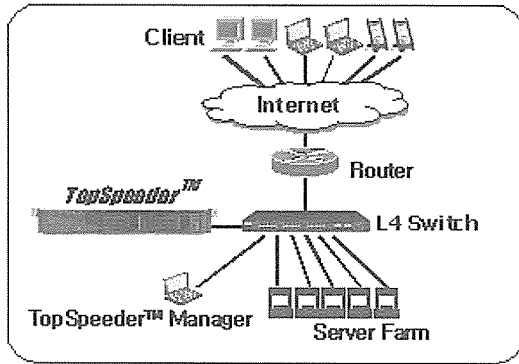


실시간 지능형 웹 가속기

드림인텍(주)(대표 김진술)의 TopSpeeder™은 사용자 응답속도 및 웹 서버 성능을 향상시키고 회선 요구량을 절감시키기 위한 목적으로 개발된 Web Acceleration 제품이다. 이 제품은 최상의 성능을 위해 LINUX kernel 2.4.0을 이용하여 Network I/O processing에 최적화된 새로운 운영체제인



TopSpeeder™ 네트워크 구성도

AccelOS™과, 이 운영체제를 기반으로 하는 WSM(Web Session Management), Data Optimizing, Caching 기능을 국내 독자 기술로 개발하였으며, 응답시간 및 웹 서버의 성능을 최대 10배 향상시키고, 회선 요구량은 50% 절감시킨다.

이 제품과 유사한 기존 제품들은 웹 가속을 위해 사용하는 기술을 모두 가지고 있지 않거나, 어플리케이션 레벨에서 작동하기 때문에 높은 성능을 발휘하지 못하고 있다. 그러나 이 제품은 웹 가속을 위해 사용하는 기술을 모두 적용하고, 빠른 처리를 위해 운영체제 레벨에서 기능을 구현한 제품으로서 높은 성능을 나타내고 있다. 또한, 사용자가 시스템을 편리하게 설정할 수 있도록 CLI 프로그램과 웹을 기반으로 하는 GUI 프로그램을 제공한다.

이 제품은 개발 당시부터 국내 및 세계시장을 목표로 개발된 제품이다. 현재 한국시장에서는 여러 업체에서 테스트 중이며 테스트한 사이트에서 호평을 받고 있다. 세계시장 진출을 위하여 11월 중순에 중국을 방문하여 China Telecom에서 설명회를 가졌으며, 12월에 대만에서 설명회를 개최할 예정이다.

1. 작품명 : TopSpeeder™ (실시간 지능형 웹 가속기)

2. 제작자 : 드림인텍(주)

대표자 : 김진술
 개발참여자 : 최태암, 정찬성, 이석문, 강기호, 원종호
 주소 : (121-190) 서울시 마포구 창전동 333번지 우성빌딩
 전화 : 02) 3141-7575
 팩스 : 02) 3141-8205
 email : jskim@dreamintek.co.kr

3. S/W 요약설명

인터넷 웹 가속을 위해 사용하는 기술은 WSM(Web Session Management), Data Optimizing, Caching 기술이 있으며 각 기능을 사용함으로써 다음 [표 1]과 같은 효과를 볼 수 있습니다.

적용 기술	기대 효과
WSM 기술	웹 서버의 성능 향상
Data Optimizing 기술	사용자 응답 속도 향상 및 네트워크 대역폭 사용 최소화
Caching 기술	사용자 응답 속도 향상과 웹 서버 증설 요구 감소

[표 1] 웹 가속 기술과 기대 효과

WSM 기술은 웹 서버와의 연결을 관리함으로써, 웹 서버가 사용자와 연결을 위해 사용하는 자원을 절감시켜 처리 능력을 향상시키게 합니다. 또한, 사용자에게 전송할 데이터의 크기를 감소시키는 Data Optimizing 기술은 네트워크 대역폭 사용량을 최소화할 뿐 아

니라, 사용자 응답 속도를 향상시키게 됩니다. 마지막으로 Caching 기술은 전송할 데이터를 자체에 저장함으로써, 사용자 요구를 서버에서 처리하지 않고 자체에서 처리하여 서버로 요구되는 트래픽을 감소시킬 뿐 아니라, 사용자 응답을 위한 페이지를 생성하는데 소요되는 시간이 제거되어 빠른 응답을 기대할 수 있습니다.

