

이동컴퓨터를 위한 Web Proxy Server의 설계

이종국, 김명철, 마중수
한국정보통신대학원

A Design of Web Proxy Server for Mobile Computer

Jong-Kuk Lee, Myung-Chul Kim, Joong-Su Ma
Information & Communications Univ.

요약

최근 이동컴퓨터(Mobile computer)는 사용자들이 기하급수적으로 늘고 있다. 그러나, 이동컴퓨터는 기존의 PC나 W/S를 사용하는 환경보다 이동성을 고려하므로, 화면 크기, 부속들의 크기와 무게, CPU에 사용되는 전력, 네트워크 인터페이스에서 제약성을 지니고 있다. 이런 제약으로 인해 이동컴퓨터는 기존의 컴퓨터 보다 성능이 떨어진다. 이런 환경에서 인터넷을 사용할 때 WWW 같은 멀티미디어 전송 서비스는 원하는 시간 내에 처리하기 힘들다. 따라서, 현재 이동컴퓨터를 위한 인터넷 서비스들은 Proxy server를 이용하여 멀티미디어 데이터의 크기를 줄이거나 문자로 처리하게 하여 사용자에게 좀 더 빠른 환경을 제공하고 있다. 그러나, 기존의 Proxy Server들에서 구현된 화면에서 보여지는 그래픽 파일(Graphic File)의 일방적인 size¹의 조정으로 파일의 양을 줄이기 때문에 성능 차이가 크게 나는 이동컴퓨터에게 똑같은 정보를 제공함으로써 환경이 다양한 이동컴퓨터에는 충분하지 못하다는 단점을 가지고 있다.

본 논문에서는 이동컴퓨터환경에서 사용자들이 원하는 WWW service를 빠른 시간에 받을 수 있게 하면서, 화면크기, 색 표현능력 등이 각기 다른 이동컴퓨터의 H/W 사양에 맞추어서 효과적으로 보낼 수 있게 포트(port)를 다양화한 Proxy server를 설계해 보도록 한다.

1. 서론

1992년 WWW의 등장과 함께 인터넷을 통한 멀티미디어 통신이 이루어지면서, 컴퓨터의 보급은 급속히 확산되었다. 점차 이동성이 없는 컴퓨터 환경에서 이동하면서도 컴퓨터를 사용할 수 있고, 언제 어디서나 네트워크를 사용할 수 있는 환경이 요구되었다. 이런 환경을 지원하는 컴퓨터 환경을 이동 컴퓨터 환경(Mobile Computing)이라고 부른다[7].

Platform	Speed(Mhz) /Mem.	Screen Size (Bit/Pixel)	Network Bandwidth
High-end PC	400/64-128M	1280x1024 (32, color)	100/10Mbps (LAN)
Notebook	333/16-64M	1024x768 (16, color)	100/10Mbps (LAN)
Handheld PC(HP660LX)	75/10M	640x240 (8, color)	1-2Mbps or 56Kbps
PDA(Palm Pilot VII)	16/2M	160x160 (2, gray)	56Kbps (Modem)

(표 1) 컴퓨터 H/W 성능 비교 (<http://www.palm.com>, <http://www.intel.com/>, <http://www.computers.com>)

이러한 이동컴퓨터 환경시장은 급속히 성장하고 있다. 이동 컴퓨터는 1998년의 경우 390만대가 팔렸고, 2003년에는 2100만대가 팔릴 것으로 예측되고 있다. 매출액에서만 1999년 23

억 달러에서 2003년에는 72억 달러로 급속적인 성장을 보일 것으로 기대된다[5]. 이런 급속한 성장 속에서 사용자들은 멀티미디어 등의 다양한 서비스를 요구하고 있다. 그러나, (표 1)에서 나타나듯이 이동컴퓨터는 PC와 상대적으로 성능이 떨어진다. 즉, 그래픽 파일 양과 처리해야 할 용량이 점점 커지는 WWW의 경우 PDA나 Handheld PC가 처리를 모두 담당하기는 비록 컴퓨터 성능이 날로 향상되고 있지만 무척 힘들다. 이를 위한 해결방법으로는 Web Proxy Server를 사용하여, 그래픽 파일의 양을 줄이는 방법이 대표적인 예이다. 현재 이 방법을 사용하는 Proxy Server들은 최근 2-3년 사이에 등장하였다. 처음 선보인 것은 Berkeley 대학의 Pythia와 Transend이며, 시제품으로는 Proxinet©에서 개발한 Webproxy와 Spyglass©에서 개발한 Proxy server 등이 있다. 그러나 이 제품들 모두 한가지 플랫폼(platform)을 기준으로 해서 만든 것으로 지금까지 나온 여러 가지 이동컴퓨터들의 화면 크기, 해상도 및 네트워크 대역폭(Network bandwidth) 등의 다양성을 충족시키지 못하고 있다[1][6].

