

소프트웨어 온디맨드 스트리밍 시스템 성능평가

김영만*, 허성진**, 최완**, 한왕원*

*국민대학교 컴퓨터학부, **한국전자통신연구원 디지털홈 연구단

Performance Evaluation for Software

On-Demand Streaming System

YoungMan Kim*, SeongJin Heo**, Wan Choi**, Wangwon Han*

*Kookmin University, **Electronics and Telecommunications Research Institute(ETRI)

E-mail : *{ymkim, wwhan}@kookmin.ac.kr, **{sjheo, wchoi}@etri.re.kr

요 약

소프트웨어 스트리밍[1][2]은 프로그램 설치 및 실행을 위하여 서버로부터 데이터 전송이 진행중인 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. 이러한 스트리밍 시스템을 사용하면 사용자는 다양한 공개 또는 상업용 응용프로그램을 광범위하게 지원 하는 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 성능평가 소프트웨어를 사용하여 SOD 스트리밍 시스템의 성능을 측정한다.

1. 서론

PC, PDA, 휴대용 개인 컴퓨터, 휴대 전화 등과 같이 데스크 탑 또는 이동식의 연산 장치에서 서버로부터 응용프로그램을 다운로드하고 설치하는 작업은 많은 시간이 소모되며, Windows와 Linux와 같은 대표적인 컴퓨팅 환경에서 공개 혹은 상업용 응용프로그램의 설치에 초보 사용자가 이해하기

힘든 지식을 요구한다. 그러나 이러한 문제는 서버로부터 응용프로그램의 전송 및 설치가 진행 중인 동안에도 해당 응용 프로그램이 실행될 수 있도록 해주는 소프트웨어 스트리밍 기술을 통해 해결 가능하다. 또한 소프트웨어 스트리밍은 응용프로그램의 실행에 불필요한 대부분의 블록들을 서버로부터 다운받지 않아도 되기 때문에 메모리와

대역폭 같은 리소스를 매우 효율적으로 활용할 수 있게 해준다.

본 논문에서는 프로그램 등록, 환경 변수 설정, 그리고 구성 파일과 관련된 컴포넌트들을 자동 설치하는 작업들을 진행하는 동안에도 프로그램 실행에 필요한 최소한의 준비가 갖추어지는 순간 프로그램의 즉시 실행 서비스를 제공하는 Software On-Demand(SOD) 스트리밍 시스템을 소개하고, SOD 스트리밍 시스템의 성능평가 소프트웨어를 사용하여 SOD 스트리밍 시스템의 성능을 측정 후 성능개선 방법을 제안한다. SOD 시스템은 직관적인 look-and-click 방식의 가상 응용 프로그램 컴퓨팅 환경을 사용자에게 제공하며, 소프트웨어 다운로드와 인스톨 절차 및 페이지 요구작업이 백그라운드에서 자동적으로 진행되도록 리눅스 환경에서 설계 및 구현되었다.

SOD 성능측정 소프트웨어를 개발하기 위한 환경으로 SOD 스트리밍 시스템을 지원하는 스트림 서버인 Z!Stream [3]서버를 사용한다. Z!Stream 서버는 마운팅, 인증, 감사작업, 접속 관리, 세션 관리, 부하 조절 등과 같은 스트리밍과 관련된 풍부한 함수 집합을 제공한다.

2. Software On-Demand (SOD) 스트리밍 시스템 소개

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 응용프로그램을 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.

본 논문에서는 이와 같은 기능을 지닌 시스템을 Software On-Demand(SOD) 스트리밍 시스템이라 부르기로 한다. 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.

3. SOD 스트리밍 시스템 성능평가

이 장에서는 2장에서 소개한 SOD 스트리밍 시스템에 대하여 성능 평가한다. SOD 스트리밍 시스템 사용자의 만족도를 측정하는 성능평가 실험을 행하고 그 결과분석을 통해 사용자에게 가상 컴퓨팅 환경에서 보다 높은 서비스 만족도를 제공할 수 있는 성능개선 방법에 대하여 제안한다.

