 응용 프로그램의 파일시스템 관련 연산이 요청되어 질 때 마다 형식에 맞추어 proc 파일 시스템에 로그를 기록한다. 기록되어진 로그는 SOD 로그 분석 엔진에 의해 읽혀져서 분석된다. SOD 로그는 연산을 요청한 시간, 연산이 완료된 시간, 연산을 요구한 프로세스의 ID, 연산의 종류, 요구된 블록의 크기 그리고 연산의 수행방향을 기록한다. 연산의 수행방향이란 SOD 시스템에서는 파일에서 하나의 페이지를 읽어오는 두가지 다른 위치가 존재하는데 SOD연산이 실제로 실행되는 위치를 말한다. SOD 시스템에서 파일을 읽어오는 위치로는 첫째 클라이언트의 로컬 파일시스템에 존재하는 디스크 캐쉬에서 원하는 파일의 페이지를 읽어올 수 있다. 나머지 위치는 네트워크를 통해 SOD서버로부터 원하는 파일의 일부분을 얻어 올 수 있는 것이다. Proc 파일 시스템에 기록된 로그는 사용자에게 보여지기 전에 SOD log 분석 엔진에 의해서 로그의 내용이 분석되고, 이렇게 분석된 로그는 텍스트 뷰어나 그래픽 뷰어에 의해 사용자의 GUI에 보여지게 된다.

4. 결론

본 논문에서는 프로그램 등록, 환경 변수 설정, 구성 파일 및 관련 컴포넌트의 자동 인스톨 기능을 자동적으로 실행함으로써 응용 프로그램의 신속한 스트림 실행을 제공하는 Software On-Demand Streaming System (SOD)을 소개하고, SOD 시스템 성능분석의 기준이 되는 평가 지표와 그 측정방법을 제안하였다. 또한 제시된 지표들을 실제로 측정하기 위한 성능평가 소프트웨어를 설계해 보았다. 앞으로 우리는 본 논문에서 설계한 SOD 성능평가 소프트웨어를 구현할 것이다. SOD시스템의 성능에 영향을 미치는 파라미터를 발견하기 위해 구현된 소프트웨어로 SOD시스템에 다양한 파라미터를 적용하여 어떤 경우에 SOD시스템이 최적화 될 수 있는지를 실험할 것이다.

참고문헌

- [1] Young Man Kim, "온디맨드 S/W 스트리밍 서비스 성능 측정 및 개선에 관한 연구" ETRI 중간 연구보고서, 2005년 9월.
- [2] SOFTonNET Inc., "Z!Stream Technology," <http://www.softonnet.com>.