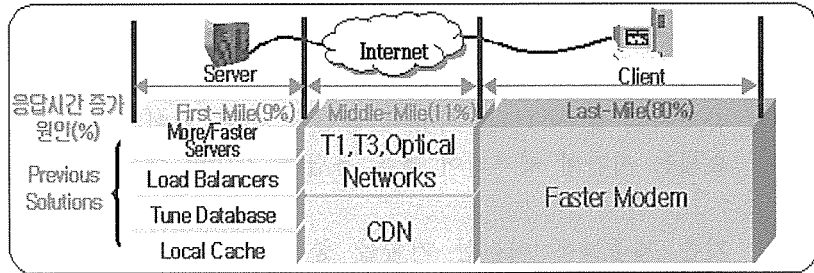
기존 제품들은 이러한 기술들 중 일부만을 지원하는 제품이지만, TopSpeeder™은 효율적인 처리를 위하여 Linux kernel을 이용하여 AccelOS™을 개발하였고, 이를 기반으로 하는 WSM, Data Optimizing, Caching 기능을 가지고 있는 제품입니다. 그렇기 때문에 TopSpeeder™을 웹 서버 앞단에 어플라이언스 형태로 설치하여 웹 서비스에 대해 높은 성능향상을 기대할 수 있습니다.

3.1 개발 배경

인터넷은 상호 정보를 공유하기 위한 수단으로 사용되기 시작하여 통신, 교육, 오락, 상거래 등으로 그 사용 범위가 확대되면서 가히 혁명이라 일컬을 정도로 인터넷을 이용하는 사용자와 기업이 폭발적으로 증가하게 되었습니다. 이러한 인터넷이 상거래의 가장 중요한 수단이 되면서, 인터넷 성능이 경쟁력의 중요한 요소가 되었습니다.

인터넷 응답속도를 향상시키기 위한 많은 노력들로 인하여, 1999년의 인터넷 성능은 1995년에 비해 100% 향상되었습니다 (출처 : “The World-Wide-Wait Status Report”, NCRI, October 1999). 그러나, 인터넷 Traffic은 4~6개월마다 2배 증가하는 반면 네트워크 인프라는 매 18개월마다 2 배가되므로, 늘어나는 트래픽 수요를 충족시키지 못할 것으로 예측하고 있습니다.

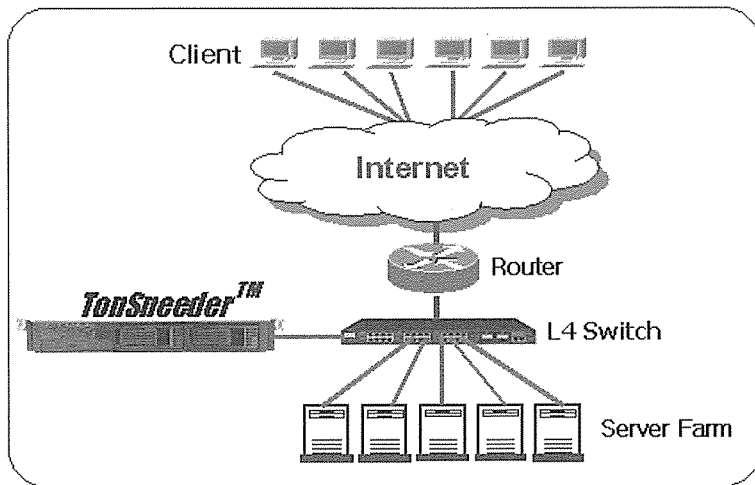
다음 [그림 1]은 인터넷 환경에서 응답시간 증가에 대한 원인을 구간별로 표현하고, 각 구간에서 응답 지연을 해결하기 위해 사용하고 있는 기술입니다. 인터넷 속도 지연에서 가장 많은 부분을 차지하는 last mile에 대한 사용자 접속 속도에 대한 해결 방법 없이는 인터넷의 속도 개선을 매우 제한적일 수밖에 없습니다. 그러나 이러한 문제는 인터넷 서비스를 제공하는 사업자의 노력만으로는 해결할 수 없다는데 문제가 있습니다.



[그림 1] 인터넷 구간별 응답지연과 해결하기 위해 사용된 기술

이러한 상황에서, 인터넷 응답속도를 향상시키기 위해 Linux kernel에서 event-driven 방식으로 WSM(Web Session Management), Data Optimizing, In-Memory Caching 기능이 지원되는 TopSpeeder™을 설계 및 구현하게 되었습니다.

3.2 시스템 개요



[그림 2] TopSpeeder™ 네트워크 구성도

TopSpeeder™은 [그림 2]와 같이 웹 서버 앞단에 어플라이언스 형태로 네트워크에 설치되어, 사용자 응답속도 및 웹 서버 성능을 향상시키고 회선 요구량을 절감시키기 위한 목적을 가지고 있는 Web Acceleration 제품입니다. 최상의 성능을 위해 Linux kernel을

이용하여 Network I/O processing에 최적화된 새로운 운영체제인 AccelOS™을 개발하였고 이 AccelOS™을 기반으로 하는 WSM, Data Optimizing, Caching 기능을 구현하였습니다.