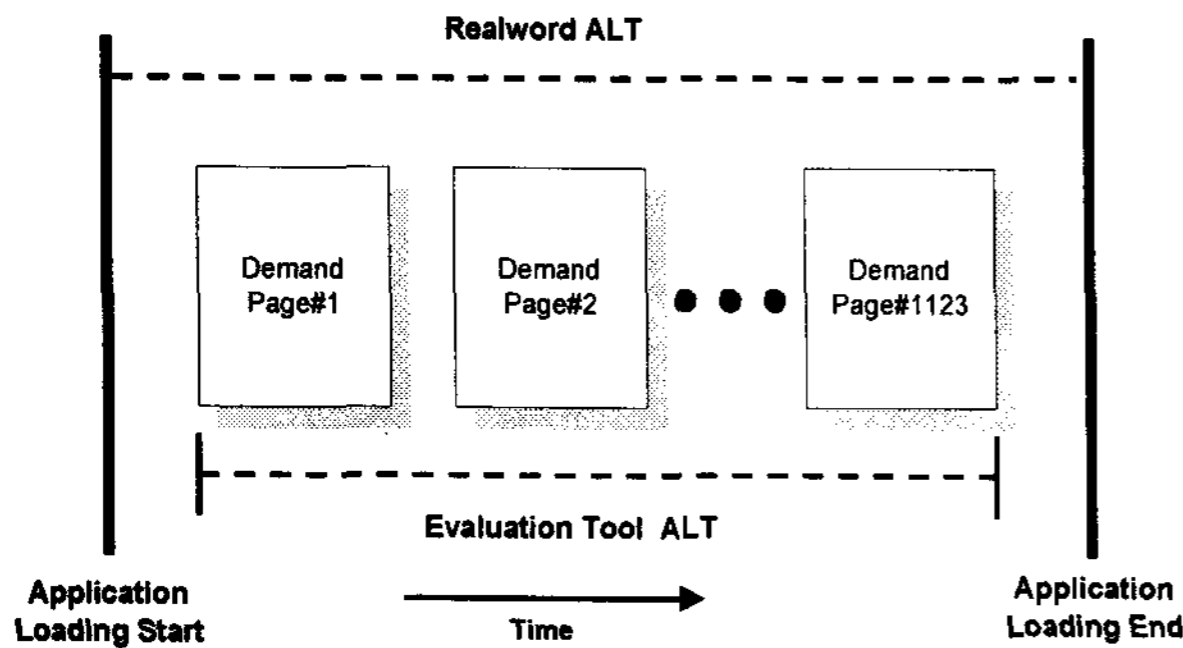
3.1 성능평가 지수

대화형 응용 프로그램 환경에서 신속한 응용 프로그램 실행 속도를 나타내는 성능평가 지수로서 사용자 인터페이스 화면에 관련된 Application Load Time(ALT), Page Load Time(PLT) 그리고 Initial Execution Loading Size(IELS)를 선정하였다.

ALT는 임의의 프로그램이 마우스에 의하여 실행 선택된 직후부터 초기 사용자 인터페이스 윈도우가 화면에 나타날 때까지의 경과시간이다. ALT는 환경 데이터의 크기, 프로그램의 초기화 또는 런칭에 관련된 데이터 크기에 직접적으로 비례하고 전송 매체의 속도에 반비례한다. 따라서 낮은 ALT

값은 프로그램이 빨리 실행된다는 것을 뜻한다.

실세계에서의 ALT는 프로그램이 선택된 직후부터 사용자 윈도우가 화면에 나타날 때까지의 절대적인 시간이지만, 성능평가 도구에서 계산되어지는 ALT는 실세계의 ALT와는 약간의 차이가 있다. [그림1]에서 보이는 바와 같이 성능평가 도구의 ALT는 첫 번째 페이지가 요구된 시간부터 사용자 윈도우가 화면에 나타나기 직전에 요구된 페이지가 처리될 때까지의 시간을 나타내므로 실세계의 ALT 보다 약간 적은 시간이 된다.

