

WAP 프록시 서버의 캐싱 성능 분석

노강래^o, 이종열, 장지산, 신동규, 신동일
 세종대학교 컴퓨터공학과
 e-mail:(knoh, leemaster, jsjang, dshin, shindk}@gce.sejong.ac.kr

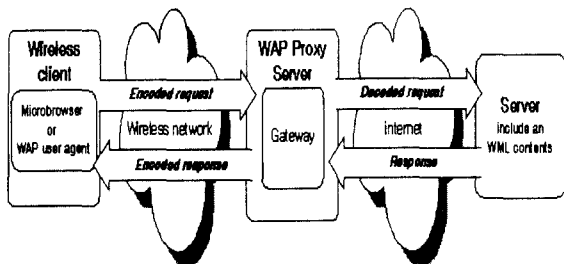
The Performance Analysis of Caching in a WAP Proxy Kang-rae Noh, Jong-youll Lee, Ji-san Jang, Dong-kyoo Shin, Dong-il Shin Department of Computer Engineering, Sejong University

요 약

무선 어플리케이션 서비스 시스템 구축시 가장 중요한 역할을 담당하는 장비는 WAP(Wireless Application Protocol) 프록시 서버이다. WAP 프록시 서버는 WAP 프로토콜과 인터넷 TCP/IP 프로토콜을 중간에서 변환해 주는 역할을 한다. 따라서, 무선 단말기로부터 요청은 WAP 프록시 서버를 통해 유선망의 서버로 보내지게 되고, 서버로부터 보내진 무선 콘텐츠는 다시 WAP 프록시 서버를 통해 무선 단말기로 전해진다. 이때, 무선 콘텐츠가 WAP 프록시 서버에 캐싱된다면, 동일한 무선 콘텐츠 요청시, 유선망을 거치지 않고도 WAP 프록시 서버에서 무선 콘텐츠를 서비스 할 수 있을 것이다. 본 논문에서는 WAP 프록시 서버에 캐싱 기능을 추가하여, 그 성능을 분석하였다. 성능 측정을 위해 C와 서블릿으로 측정 모듈을 작성 했으며, 웹 브라우저를 통해 실시간으로 측정 결과를 확인 가능하도록 하였다.

1. 서론

WAP[1] 프록시 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 이용 할 수 있도록 해 준다.[그림1]은 무선 응용 시스템 구축시, WAP 프록시 서버의 기능을 보여준다.



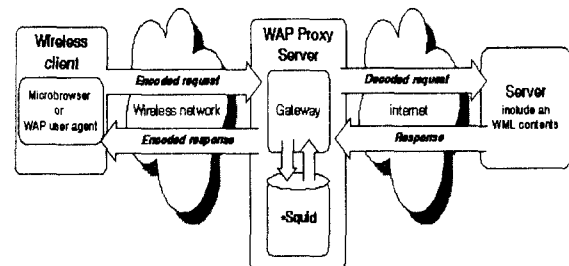
[그림1] WAP 프록시 구성도

WAP 프록시 서버는 게이트웨이(Gateway), 인코더(Encoder), 디코더(Decoder)로 구성되어있다. 문서 검색 속도의 향상을 위하여 WAP 프록시 서버에는 캐싱 기능의 추가가 가능하며, WML URL의 요청시, 그 응답시간을 줄임으로서 보다 빠른 무선 서비스를 제공 할 수 있다. 캐싱 기능을 추가한 WAP 프록시 서버의 성능을 측정하기 위하여 성능 측정을 위한 모듈을 구현했으며, 측정 결과를 애플릿을 통해 실시간으로 조회하도록 하였다. 본 논문의 구성은 다음과 같다. 2장에서는 캐싱 서버의 기능을 갖는 WAP 프록시 서버에 대해 간략히 소개하고, 3장은 성능 측정 시스템의 소개, 4장은 성능 측정

결과를 분석하며, 5장에서는 결론 및 향후 연구과제를 제시한다.

2. 캐싱 서버 기능을 갖는 WAP 프록시 서버

WAP 프록시 서버의 캐싱 기능의 성능측정을 위하여 [그림2]에 보는 바와 같이 WAP 프록시 서버를 구성하였



[그림2] 캐싱 기능을 가지는 WAP 프록시 서버의 구성도

다. WAP 프록시 서버로는 Kannel 게이트웨이를 이용하였으며, WAP 프록시 서버의 입출력 데이터의 캐싱을 위하여 캐싱 서버인 Squid를 이용하였다.

