

WAP 프록시의 캐시 성능 분석

이종열^o, 노강래, 이정은, 신동규, 신동일
 세종대학교 컴퓨터 공학과
 e-mail:{leemaster}@gce.sejong.ac.kr

The Performance Analysis of Caching in a WAP Proxy

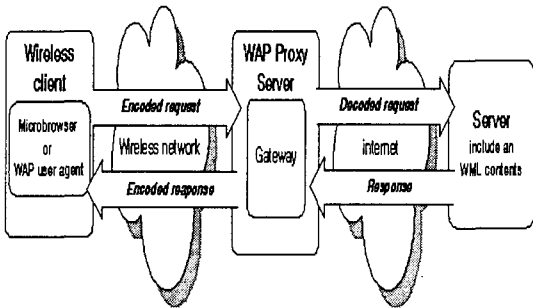
Jong-youl Lee^o, Kang-rae Noh, Jung-eun Lee,
 Dong-kyoo Shin, Dong-il Shin
 Dept of Computer Science, Sejong University

요약

무선 어플리케이션 서비스 시스템 구축시 가장 중요한 역할을 담당하는 장비는 WAP Proxy 서버이다. 캐시 기능이 추가된 WAP Proxy 서버는 캐시 서버와 Gateway가 연동하여 해당 URL의 Request에 대한 응답 시간(response time)을 줄이기 위해 구현된 시스템이다. 본 논문은 캐시 기능을 이용한 WAP Proxy 서버의 성능을 측정하기 위해, 별도로 구현한 성능 측정 시스템을 이용, 그 성능을 비교 분석했다. 성능 측정 시스템은 C로 구현했으며, 실시간으로 그 결과를 확인 할 수 있다.

1. 서론

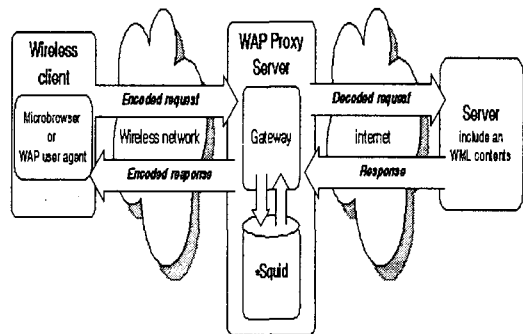
WAP[1] Proxy 서버는 네트워크상의 낮은 대역폭을 효율적으로 활용하기 위해 웹 콘텐츠를 이진 데이터로 인코딩 하여 무선 단말기에 전달한다[2]. 이는 전송에 따른 오버헤드를 줄임과 동시에 무선 콘텐츠를 비교적 빠른 속도로 이용할 수 있게끔 해 준다.



(그림1) WAP Proxy 구성도

(그림1)은 무선 응용 시스템 구축 시, WAP Proxy 서버의 기능을 보여준다. WAP Proxy 서버는 Gateway, 인코더, 디코더로 구성되어있다. 성능 측정을 위한 WAP Proxy에는 캐시서버가 추가된다.

따라서 WML URL의 Request시, 그 응답시간을 줄임으로서 보다 더 빠르게 무선 서비스를 제공 할 수 있다. 캐시기능이 추가된 WAP Proxy 서버의 성능을 측정하기 위해서 별도의 성능 측정 시스템을 구현했으며, 그 결과는 애플릿을 통해 실시간으로 보고된다. 본 논문의 구성은 다음과 같다. 2장에서는 캐시서버의 기능을 갖는 WAP Proxy 서버에 대해 간략히 소개하고, 3장은 성능 측정 시스템의 소개, 4장은 성능 측정 결과를 분석하며, 5장에서 결론 및 향후 과제를 기술한다.



(그림2) 캐시 서버 구조를 갖는 WAP Proxy 구성도

2. 캐시 서버 기능을 갖는 WAP Proxy 서버

2.1. Kannel Gateway

WAP Gateway로는 Kannel Gateway를 사용했다. Kannel은 Wapit[3]에서 WAP Gateway 개발을 목적으로 시작한 Open Source 프로젝트이다. Kannel WAP Gateway는 무선 어플리케이션 서비스를 위한 기반 기술의 필수적인 요소를 제공하며, WAP 과 SMS Gateway의 역할을 동시에 수행한다.