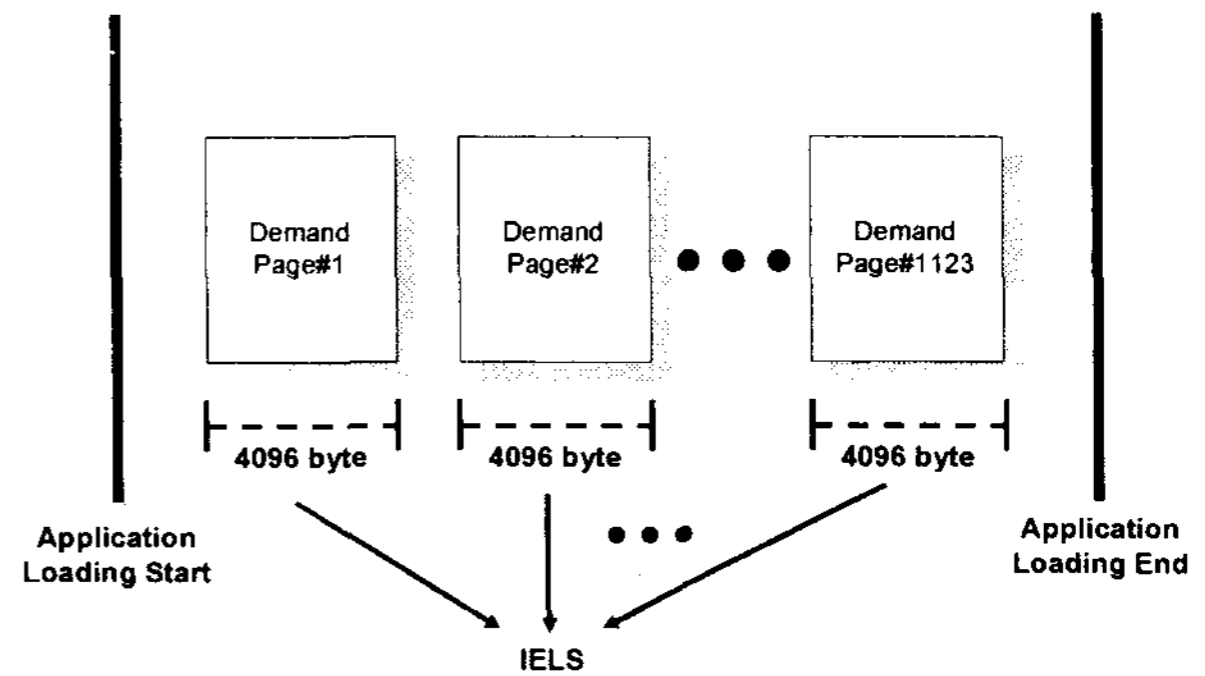


[그림 1] 성능평가 도구 ALT

PLT는 임의의 프로그램에 의해서 요구된 페이지가 저장매체 혹은 네트워크로부터 읽혀지고 처리되는 시간을 나타낸다. ALT 동안에 요구된 페이지들에 대한 각각의 PLT가 로그파일에 기록되고, PLT의 평균을 계산함으로써 프로그램의 PLT가 결정되어진다. [그림1]를 예로 들면 ALT동안 1123번의 페이지 요구가 있었고, 1123번의 각 페이지 요구에 대해 PLT가 기록되고 프로그램에 대한 PLT는 1123개 PLT의 평균을 계산하므로써 구할 수 있다. 낮은 PLT는 결과적으로 ALT를 작아지게 하므로 PLT가 낮을수록 프로그램은 빨리 실행된다는 것을 뜻한다.

마지막 성능평가 지수인 IELS는 ALT동안에 저장매체 혹은 네트워크로부터 읽혀진 총 바이트 수를 의미한다. [그림2]에서 보는 바와 같이 페이지 요구시 읽혀지는 바이트수를 모두 더해서 ALT동안의 IELS를 결정한다. IELS가 크다는 것은 프로그램이

실행되기 위해 읽혀져야 하는 바이트수가 크다는 것을 의미하므로 IELS가 크면 클수록 프로그램은 늦게 실행될 것이다.