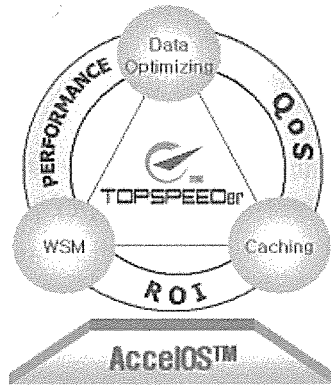
AccelOS™은 Network I/O Intensive Operation을 최적화하기 위한 목적으로, Linux kernel을 수정하여 Network I/O processing에 대한 처리과정과 kernel의 자료구조를 단순화시키고, event-driven 방식으로 동작하게 구현함으로써, 기존 Application level에서 구현된 방법에 비해 처리량은 4배 증가함과 동시에 응답 속도는 59배 향상되었습니다.

AccelOS™에서 운용되는 WSM 기능은 웹 서버와의 TCP session을 관리하여 서버 성능을 향상시키고, Data Optimizing 기능은 Text Compression과 Image Reduction 모듈을 이용하여 사용자에게 전송 될 Object의 크기를 줄임으로써 사용자 응답속도를 향상시키고, 회선 요구량을 절감시키게 됩니다. Caching 기능은 Data Optimizing 기능에 의해 Optimize된 Object를 메모리에 저장하여 같은 요청이 요구되었을 때, 메모리에 저장된 Object를 전송함으로써 Optimize에 대한 overhead를 제거하여 사용자 응답속도가 향상되고 웹 서버의 부하도 감소하게 됩니다.

3.3 시스템 특징

TopSpeeder™은 사용자 응답속도 및 웹 서버 성능을 향상시키고 회선 요구량을 절감시키기 위한 목적을 가지고 있는 웹 가속 제품으로, 웹 서버가 있는 네트워크 구성에 어플라이언스 형태로 설치되는 제품입니다.

웹 가속을 위해 개발된 기능들이 최대한의 성능을 발휘할 수 있도록 Linux Kernel을 이용하여 AccelOS™을 개발하였으며, 이 운영체제를 기반으로 웹 가속과 관련된 WSM, Data Optimizing, In-Memory Caching 기능들이 다음 [그림 3]과 같이 지원됩니다.



- ❖ AccelOS™
 - Network I/O에 대해 event-driven 방식으로 kernel 안에서 구현하여 처리 속도 대폭 향상
- ❖ Data Optimizing
 - Contents의 크기를 최적화 시켜 응답속도 향상 및 회선요구량 절감
- ❖ WSM (Web Session Management)
 - 웹 서버의 커넥션을 관리하여 서버 성능 극대화
- ❖ Caching (Optimized)
 - 빈번히 요청되는 Data를 Optimized Caching하여 응답속도 향상

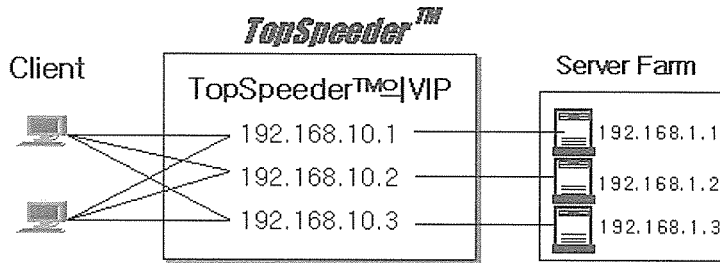
[그림 3] TopSpeeder™의 기능

서버의 성능을 향상시키기 위해 TopSpeeder™은 WSM 기능을 이용하여 웹 서버가 사용자와 직접 연결하지 않고 TopSpeeder™에 의해 사용자와의 웹 세션을 관리하도록 합니다. 또한 In-Memory Caching 기능에 의해 정적 데이터는 TopSpeeder™에 의해 처리됨으로써, 서버의 부하를 줄일 수 있게 됩니다. 즉, TopSpeeder™은 웹 서버와의 세션 관리와 In-Memory Caching 기능을 이용하여 웹 서버의 성능을 최고 10배까지 향상시킵니다.

사용자 응답속도와 회선 요구량을 절감시키기 위해 TopSpeeder™은 Data Optimizing 기능을 사용합니다. Data Optimizing 기능에 의해 TopSpeeder™은 사용자에게 전송할 페이지의 각 Object 사이즈를 줄이고, 이로 인하여 사용자의 응답속도는 최고 10배까지 향상되며, 회선 요구량은 50% 정도 절감하게 됩니다. 또한, In-Memory Caching 기능으로 인하여 TopSpeeder™의 Data Optimizing 기능을 위한 부하를 감소시키고, 웹 서버의 응답 페이지 생성 시간이 제거되어 더 빠른 응답속도를 제공합니다.