2.1 Kannel 게이트웨이

WAP 게이트웨이는 Kannel 게이트웨이를 사용했다. Kannel은 Wapit[3]에서 WAP 게이트웨이 개발을 목적으로 시작한 오픈 소스 프로젝트이다. Kannel WAP 게이트웨이는 무선 어플리케이션 서비스를 위한 기반 기술의 필수적인 요소를 제공하며, WAP 과 SMS(Short Message Service)게이트웨이의 역할을 동시에 수행한다.

2.2 Squid

Squid[4]는 Gopher, FTP, HTTP를 지원하기 위한 고수준의 캐싱 서버이다. 전형적인 캐시 프로그램과는 달리 Squid는 non-blocking, I/O-driven 처리에 대한 모든 요구를 다룰 수 있다. Squid는 Kannel과 마찬가지로 오픈 소스 프로젝트이며, ARPA 기금의 지원 하에 미 콜로라도 대학의 Harvest 프로젝트로부터 분리되었다.

2.3 캐싱 서버와 Kannel 게이트웨이와의 연동

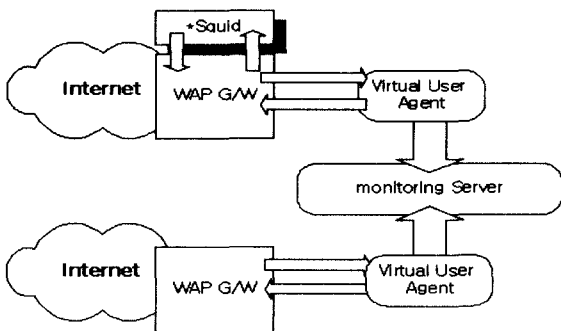
Kannel WAP 게이트웨이를 설치하면 설치 디렉토리에 실행 파일과 함께 wapkannel.conf 라는 설정 파일이 생성된다. 이 설정 파일을 통해 내부의 각 모듈간의 통신을 위한 포트 및 IP를 설정할 수 있으며, 외부 프록시 서버의 IP와 포트도 설정할 수 있다[5]. [그림3]에서 박스안의 부분이 Squid와의 연동을 위해 설정한 부분이다.

```
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



[그림4] 성능 측정 시스템의 구성도

Virtual User Agent는 WAP 콘텐츠를 요청하는 WAP 애플레이터의 역할을 담당한다. Virtual User Agent는 1

초에 두 번씩, 지정된 WAP 프록시 서버에 URL 요청을 보낸다. 또한, 요청에 대한 응답을 WAP 프록시 서버로부터 받아 응답시간을 점검하여 모니터링 서버로 측정된 데이터(데이터량,경과시간,URL)를 주기적으로 보낸다. 모니터링 서버는 Virtual User Agent로부터 받은 데이터를 큐(Queue)에 저장할 하고 웹상의 애플릿으로부터 요청이 오면 데이터를 큐에서 꺼내 보내준다.

4. 성능 측정 결과 분석

4.1 WAP 프록시의 성능 측정

WAP 시스템의 성능 측정 기준 중 가장 중요한 두 가지는 지연시간(Latency)과 콘텐츠의 처리율(Throughput)이다[6]. 처리율 계산공식은 [그림5]와 같다. 이 공식에서 Total request는 전체 요청횟수를 의미한다. Class percentage는 해당 Class(HTML,Image,Video)가 전체 요청에서 차지하는 비율을 나타낸다. 본 성능 측정에서는

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

[그림5] 단위 시간당 처리율 (Throughput)

이미지가 없는 WML 문서로 테스트를 한정되었기 때문에 100%로 했다. Average size는 평균 파일 사이즈를 의미한다. Observation period 는 관측 시간을 의미하며, 요청에 대한 응답이 전부 도착했을 때까지 걸린 시간을 기준으로 택했다.