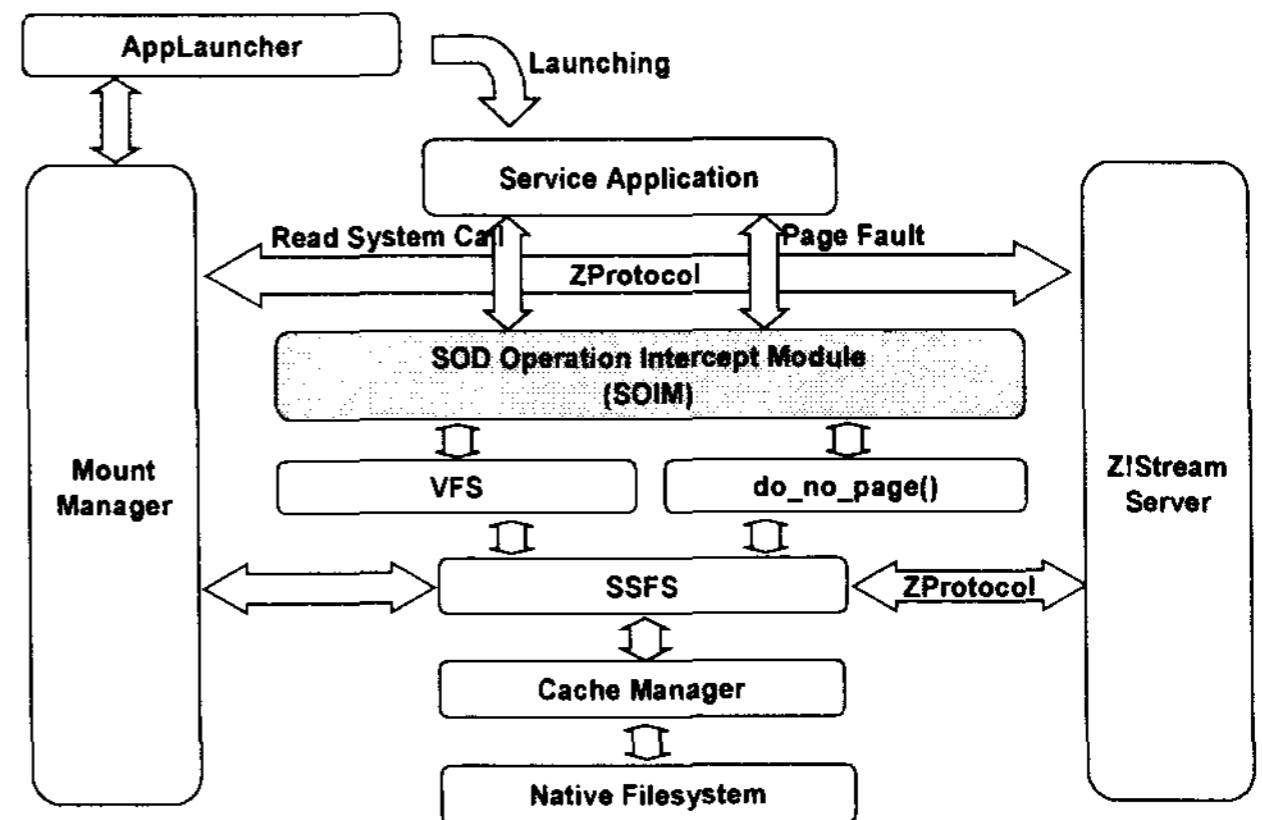


[그림 2] 성능평가 도구 IELS

3.2 성능평가 모듈설계 및 구현

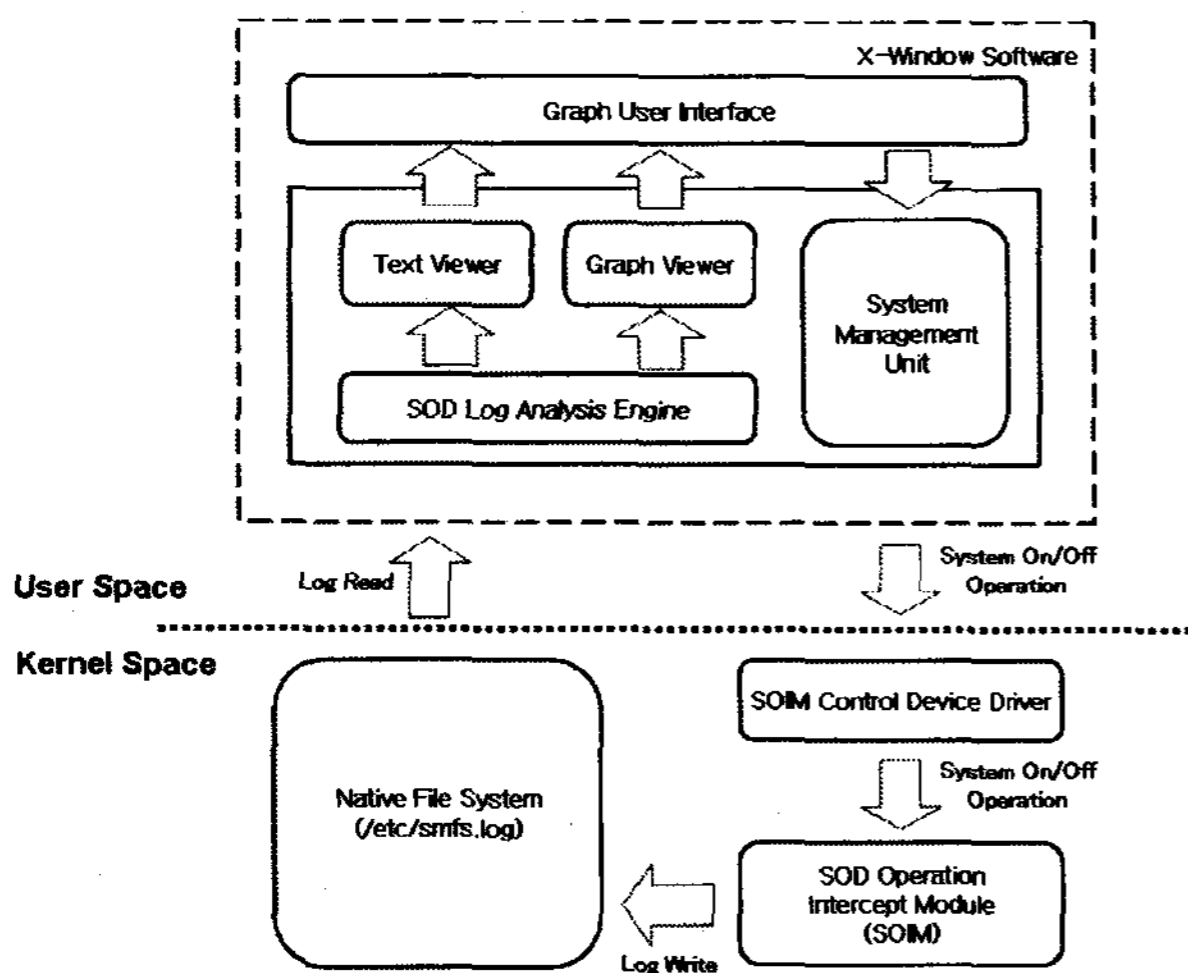
이번 절에서는 앞에서 설명한 SOD 시스템의 성능을 평가하는 성능평가용 소프트웨어를 설계하고 실제 구현이 어떻게 이루어졌는가를 설명한다.

SOD 성능평가 소프트웨어의 핵심부분인 SOD Operation Intercept Module(SOIM)은 SOD 응용 프로그램들이 요구 페이지 검색 모듈 SSFS를 사용하기 위해 호출하는 시스템 콜을 가로채고 연산에 대한 로그를 기록하는 역할을 담당한다. SOIM은 파일시스템에 대한 읽기 연산을 수행하는 시스템 콜과 페이지 폴트 발생시 페이지 폴트를 처리하는 시스템 콜에 대한 로그를 기록함으로써, 각 성능



[그림 3] SOD 성능평가 모듈

평가 지수를 측정할 수 있도록 도와준다. 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 SOD와 관련된 시스템 콜들은 운영체제에 의하여 처리되기 전에 SOIM 모듈이 가로채어 진다. 시스템 콜을 가로챈 SOIM 모듈은 연산에 대한 로그를 기록하고 SOD 연산이 제대로 실행될 수 있도록 원래 시스템 콜이 처리되는 부분으로 시스템 콜을 전달한다.