2. 관련연구(Transend)

Transend란 Berkeley 대학의 컴퓨터과학부(Computer Science Division)에서 제공하는 transformational proxy service를 말한다. Transend는 캐쉬(cache)한 멀티미디어 데이터를 변환시키는 Proxy server와 그것을 이용한 web browser로 이동컴퓨터의 문제를 해결하려고 하고 있다. 즉, 여러 다른 환경에서

¹ size : 이 논문에서는 size를 그래픽 파일이 스크린에 보여지는 크기로 한다.

사용자 자신이 원하는 품질로서 Proxy server가 멀티미디어 데이터를 줄여서 web browser로 보낸다는 것이다. 이 방법을 사용한다면 39.3KB의 멀티미디어 데이터를 28.8 kbps 모뎀으로는 11초, 14.4 kbps 모뎀으로는 22초 이상 걸리던 전송시간이, 30ms의 시간을 투자하여 멀티미디어 데이터를 줄이면, 3.9KB로 양을 줄일 수 있을 뿐더러 28.8kbps모뎀으로는 1초 정도 걸리게 되고, 14.4 kbps 모뎀으로는 2초가 걸리게 된다[4].

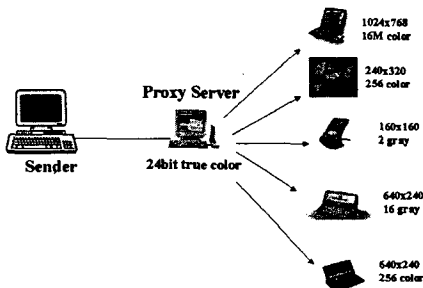
여기에서 확장하여 논문[2]의 저자들은 단순한 어플리케이션(application)의 개발이 아닌 TACC(Transformation, Aggregation, Caching, Customization)라는 architecture를 보여주고 있으며, TACC의 예가 논문[3]에서 개발한 Transend이다.

여기에서는 그래픽 파일(Graphic file)의 양을 줄이는 부분에서 초기 모델인 Pythia는 칼라와 흑백의 여부, 색상의 수, 압축시킬 때 걸리는 최대 시간의 설정 변수인 Max wait time, 그래픽 파일의 size의 여부를 결정짓는 size조정의 4가지 변수를 사용자들이 단계별로 설정을 하게 해서 그래픽 파일들을 압축하고 있다. Pythia에 비해, Transend는 2가지 모드(mode)를 가지고 있다. 첫 번째는 사용자 모드(user mode)로서, quality에 관한 5단계의 조정변수를 가지는 경우이다. 두 번째는 전문가 모드(expert mode)로서, web page에서 보여지는 그래픽 파일의 최대 size와 최소 size, size를 줄이는 비율을 지정할 수 있고, 그와는 별도로 quality 변수들을 바꿀 수 있다.

3. 문제점 제기

지금까지의 연구에서 발생하는 문제는 다음과 같다.

- 1) 디스플레이(Display)의 비효율성 문제 : Transend의 경우, 이동컴퓨터 디스플레이 크기에 관계없이 변수 값에 따라서 그래픽 파일의 size를 조정하므로 원래 PC나 W/S에서 보던 그래픽 파일의 조건과는 틀리게 조정된다.
- 2) 확장성(Scalability) 문제 : Transend의 경우, 한 Server에 다양한 이동컴퓨터의 입출력이 행해지고, 그 해당 이동컴퓨터마다 사용자의 Profile을 가지고 있고, 그 사용자마다 줄이는 비율과 원하는 품질(quality)이 틀리다. 따라서, 각각의 이동컴퓨터마다 캐쉬를 따로 가지고 있어야 한다. 그러므로, 각 사용자마다 캐쉬를 가짐으로써 프로세싱(processing) 낭비 및 적중률(hit ratio)가 떨어지게 되며 수많은 사용자가 요구할 시에는 확장성에 문제가 발생한다.



(그림 1) Proxy server가 다른 환경의 이동컴퓨터들에게 data를 전송

(그림 1)에서 나타났듯 색상 수가 24bit true color이면서 500x600pixel 의 500KB의 그래픽 파일을 Transend와 같이 quality를 3으로 보장해서 모두에게 보낸다고 생각해보자. 이때 Transend가 위의 그래픽 파일을 50KB로 줄이면서 color 수는 256 color로 size는 100x120pixel 정도로 줄였다고 가정한다. 위의 상황일 때 16M color까지 처리가 가능한 notebook으

로서는 자신이 처리할 수 있는 용량이 16M color에 성능이 W/S과 비슷함에도 불구하고, 품질이 떨어지는 그래픽 파일을 받아야 한다는 점이다. 그에 비해 240x320 16gray의 디스플레이를 가진 HPC인 경우 자신이 바로 보여줄 수 없는 그래픽 파일이 오기 때문에 색상들을 흑백으로 변환하고 size 변환에 걸리는 시간으로 인해 프로세싱 시간이 오래 걸린다는 문제점이 있다.