3.4 시스템 구성

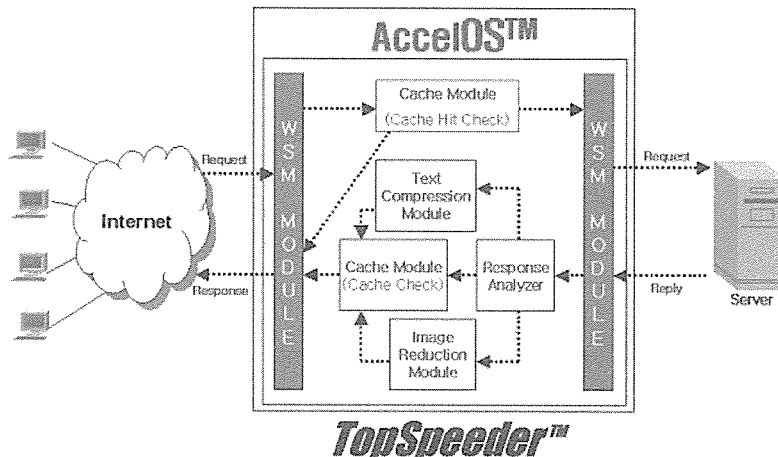
TopSpeeder™은 웹 가속을 위해 웹 서버 앞단에 어플라이언스 형태로 [그림 2]와 같이 네트워크에 설치되는 제품으로서, 2개의 CPU와 100M나 1000M를 지원하는 랜 포트, 2기가의 메모리를 가지고 있는 하드웨어에 Network I/O processing에 최적화된 AccelOS™을 탑재하였으며, 이 AccelOS™을 기반으로 하여 WSM, Data Optimizing, In-Memory Caching 기능을 지원합니다.



[그림 4] TopSpeeder™의 VIP 설정

TopSpeeder™은 [그림 4]와 같이 VIP를 이용하여 웹 서비스를 하게됩니다. 즉, 웹 서버의 앞단에서 TopSpeeder™이 실제 웹 서버의 각 IP와 1:1로 대응될 수 있도록 VIP를 설정해야 하며, 사용자(Client)는 TopSpeeder™의 VIP를 사용하여 웹 요청을 합니다. 사용자가 TopSpeeder™의 VIP를 이용하여 웹 요청을 하면, TopSpeeder™은 처리 가능한 데이터인 정적 데이터에 대해 바로 사용자에게 응답을 하게 됩니다. 동적 데이터인 경우 VIP에 연결된 웹 서버와 접속하여 사용자에게 전달할 데이터를 받아 Data Optimizing 과정을 거친 후, 사용자에게 전달하게 됩니다.

3.5 프로그램구성



[그림 5] TopSpeeder™의 구성도

TopSpeeder™은 [그림 5]와 같이 AcceIOS™을 기반으로 WSM, Data

Optimizing, In-Memory Caching 기능을 가지고 있습니다.

AccelOS™

프로그램을 개발할 때, 최상의 성능을 얻기 위해 여러 가지 프로그램 구조 중에서 하나를 선택하게 됩니다. 프로그램 구조의 예에는 multi-threaded single-process 구조, multi-threaded multi-process 구조, single-process asynchronous I/O 구조, mixed asynchronous I/O with threads 구조 등이 있고 이러한 구조를 사용하여 application mode에서 운영되도록 개발하는 것이 일반적입니다.

그러나, 일반적인 프로그램 구조를 사용해서는 H/W 가격 대비 성능 비율에 있어서 경쟁력을 가질 수 없습니다. 그래서 최고의 H/W 가격 대비 성능 비율을 가질 수 있는 프로그램 구조를 개발하던 중에 운영 체제(Operating System)의 system call, scheduling, context switching, I/O event 전달 등의 처리로 인한 overhead가 매우 높다는 점과 application mode에서 동작시키기 위해 TCP의 session을 관리하기 위한 데이터가 file descriptor-> I-node->socket->sock으로 구성되어 시스템 자원의 낭비가 심하다는 점에 착안하게 되었습니다. 이러한 착안을 바탕으로 Network I/O processing에 최적화된 새로운 운영체제인 AccelOS™을 개발하였습니다. 이로 인하여 웹 데이터에 대해 실시간 처리가 가능하게 되었고, 어플리케이션 레벨에서 구현된 경우에 비해 다음 [표 2]와 같은 성능 향상을 볼 수 있었습니다.