2.2. Proxy 서버

Proxy 서버는 일반 사용자(Clients)를 대신하여 requests를 생성하거나 처리할 목적으로 만들어진 중간 계층의 프로그램으로서 사용자와 Server 두 가지 역할을 동시에 수행하는 시스템이다.[4]

2.2.1. Proxy 서버의 중요한 용도

1) 방화벽 내부에 있는 사용자들이 외부 인터넷에 접속하도록 허용하는 역할을 사용자들의 요청을 대신해서 방화벽 외부에 있는 실제 인터넷 서버에 전달하는 역할

2) 웹 오브젝트 등을 자신의 로컬 디스크 영역에 캐싱 하는 기능

Proxy Server의 다양한 기능 중에서 웹과 관련하여 사용자를 대신하여 인터넷 상의 웹 서버에 대한 요청을 생성하거나, 처리하는 역할을 수행함과 동시에 처리한 웹 오브젝트를 향후 발생 가능한 다른 사용자의 요청에 대처하기 위해 자신의 Local 디스크에 저장하는 기능을 수행하는 특화된 Web Cache인 Wap Proxy Server로 사용할 것이다.

2.2.2. Web 캐시의 역할

1) 사용자들의 웹 오브젝트에 대한 요청을 감지하여, 사용자 대신 실제 웹 서버에 오브젝트를 요청하여 사용자에게 전달함과 동시에 받은 오브젝트를 향후 사용에 대비하여 자신의 로컬 디스크 영역에 캐싱 한다.

2) 같은 오브젝트에 대한 다른 사용자의 요청이 감지되면, 실제 웹 서버에게 해당 오브젝트에 대한 요청 작업을 하지 않고 자신의 로컬 디스크에 캐싱 되어 있던 오브젝트를 대신 전달해 준다.

2.2.3. Web Cache를 사용하는 중요한 3 가지 이유

1) Hit Ratio 향상[5]

2) To Reduce Latency[5]

- 사용자의 요청을 처리하는 시간이 줄어든다.
- 일반 사용자는 웹 상의 오브젝트를 요청했을

때 빠른 응답을 원한다. 따라서 캐시 서버를 사용함으로써 이러한 사용자들의 욕구를 해결해 줄 수 있다.

3) To Reduce Traffic

- 네트워크 Bandwidth의 비용을 절감한다[5].

2.2.4. Web Cache 사용의 장점[6]

Fast Access : 사용자 네트워크 사용 시간 감소

Efficiency : 네트워크 대역폭의 효과적인 절약

Load balancing : 서버 측 부하 감소

사용자 로컬 디스크의 절약

고용량의 컨텐츠의 빠른 전송

2.3. Squid[7]

Squid 는 고퍼, FTP, HTTP를 지원하기 위한 고 수준의 캐시 서버이다. 전형적인 캐시 프로그램과는 달리 Squid는 non-blocking, I/O-driven 처리에 대한 모든 요구를 다룰 수 있다. Squid는 Kannel 과 마찬가지로 open source 프로젝트이며, ARPA 기금의 지원 하에 미 콜로라도 대학의 Harvest 프로젝트로부터 분리되었다.

2.4. 캐시 서버와 Kannel Gateway와의 연동

Kannel WAP Gateway를 설치하면 설치 디렉토리에 실행 파일과 함께 wapkannel.conf 라는 설정 파일이 생성된다. 이 설정 파일을 통해 내부의 각 모듈간의 통신을 위한 포트 및 IP를 설정할 수 있으며, 외부 Proxy 서버의 IP와 포트도 설정할 수 있다[8]. (그림3)에서 박스 안의 부분이 Squid와의 연동을 위해 설정한 부분이다.