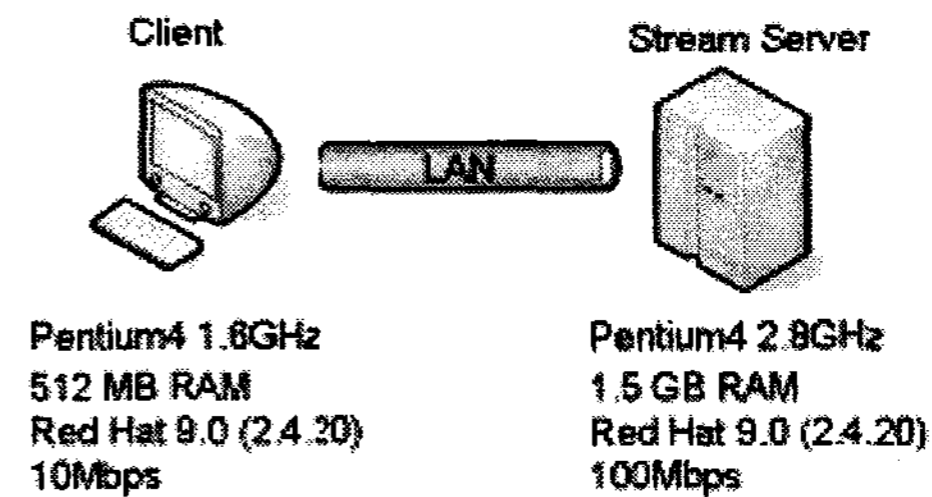
[그림 4] SOD 성능평가 소프트웨어 구조

[그림 4]은 SOD 성능평가 소프트웨어의 전체 구조를 나타내고 있다. SOD 성능평가 소프트웨어는 크게 SOD 응용 프로그램의 연산을 가로채어 파일에 로그를 기록하는 SOIM, 파일에 기록된 SOD 연산의 로그를 분석하기 위한 SODLog 분석 엔진, 성능평가 소프트웨어의 On/Off와 같이 시스템과 관련된 기능들을 수행하고 관리하는 System Management Unit 그리고 성능평가 담당자에게 평가결과를 보여주는 GUI로 이루어진다.

SOD 성능평가 소프트웨어의 실행흐름은 사용자가 성능평가 소프트웨어의 GUI를 사용하여 로그기

록의 시작을 요구하면 System Management Unit은 커널 디바이스 드라이버로 구현된 SOIM Control Device Driver를 통해 SOIM에게 로그를 기록하도록 지시한다. 로그기록을 지시받은 SOIM은 SOD 응용 프로그램의 파일시스템 관련 연산이 요청될 질 때 마다 형식에 맞추어 로그파일에 로그를 기록한다. 이렇게 기록되어진 로그는 SOD 로그분석 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 시스템에서 파일 페이지를 실제로 가져오는 위치를 말한다. SOD 시스템에서 파일을 읽어오는 위치로는 세군데가 있는데, 첫째 네트워크를 통해 SOD 서버로부터 원하는 파일의 일부분을 얻어 올 수 있는 것이다. 두 번째는 클라이언트의 로컬 파일시스템에 존재하는 디스크 캐시에서 원하는 파일의 페이지를 읽어올 수 있다. 나머지 위치는 커널에서 제공하는 메모리 캐시에서 파일의 페이지를 읽어오는 것이다. 로그파일에 기록된 로그는 텍스트 뷰어나 그래픽 뷰어에 의해 사용자의 GUI에 보여지게 된다.

3.3 성능평가를 위한 실험환경



[그림 5] SOD 성능평가 실험환경

[그림 5]은 SOD 시스템의 성능평가를 위한 실험환경을 나타내고 있는 그림이다. 서버와 클라이언트는 랜을 통해서 연결되어 있고 하드웨어 사양은 그림에 나타난바 와 같으며, 본 논문에서는 좀 더 정확한 데이터를 얻기 위해 Abiword, Volley,