측정 항목	AccelOS™	Application Level	성능 향상
Gets/sec	1,310.5	299.5	4배
TTLB(ms)	2.3	135.1	59배

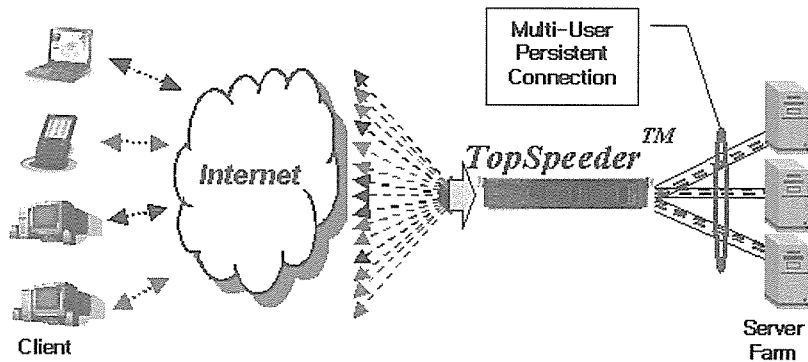
[표 2] AccelOS™의 성능 비교표

WSM(Web Session Management)

웹에서 한 페이지는 평균 40개의 object로 구성되어 있고, HTTP version 1.0에서는 object 하나를 다운로드하기 위해 한번의 TCP connection이 발생하기 때문에 한 페이지를 보기 위해서 40번의 TCP connection이 발생합니다. 이러한 방식은 다음과 같은 두 가지 문제점을 가지고 있습니다.

- 네트워크 Bandwidth를 충분히 활용하지 못함
- TCP connection 관리에 사용되는 서버의 자원(CPU time, memory 등) 낭비 심함

상기의 문제점을 개선하고자 HTTP version 1.1에서는 persistent connection이라는 기능을 추가하였는데, 이는 하나의 TCP connection을 이용하여 다수개의 object를 다운로드 할 수 있도록 한 것입니다. 그러나 이러한 방식은 TCP connection의 유희상태로 인한 서버 자원을 낭비한다는 문제점이 발생하기 때문에, persistent connection 기능을 충분히 활용하지 못하게 됩니다.



[그림 6] WSM 기능

이러한 점에 착안하여 TopSpeeder™은 [그림 6]과 같이 TopSpeeder™이 미리 서버와 연결된 상태에서 사용자의 연결을 받아들입니다. 사용자와 연결된 TopSpeeder™은 사용자의 요구를 연결된 connection을 이용하여 전달하고, 그에 대한 응답을 서버로부터 받게 됩니다. 즉 이러한 방식은 서버와의 연결을 다수의 사용자가 공유하여 사용할 수 있게 함으로써, persistent connection 기능을 충분히 활용하게 되고, 서버의 TCP connection에 사용되는 시스템 자원의 낭비를 최소화하여 content 생성과 전달에 가능한 많은 시스템 자원이 사용될 수 있도록 합니다. 특히, 사용자의 접속속도가 느린 경우 content의 전송이 끝날 때까지 서버는 시스템 자원(특히 메모리)을 활용하지 못하게 됩니다. 즉, 서버의 메모리 활용율이 낮게 되어 서버 성능을 떨어지게 하는 원인 중 하나가 됩니다. 이러한 점을 개선하기 위해 TopSpeeder™은 서버로부터 빠른 속도로 content를 전달

받아 저장함으로써, 서버가 해당 메모리를 다른 목적으로 사용할 수 있도록 합니다.

사용자가 증가하면서 응답시간이 증가하게 되는데, 일정한 한계를 초과하는 경우 processing time이 응답속도를 저하시키는 가장 큰 원인이 됩니다. 그러나 WSM을 적용한 경우 기존에 연결된 TCP connection을 이용하여 처리하기 때문에, 응답속도의 증가를 억제할 수 있게 됩니다. 따라서, WSM이 없는 경우 사용자에게 만족스러운 응답시간을 제공하기 위해서는 서버를 계속하여 증설하여야 하지만, WSM을 사용하면 적은 수의 서버로 보다 많은 사용자를 수용할 수 있게 되어 서버 증설 요구를 감소시킬 수 있습니다.