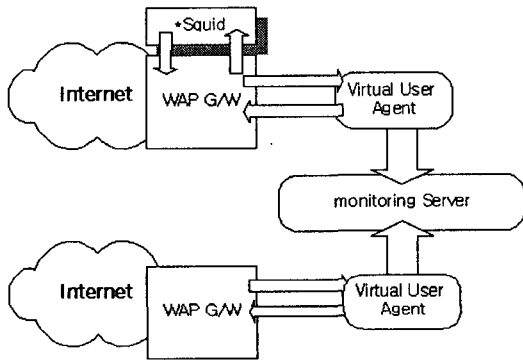
```
admin-port = 13000
wapbox-port = 13002
admin-password = bar
wdp-interface-name = "*"
#log-file = "/tmp/kannel.log"
#log-level = 0
box-deny-ip = "*. *.*.*"
box-allow-ip = "127.0.0.1"
#admin-deny-ip = ""
#admin-allow-ip = ""
#access-log = "access.log"
http-proxy-host="210.107.236.188"
http-proxy-port="3128"
```

```
group = wapbox
bearerbox-host = localhost
#log-file = "/tmp/wapbox.log"
#log-level = 0
syslog-level = none
```

(그림3) wapkannel.conf

3. 성능 측정 시스템 소개

Virtual User Agent는 1초에 두 번씩, 지정된 WAP Proxy에 URL Request를 보낸다. 또한 Request에 대한 응답을 WAP Proxy로부터 받아 응답 시간을 체크하여 Monitoring Server로 50 Request마다 주기적으로 보낸다. Monitoring Server는 Virtual User Agent로부터 받은 데이터(응답시간, 데이터량, URL)를 Queue에 저장하고 웹 상의 애플릿으로부터 요청이 오면 데이터를 Queue에서 꺼내 보내준다.



(그림4) 성능 측정 시스템의 구성도

4. 성능 측정 결과 분석

4.1. WAP Proxy의 성능 측정

WAP 시스템의 성능 측정 기준중 가장 중요한 두 가지는 Latency와 WAP 콘텐츠의 처리율(Throughput)이다[9]. 처리율 계산 공식은 (그림5)와 같다.

$$\text{Class Throughput} = \frac{\text{total request} \times \text{class percentage} \times \text{average size}}{\text{Observation period}}$$

(그림5) 단위 시간당 처리율 (Throughput)

Total request는 전체 요청 횟수를 의미한다. Class percentage는 해당 Class(예, HTML, Image, Video)가 전체 request에서 차지하는 비율을 나타낸다. 본 논문에서는 WML 문서 페이지만 테스트를 한정시켰기 때문에 1로 한다. average size는 평균 파일 사이즈를 의미한다. Observation period는 관측 시간을 의미하며, 요청에 대한 응답이 전부 도착했을 때까지, 걸린 시간을 기준으로 택했다.

4.2. 성능 측정 시나리오.

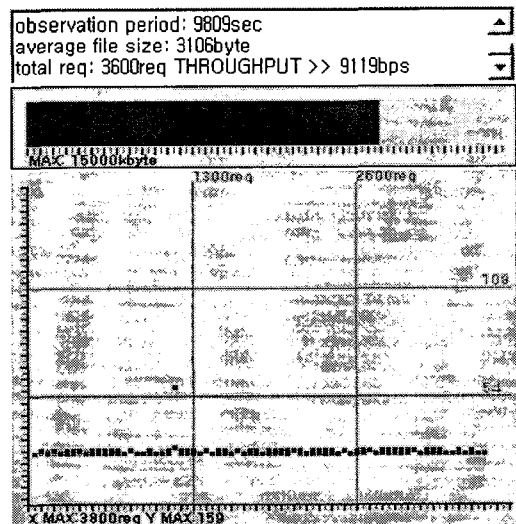
4 개의 WML 사이트가 각각 1개씩 4개의 Virtual User Agent에 배정된다. 각 Virtual User Agent는 Site당 2번씩 3600번의 URL Request를 보낸다. 4개의 Site URL은 다음과 같다.

- http://0.co.kr/bm/
- http://www.m0m.co.kr/wap114/index.wml
- http://210.107.236.185/wml/proxy/test.wml
- http://wap.oni.game.com/

성능 측정을 위해서는 2대의 서버가 필요하다. 첫 번째 서버는 Squid 및 Kannel WAP Gateway가 설치되어야 하고, 두 번째 서버는 Kannel WAP Gateway만 설치한다. 양쪽 서버에는 성능 측정을 위한 프로그램이 설치되어야 한다. 실행 환경은 Squid가 설치될 WAP Proxy 서버의 경우, 300MHz의 펜티엄II CPU, 512이상의 RAM, 9GB이상의 하드용량이 요구된다.