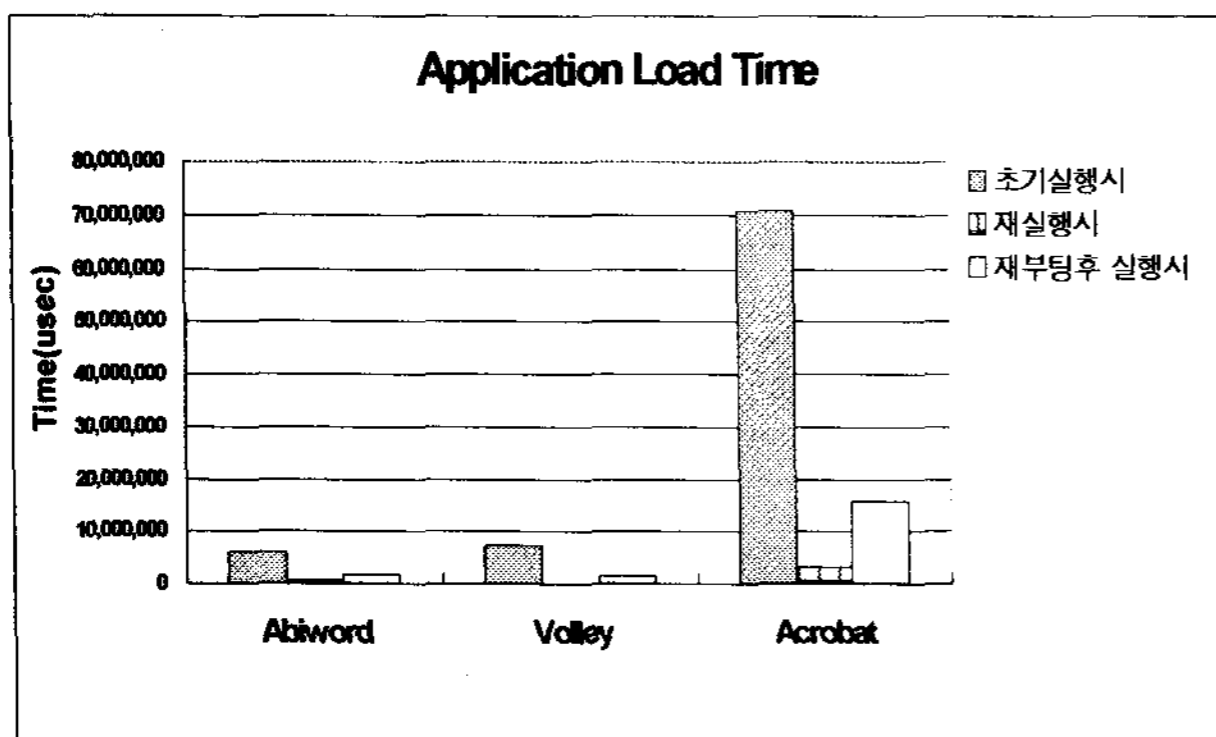
Acrobat 세가지 프로그램의 ALT, PLT 그리고 IELS 를 여러번 측정한 결과들의 평균값을 각 프로그램의 최종 값으로 선택하였다. SOD 성능평가 소프트웨어에 의해서 성능평가 되어지는 프로그램의 설치용량은 [표 1]과 같다.

	Acrobat	Volley	Abiword
전체용량	151.78MB	7.37MB	30.12MB

[표 1] 성능평가되는 프로그램의 설치용량

3.4 실험결과 및 분석

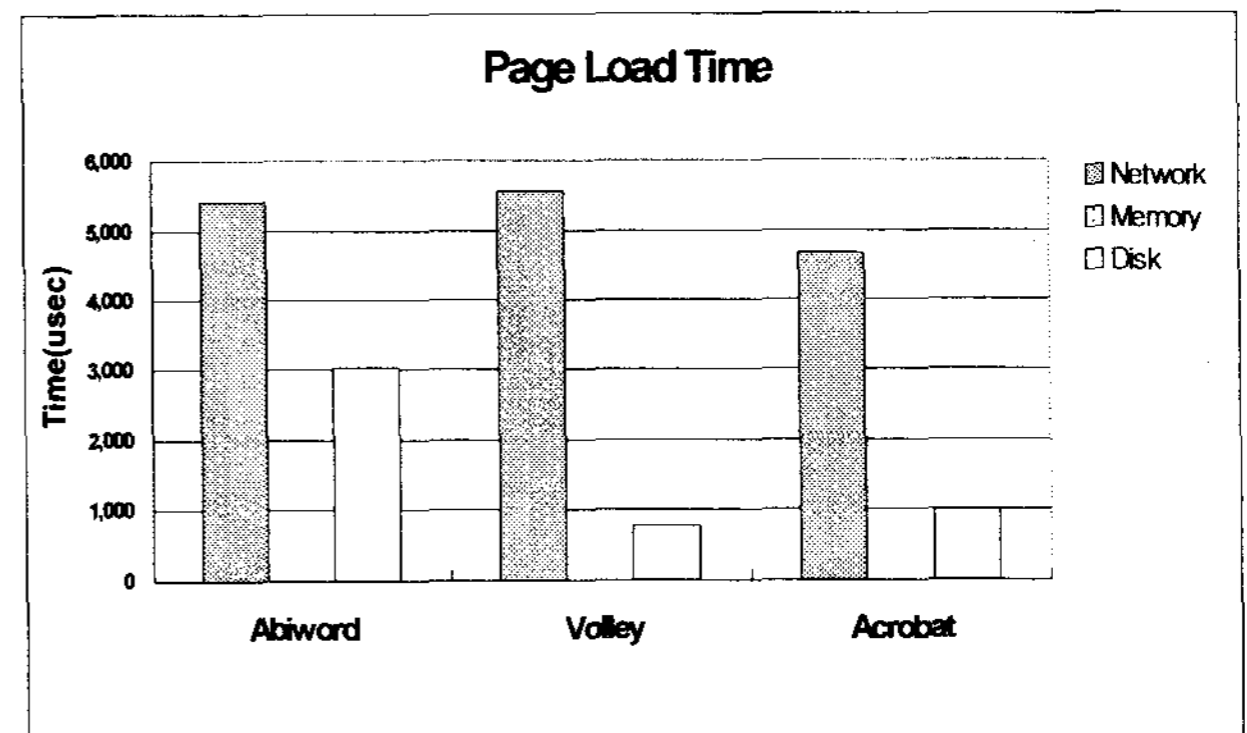
이번 절에서는 여러 상황 하에서 ALT를 측정한 결과를 소개한다. 측정 데이터의 정확성을 높이기 위해 측정한 데이터를 여러 차례 반복하여 이들의 평균을 구하였다. 우선 새로운 응용 소프트웨어의 최초 실행시의 초기 ALT를 측정하였다. 또한 동일한 프로그램에 대하여 재실행시 페이지 캐싱 메커니즘이 가진 성능향상 정도를 밝히기 위하여 재실행 ALT 값을 측정하였다. 또한 IELS, 즉 최초의 ALT 기간 동안 다운로드 되는 패킷 데이터의 크기 또한 측정하였다.