Data Optimizing

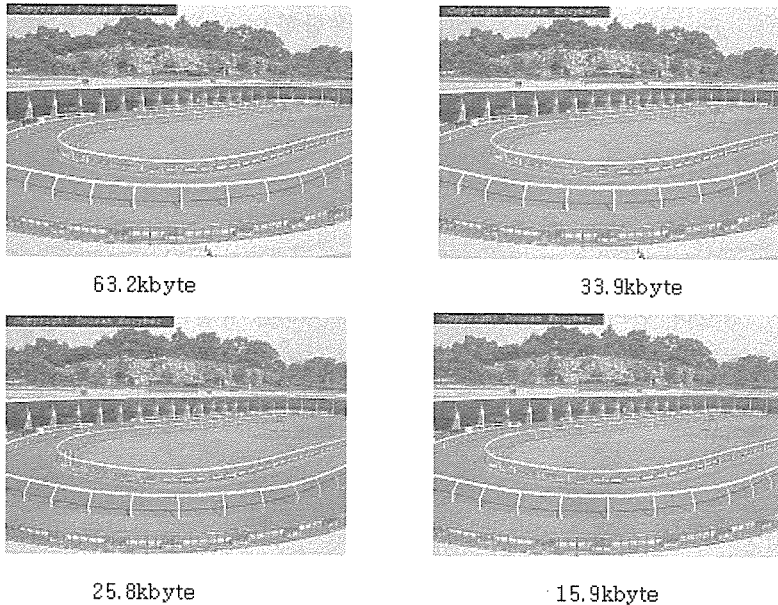
사용자 응답속도를 개선하기 위해서 홈페이지의 복잡도를 낮추어 사용자에게 전송될 데이터 양을 줄이도록 권고하고 있습니다. 그러나 많은 사이트들은 높은 품질의 시각적 이미지로 고객의 감성에 호소하여 구매욕구를 높이고자 노력하고 있습니다. 문제는 이러한 content를 사용자의 컴퓨터로 전송 받는데 오랜 시간이 필요해 오히려 이용자의 발길을 다른 곳으로 돌리게 하는 요인이 된다는 점입니다. 때문에 TopSpeeder™은 전송할 데이터 양을 줄이기 위해 홈페이지 크기를 줄이지 않아도 홈페이지의 크기를 줄인 효과가 나도록 하여 다운로드 시간을 획기적으로 줄이는 Data Optimizing 기술을 사용합니다. Data Optimizing 기술에 의해 사용자 응답 속도를 향상시키는 예를 들면, 만약 10Kbyte의 object(웹 페이지를 구성하는 단위)를 1Kbyte/sec로 전송하면 10초가 소요된다고 가정했을 때, 10Kbyte를 1Kbyte로 축소할 경우 1초가 걸리게 됩니다. 즉 응답속도가 10배 향상되는 결과가 얻을 수 있습니다.

Object의 종류	적용 기술
Text-based	Data Compression
Graphic(jpeg, gif)	Image Reduction

[표 3] Object에 따른 Data Optimizing 기술

[표 3]과 같이 TopSpeeder™에서 사용하는 Data Optimizing 기술은 Object의 종류에 따라 Data Compression 기술과 Image Reduction 기술을 적용하고 있습니다. Graphic object인 경우 Image Reduction

을 수행하기 때문에 원상복구가 불가능한 점이 문제가 될 수 있다고 생각할 수 있을 것이다. 그러나 이미지 품질을 사용자가 원본과의 차이점을 판별할 수 없을 정도로 감소시키기 때문에 문제가 되지 않습니다. 이를 위해 아래 [그림 7]에서 여러 단계로 Image Reduction한 예를 보여 주고 있습니다.



[그림 7] Object의 Image Reduction 결과

[그림 7]은 원본 크기에 비해 Image Reduction한 후, 크기가 25%로 줄었음에도 불구하고, 이미지 품질은 거의 변화가 없음을 알 수 있습니다.

In-memory Caching

일반적으로 Web Server의 성능을 향상시키기 위하여 Cache Server를 사용하고 있습니다. 그러나, 이러한 Cache Server는 Disk 장치에 object를 저장하기 때문에 사용자의 요청에 대해 응답을 위하여 Disk I/O 작업을 거치게 됩니다. 이러한 작업의 빈번한 발생은 응답속도 지연 현상을 초래하게 되어 성능 저하가 발생하게 됩니다.