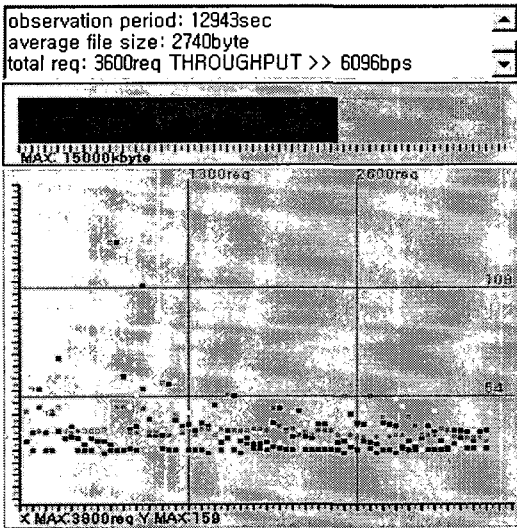
4.3. 성능 측정 결과 및 분석

측정 시간은 네트워크 트래픽이 심한 새벽 시간대(12:00-13:00)로 정했다. (그림6)(그림7)은 성능 측정 시스템의 구동 후, 측정 결과를 보여준다. (그림6)은 캐시 서버를 사용한 경우이며, (그림7)은 캐시 서버를 사용하지 않은 경우이다. 점의 색깔은 각



(그림 6) 캐시를 이용한 WAP Proxy의 성능 측정 결과 화면.

Site마다 고유하다. 데이터 량을 보면(막대 그래프), 현격한 차이를 보이는데, 이는 응답 데이터의 손실을 의미한다. 그림에서 X축은 Request횟수를 Y축은 응답 시간을 나타낸다. 응답 시간은 Virtual User Agent에서 50회의 Request를 보내고 부터, 해당 응답이 전부 도착하기까지의 걸린 시간을 의미한다.



(그림 7) 캐시를 이용하지 않은 WAP Proxy 성능 측정 결과 화면

응답 시간 면에서 보면, 캐시 서버를 사용한 WAP Proxy가 모든 사이트에서 일정한 응답 시간을 보여 준다. 처리율은 계산 결과 캐시 서버를 사용한 WAP Proxy가 9119bps가 나왔으며, 사용하지 않은 곳은 6096bps가 나왔다. 따라서 WAP Proxy의 성능이 캐시 서버를 사용한 경우에 50%정도의 향상된 결과를 가져왔다.

5. 결론 및 향후 연구과제

WAP Proxy에 캐시 서버를 추가하여, 기존의 방식보다 더 좋은 성능을 얻을 수 있었다. 이를 활용한다면, 네트워크 트래픽에 상관없이 안정적으로 WML 콘텐츠를 제공 할 수 있을 것이다. 이후 연구과제는 WAP Proxy 서버에 모니터링 기능을 추가하여, 보다 효율적으로 WAP 시스템 관리를 실현하는 방향과 무선 콘텐츠에 적합한 캐시 모델에 대한 연구이다.

참고문헌

[1][WAP]"Wireless Application Protocol Architecture Specification", WAP Forum, April 30, 1998. URL : <http://www.wapforum.org/>

[2] Leavitt , Will wap deliver the wireless internet , Volume: 33, Page(s): 16 -20 , 5 May 2000.

[3] <http://www.wapit.com/>

[4] MAJ Richard Howard, Directorate of Information Management, MAJ Bernard J. Jansen Department of EE & CS : A Proxy Server Experiment : an Indication of the Changing Nature of the Web

[5] Anja Feldmann Ramón Cáceres Fred Douglis Gideon Glass Michael Rabinovich AT&T Labs-Research, Florham Park, NJ,USA : Performance of Web Proxy Caching in Heterogeneous Bandwidth Enviroments, pages 110-112

[6] Jia Wang Cornell Network Research Group : A Survey of Web Caching Schemes for the Internet

[7] <http://www.squid.org/>

[8] Lars Wirzenius , Kannel 1.0.2 User's Guide , Volume: 43, Page(s): 23-24

[9] Daniel A. Menasce , Virgilio A.F. Almeida , Capacity Planning for Web performance , Page(s): 82-83