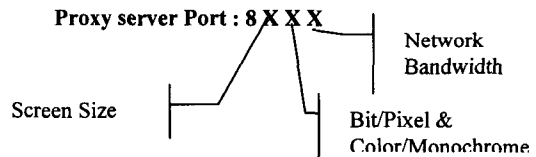
위와 같은 문제점들로 인해 이동컴퓨터의 H/W 사양에 맞게 그래픽 파일의 size를 줄일 수 있어야 한다. 즉, 기존의 Transend는 이동컴퓨터의 H/W에 관계 없이 그래픽 파일의 quality와 size를 어느 정도 줄이는가를 정했지만, 본 연구에서는 이동컴퓨터의 H/W에 맞도록 줄이면서, 확장성을 위해서 아래와 같이 동작하도록 한다.

- Proxy Server를 H/W 사양에 따라 다양하게 포트(Port)를 설정하고 여러 개의 Proxy Server가 작동되게 한다.
- Web browser는 자신의 H/W 사양에 따라 Proxy server의 포트를 설정한다.
- Web browser는 자신의 H/W 사양에 맞게 size가 줄여진 그래픽 파일을 Proxy server를 통해 받게 된다.

4. 설계와 구현

4.1. 포트(Port)를 이용한 모델의 설계

초기 모델은 현재 사람들이 가장 많이 사용하는 HTTP 1.0을 기반으로 하여, HTTP 1.1에서도 사용할 수 있도록 프로토콜을 바꾸지 않고 포트를 이용하는 Proxy server를 설계한다. 본 논문에서는 (그림 2)와 같은 방법으로 포트번호를 규정한다.



Nu m.	Screen size	Nu m.	Bit/Pixel	Nu m.	Network bandwidth
0	1024x768	0	16 / Color	0	none
1	640x480	1	8 / Color	1	2Mbps(W-LAN)
2	640x240	2	4 / Color	2	Modem(56Kbps)
3	240x320	3	1 / Color	3	not reserved
4	160x160	4	not reserved	4	not reserved
5	not reserved	5	16 / Mono.	5	not reserved
6	not reserved	6	8 / Mono.	6	not reserved
7	not reserved	7	4 / Mono.	7	not reserved
8	not reserved	8	1 / Mono.	8	not reserved
9	not reserved	9	not reserved	9	not reserved

(그림 2) 포트의 설정

(그림 2)에서 포트번호를 8로 시작하게 한 것은 HTTP의 예약된 포트번호가 80번이기 때문이고, 이동컴퓨터의 스크린 크기를 그 다음 번호로 규정한다. 여기에서 5 이상은 다양한 모니터의 설정을 위해 예약을 하지 않는다. 그에 비해 색상의 수와 흑백 유무는 한정적으로 하였는데, 이는 위의 색 범위가 보통 많이 사용하는 색 범위이고, 사용자 정의의 색상수를 설정하기 위해서 4.9번을 비워 놓았다. 마지막 자리 값은 네트워크의 대역폭을 의미한다.

예를 들면 123.24.31.1를 Proxy server로 사용하며 이동컴퓨터가 화면 크기 240x320이며, 칼라(Color)화면을 사용하면서 8bit/pixel을 쓰는 디스플레이를 가지고 있으며, 네트워크 인터페이스로는 W-LAN(Wireless LAN)을 사용한다고 가정한다

다. 그렇게 하면, 아래와 같은 환경설정으로 Proxy server를 설정할 수 있다.

Proxy Sever IP/ Port : 123.24.31.1/8311

4.2. 그래픽 파일 감소 프로그램(Distillation maker) 구현

대표적인 이동컴퓨터를 위한 Proxy server인 Transend의 초기 버전인 Pythia는 다음과 같은 옵션으로서 그래픽 파일의 크기를 줄인다.

- Color & Mono
- Colors : 8bit, 4bit, 2bit, 1bit
- Size : fit and original

위와 같이 데이터를 줄였을 때 다양한 이동컴퓨터의 화면 크기와 색에 맞추지 못하고 옵션은 Pythia 안의 정해진 값을 사용해야 한다. 따라서 이런 문제점을 해결할 수 있는 Distillation maker를 제작한다.