TopSpeeder™은 이러한 문제점을 해결하기 위해, Disk 대신 미리 할당된 Memory에 caching될 object를 저장하고 관리함으로써, Disk

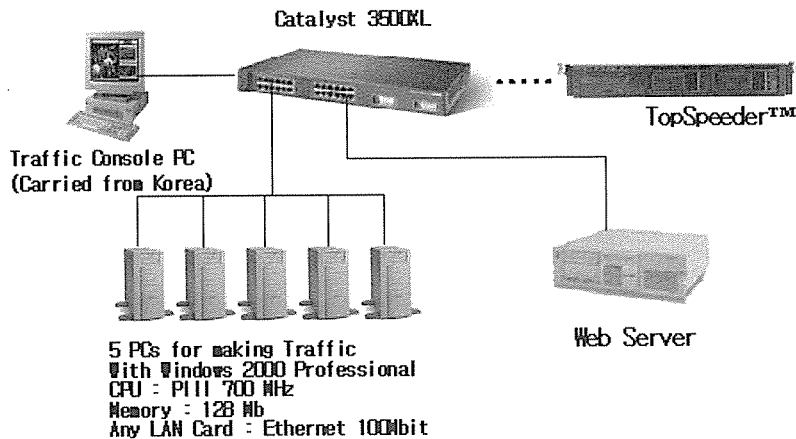
I/O와 Memory 할당 시간을 제거하기 때문에 사용자의 요청에 빠르게 응답할 수 있습니다.

Data Optimizing 과정에 의해 최적화된 object를 Memory에 caching 함으로써 같은 object에 대해 반복적으로 optimizing되는 것을 방지하고, 이로 인하여 TopSpeeder™의 부하를 상당히 절감하여 더 높은 성능 향상을 제공합니다. 또한, 사용자가 데이터 요구 시 Memory에 caching된 Object를 이용하여 응답함으로써, 각 요구에 대해 더 빠른 응답시간을 제공하고, 초당 요구 처리 수(Reqs/sec)도 증가합니다.

3.6 성능 평가

테스트 조건 및 환경

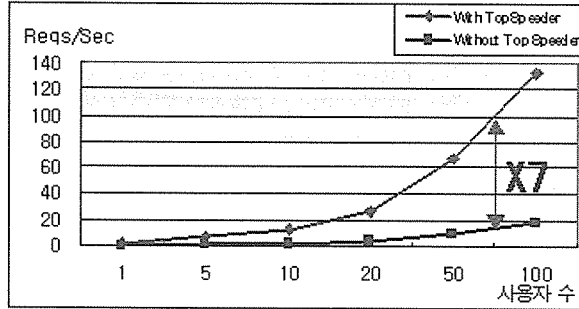
TopSpeeder™의 성능 평가를 위해서 [그림 8]과 같이 네트워크를 구성하였으며, 인터넷 웹 서버의 성능을 테스트하는 유틸리티인 Microsoft사의 Web Application Stress Tool 1.1을 이용하여 성능을 평가하였습니다.



[그림 24] 테스트를 위한 네트워크 구성도

[그림 8]은 TopSpeeder™의 성능평가를 위한 네트워크 구성도로서, Traffic 발생을 위한 5대의 client와 Traffic 생성을 모니터링하기 위한 Traffic Console PC에 Web Application Stress Tool 1.1을 설치하였으며, Web Server를 위해서 듀얼 펜티엄III CPU를 탑재한 PC 서버에 IIS를 설치하여 테스트하였습니다.

테스트 결과 - 저속 네트워크 환경

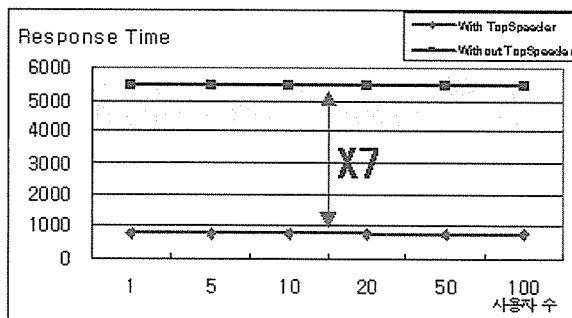


[차트 1] 사용자 수에 따른 초당 요구 수 (56Kbs)

저속 네트워크 환경에서 사용자 수에 따라 초당 처리되는 요구 수를 [차트 1]에서 나타내고 있으며, 저속 네트워크 환경에서는 성능이 [표 5]에서 볼 수 있듯이 약 700% 향상됨을 알 수 있습니다.

	1	5	10	20	50	100
Server Direct	0.18	0.92	1.83	3.67	9.16	18.33
TopSpeeder ^{1M}	1.21	6.62	13.07	26.42	66.81	132.30
성능향상비	672.2	719.6	714.2	719.9	729.4	721.8

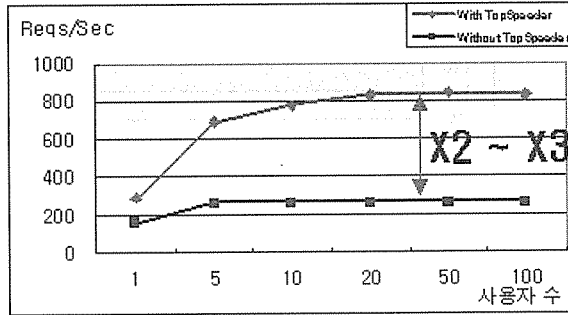
[표 6] 사용자 요구 수에 따른 성능 향상 비 (56Kbps)



[차트 2] 사용자 수에 따른 응답 시간 (56Kbps)

[차트 2]는 사용자 수에 따른 응답시간을 나타내는 차트로서 초당 요구 수를 나타내는 [차트 1]과 같이 저속인 경우 응답시간(Response Time)도 약 700% 정도의 성능이 향상됩니다.