[그림 6] Application Load Time 비교

[그림 6]는 프로그램이 사용자에게 의하여 실행 선택된 직후부터 초기 사용자 인터페이스 윈도우가 화면에 나타날 때까지의 경과시간인 ALT를 측정한 결과값을 나타내고 있다. ALT는 각 프로그램에 대하여 초기실행시, 재실행시, 재부팅후 실행시의 세 경우에 대하여 측정하였다. 초기실행시에

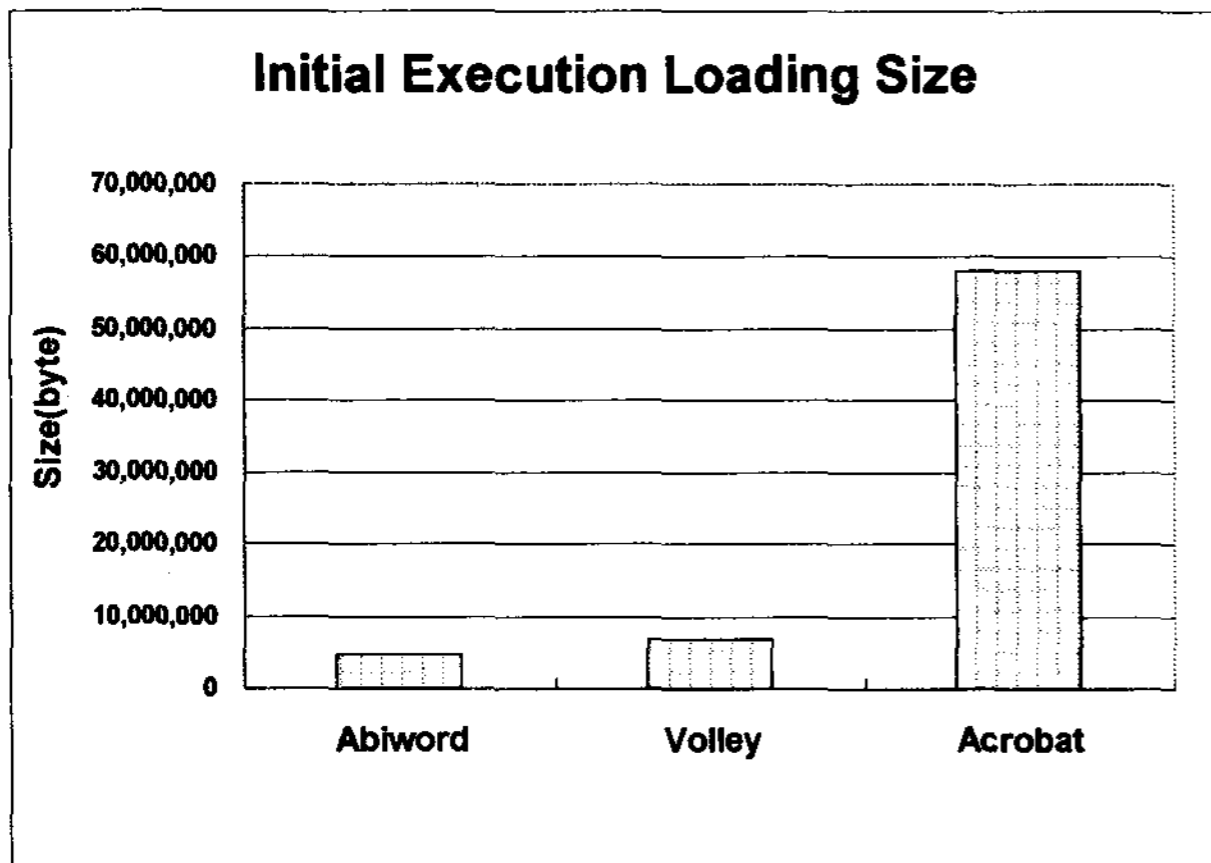
는 디스크 캐시나 메모리 캐시에 요구된 페이지가 존재하지 않으므로 페이지를 네트워크를 통해 얻어오게 된다. 그리고 초기실행한 이후에 바로 프로그램을 재실행하게 되면 한 번 얻어온 페이지는 이미 메모리 캐시에 저장되어 있으므로 페이지를 메모리 캐시에서 읽어오게 된다. 마지막으로 컴퓨터를 재부팅하고 프로그램을 실행하면 메모리 캐시가 재부팅시에 플리쉬 되었으므로 메모리 캐시에는 아무 페이지가 남아있지 않게된다. 하지만 이전에 네트워크를 통해 얻어온 페이지는 디스크 캐시에 저장되어 있으므로 해당 페이지를 네트워크가 아닌 로컬 디스크 캐시로 부터 읽어오게 된다. [그림 6]에서 보는 바와 같이 ALT는 재부팅후의 실행시가 초기실행시보다 낮고, 재부팅후 실행시 보다 재실행시가 낮은 것을 확인 할 수 있다. 즉 네트워크 보다는 디스크 캐시, 디스크 캐시 보다는 메모리 캐시에서 페이지를 얻어오는 것이 프로그램이 빨리 실행 된다는 것을 의미한다.