4.2.1. 제작 환경 및 설계

Java를 이용해 제작하였으며, Distillation maker의 옵션을 다음과 같이 하였다.

- 그래픽 파일 이름
- Color 수 (색상 수)
- 이동컴퓨터의 화면 크기
- Color or Gray

여기에서 나오는 그래픽 파일의 size는 폭과 높이 중 어느 하나가 30pixel을 넘지 않을 때는 파일을 줄이지 않았다.



원래의 Image (882x588 pixel)

Pythia의 경우



Color(8,4,2,1bits) size : 75x50 pixel

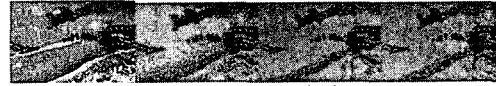


Gray (8,4,2,1 bits) size : 75x50 pixel

Improve된 Distillation maker의 경우



Color (8,4,2,1 bits) size : 551x367 pixel



Gray(8,4,2,1 bits), size : 551x367 pixel
(그림 3) Distillation의 비교

만약 그래픽 파일의 width가 1024 pixel을 넘을 때는 아래와 같이 한다.(1024x768화면을 표준 화면으로 가정).

$$X_{new} = X_{com}$$

$$Y_{new} = \frac{X_{new} \times Y_{img}}{X_{img}}$$

- Xnew : distillation되는 Image의 width
- Ynew : distillation되는 Image의 height
- Xcom : computer의 화면 크기 중 width
- Ximg : 원래 Image의 width
- Yimg : 원래 Image의 height

만약 Image의 width가 1024 pixel이하일 때는 아래와 같이 하였다.

$$X_{new} = \frac{X_{com} \times X_{img}}{1024}$$

$$Y_{new} = \frac{X_{new} \times Y_{img}}{X_{img}}$$

위와 같이 한다면 화면 크기를 꼭 채우거나 넘는 그래픽 파일은 화면에 맞게 줄어들면서, 화면 크기보다 작은 파일은 이전의 화면에 비례해서 줄이게 된다.

(그림 3)의 원래 그래픽 파일을 일반 컴퓨터에서 보는 것으로 가정해보자. 1024x768화면에서 882x588 pixel의 size로 보던 그래픽 파일을 Pythia는 640x320화면을 가진 이동컴퓨터라도 75x50정도의 작은 size로 보지만 개선된 Distillation maker는 화면비례에 맞추어 551x367의 size로 볼 수 있게 된다.

5. 결론

이상으로 다양한 이동컴퓨터를 위한 새로운 Proxy server를 설계하고 그의 중추적인 부분인 Distillation maker를 만들어 보았다. 여기에서 구현한 Distillation maker로써 다양한 이동컴퓨터 H/W의 환경에 맞게 줄여지게 된다. 그러므로, 다양한 디스플레이를 가진 이동컴퓨터에 일괄적으로 줄이던 기존 방식 보다 좋은 환경을 사용자에게 제공할 수 있다. 점차 이동컴퓨터가 다양하게 되면서 이와 같은 방법으로 사용자의 욕구를 충족시킬 수 있으리라 생각된다. 앞으로 이것을 기반으로 세분화된 포트를 계층적으로 규정하는 것과 본격적인 Proxy server로의 이식이 진행될 예정이다.

참고문헌

- [1] Armando Fox, "Information Delivery Infrastructure in the Mobile, Wireless Age - White paper", Proxinet co., 1999, <http://www.proxinet.com/>
- [2] Armando Fox, Ian Goldberg, Steven D. Gribble, David C. Lee, Anthony Polito and Eric A. Brewer, "Experience With Top Gun Wingman: A Proxy-Based Graphical Web Browser for the USR PalmPilot", Proc. IFIP International Conference on Distributed Systems Platforms and Open Distributed Processing (Middleware '98), Lake District, UK, Sept. 1998.
- [3] Armando Fox, Steven D. Gribble, Yatin Chawathe and Eric A. Brewer, "Adapting to Network and Client Variation Using Active Proxies: Lessons and Perspectives", IEEE Personal Communications, September 1998.
- [4] Armando Fox, S. Gribble, etc., "The Transend Service", Transend service homepage, 1998, <http://transend.cs.berkeley.edu/>
- [5] Stephanie Miles, "Huge leaps predicted for handhelds", CNET News.com, May 24, 1999.
- [6] Spyglass co. "Mobile Data Solution", Spyglass co., 1999, <http://www.spyglass.com/solutions/mobiledata/>
- [7] T. Imielinski, H.F. Korth, "Introduction to Mobile Computing", Mobile Computing, Kluwer Academic Pub., page 1, 1996.