테스트 결과 - 고속 네트워크 환경(LAN : 100M)

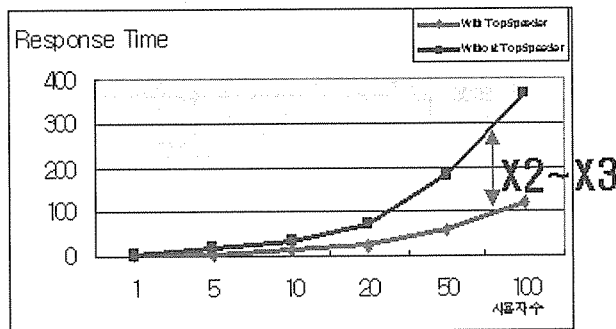


[차트 3] 사용자 수에 따른 초당 요구수 (LAN)

[차트 3]은 고속의 네트워크 환경에서 사용자 수에 따라 초당 처리되는 요구를 나타내는 차트로서 성능 향상 비는 다음 [표 5]와 같습니다. 즉, 고속 네트워크 환경에서는 200% ~ 300% 정도의 성능이 향상됩니다.

	1	5	10	20	50	100
Server Direct	156.74	260.43	261.88	260.3	261.21	260.72
TopSpeeder ^{1M}	293.99	688.38	779.92	831.74	842.01	835.62
성능향상비	187.6	264.3	297.8	319.9	323.3	320.5

[표 7] 사용자 요구 수에 따른 성능 향상 비 (LAN)



[차트 4] 사용자 수에 따른 응답시간 (LAN)

[차트 4]는 고속의 네트워크 환경에서 사용자 수에 따른 응답시간을 나타낸 결과로서 200% ~ 300%이상 성능이 향상되었습니다.

결론적으로, 임의적으로 만든 네트워크의 다양한 속도에 대해서

TopSpeederTM은 서버의 성능을 최대한 활용할 수 있도록 도와주며, 네트워크의 BandWidth를 최대한 활용할 수 있도록 하여 주어진 QoS 환경을 100% 활용할 수 있도록 합니다.

4. 개발단계별 기간 및 투입인원수

구분		개발 일정																	인원
		2000년									2001년								
		6월	7월	8월	9월	10월	11월	12월	1월	2월	3월	4월	5월	6월	7월	8월	9월		
시스템 계획 및 자료 분석	프로젝트 구상	■																1	
	구축목표 설정		■															2	
	구축내용 결정		■	■														2	
	자료 수집			■	■	■												3	
요구사항 분석 및 설계	자료 분석			■	■	■												3	
	동종시스템 분석				■	■												3	
	Prototyping				■	■	■											3	
	프로세스 설계					■	■	■										3	
시스템 개발	설계 명세서 작성				■	■	■											3	
	개발 환경 구축						■											3	
	event- driven 방식							■	■	■	■	■						1	
	WSM							■	■	■	■	■						1	
	Caching							■	■	■	■	■						1	
	Data Optimizing								■	■	■	■	■					1	
운영 및 유지 보수	시스템 운영 지침서작성														■			1	
	기타 메뉴얼 작성															■		1	
	시스템 β -테스트														■	■		1	
	시스템 보완사항 보수															■	■	2	
개발 완료	개발종료															■		3	

5. 사용 또는 개발언어, TOOL

System Programming	Compile	gcc-2.96.69
	Debugging	gdb 5.00
	Profiling	kernprof-1.2.1
Web Programming	gcc-2.96.69, JavaScript	

6. 사용시스템

개발 시스템

사용 모델	eSlim SUI-2000	eSlim의 server 제품
사용OS	Linux Kernel	Version : 2.4.0
CPU	Dual Pentium III 1GHz	Intel
RAM	2GB	ECC type
Network Card	Triple 10/100 LAN	Intel

테스트 시스템 - 5 대

사용OS	Windows 2000	Professional Version
CPU	Pentium III Celeron 800	Intel
RAM	256MB	ECC type
Network Card	10/100 LAN	Realtek 8209
테스트 프로그램	Web Application Stress Tool 1.1	MicroSoft Co.