[그림 7] Page Load Time 비교

[그림 7]에서 보는 바와 같이 하나의 페이지를 읽어오는 데 소요되는 시간을 나타내는 PLT 또한 ALT와 마찬가지로 네트워크 보다는 디스크가, 디스크 보다는 메모리가 더 적은 시간이 걸리는 것을 확인할 수 있었다.

마지막으로 IELS는 Abiword의 경우 4.5MB, Volley의 경우 6.8MB, Acrobat의 경우 57.8MB를 요구하는 것을 [그림 8]에서 확인 할 수 있다.



[그림 8] Initial Execution Loading Size 비교

결론적으로 ALT는 환경 데이터의 크기, 프로그램의 초기화 또는 런칭에 관련된 데이터 크기를 의미하는 IELS와 직접적으로 비례하고, 하나의 페이지를 읽어오는데 소요되는 시간인 PLT와 비례하는 것을 확인 할 수 있었으며, 전송매체의 속도와 ALT가 반비례 관계라는 것을 확인 할 수 있었다.

3.5 성능 개선 방법

본 절에서는 앞 절에서 설명한 실험 결과의 분석으로부터 ALT를 줄이는 두 가지 성능 개선 방법을 제안한다.

임의의 프로그램이 마우스에 의하여 실행 선택된 직후부터 최초의 UI 윈도우가 화면에 나타날 때까지 다운로드 되어야 하는 페이지 전체를 백그라운드에서 최대 네트워크 속도로 다운로드한다면 ALT의 최소화가 이루어 질 수 있을 것이다. 이 제안 방법을 Application Initiation Accelerator(AIA) 방법이라고 부른다. Statistical Predictor Prefetching(SPP)으로 불리는 또 다른 성능 개선 방법은 자주 액세스하는 페이지를 접근 빈도와 시간 순서에 따라 스트리밍 서버에 기록한 후, 서버는 이 페이지를 백그라운드를 통해 SOD 시스템으로 송신하여 로컬 캐시에 저장하도록 한다. 이러한 페이지는 사용자 응용 프로그램 프로세스가 조

만간 접근할 가능성이 높은 부분이며, 앞 절의 PLT 측정을 통해 확인한 상대적으로 느린 네트워크가 아닌 로컬 캐시를 통해 페이지를 가져오기 때문에 보다 빠른 사용자 응용 프로그램의 실행을 가져올 수 있을 것이다.

4. 결론

본 논문에서는 프로그램 등록, 환경 변수 설정, 구성 파일 및 관련 컴포넌트의 자동 인스톨 기능을 자동적으로 실행함으로써 응용 프로그램의 신속하고 편리한 스트리밍 실행 서비스를 제공하는 Software On-Demand(SOD) 스트리밍 시스템을 소개하였다.

본 논문에서 제안된 SOD 시스템의 효용성을 보여주기 위하여 다양한 응용 프로그램 소프트웨어에 대한 SOD 스트리밍 실험을 통하여 성능측정 작업을 수행하였으며, 측정 결과의 분석을 통해 두가지 성능 개선 방법을 제안하였다. Accelerator Initiation Accelerator(AIA) 방법은 응용 프로그램의 초기 로드시간 ALT를 최소화하며, Statistical Predictor Prefetching (SPP) 방법은 응용 프로그램의 대기 시간을 현저하게 줄일 것으로 판단되는데 향후 제안한 두가지 성능 개선 방법을 구현하고 그 성능을 입증하기 위한 실험을 실시할 것이다.

[참고문헌]

- [1] Kuacharoen, P., Mooney, V., and Madisetti, V., "Software streaming via block streaming," Proceedings of the Design Automation and Test in Europe, pp. 912-917, Mar. 2003.
- [2] Raz, U., Volk, Y., and Melamed, S., "Streaming modules," U.S. Patent 6,311,221, Oct. 2001.
- [3] SOFTonNET Inc., "Z!Stream Technology," <http://www.softonnet.com>.