4.2 성능 측정 시나리오

4개의 무선 사이트가 각각 1개씩 4개의 Virtual User Agent에 배정된다. Virtual User Agent는 아래의 사이트로 각각, 1초에 2번씩 30분간 3600번의 URL 요청을 보낸다. 무선 사이트의 URL은 다음과 같다.

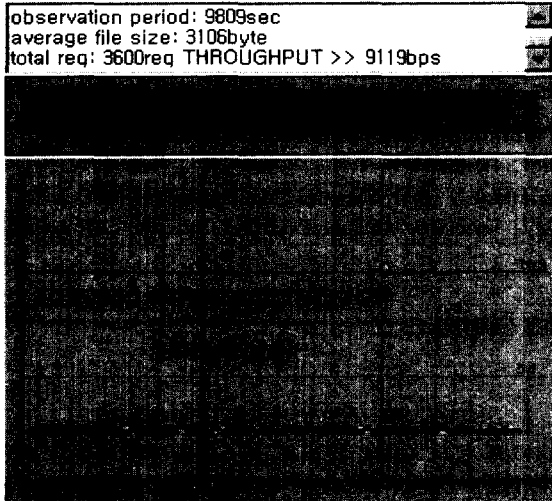
- ◆ http://0.co.kr/bm/
- ◆ http://www.m0m.co.kr/wap114/index.wml
- ◆ http://210.107.236.185/wml/proxy/test.wml
- ◆ http://wap.oni.game.com/

성능 측정을 위해서는 두 대의 서버가 필요하다. 첫 번째 서버는 Squid 및 Kannel WAP 게이트웨이가 설치되어야 하고, 두 번째 서버는 Kannel WAP 게이트웨이만 설치한다. 양쪽 서버에는 성능 측정을 위한 측정 모

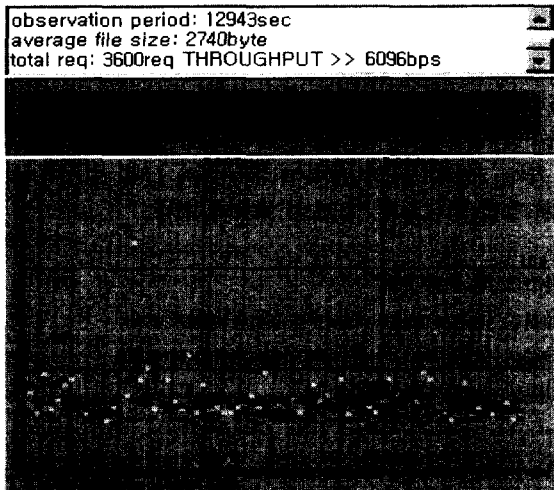
들을 설치한다. 실행 환경은 Squid가 설치될 WAP 프록시 서버의 경우, 펜티엄II 300MHz이상의 CPU, 512메가 이상의 램, 9GB이상의 하드용량이 요구된다.

4.3 성능 측정 결과 및 분석

측정 시간은 네트워크 트래픽이 심한 새벽 시간대(00:00-01:00)로 정했다. [그림6][그림7]은 성능 측정 시스템의 구동 후, 측정 결과를 보여준다. [그림6]은 캐싱 서버를 사용한 경우이며, [그림7]은 캐싱 서버를 사용하지



[그림6] 캐싱 기능을 사용한 WAP 프록시 서버의 성능 측정 결과 화면



[그림7] 캐싱 기능을 사용하지 않은 WAP 프록시 서버의 성능 측정 결과 화면

않은 경우이다. 각 사이트로부터의 응답시간을 표현하기 위하여 각각 다른 색깔의 점을 이용하여 그래프로 나타냈다. 그래프에서 X축은 요청횟수를, Y축은 응답시간을 나타낸다. 응답 시간은 Virtual User Agent에서, 50회의 요청을 보내고 난 후부터, 요청에 대한 응답이 전부 도착하기까지의 걸린 시간을 의미한다. URL 요청에 대한 응답 시간은 캐싱 기능을 추가한 WAP 프록시 서버가 모든 사이트에서 거의 일정한 응답 시간을 보여준다. 처리율은 캐싱 기능을 사용한 WAP 프록시 서버가 9119bps가 나왔으며, 캐싱 기능을 사용하지 않은 WAP 프록시 서버의 경우 6096bps가 나왔다. 따라서, WAP 프록시의 성능은 캐싱 기능을 사용한 경우, 50%정도의 향상된 결과를 가져왔다.

5. 결론 및 향후 연구과제

WAP 프록시 서버에 캐싱 기능을 추가하여, 캐싱 기능이 없는 WAP 프록시 서버의 경우보다 처리를 면에서 더 좋은 성능을 얻을 수 있었다. 이를 활용한다면, 네트워크 트래픽에 상관없이 비교적 안정적인 무선 콘텐츠 서비스를 제공할 수 있을 것이다. 향후 연구 과제는 WAP 프록시 서버에 모니터링 기능을 추가하여, 보다 효율적으로 WAP 시스템 관리를 실현하는 방향과 무선 콘텐츠에 적합한 캐싱 모델에 대한 연구이다.

6. 참고 문헌

- [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] <http://www.squid.org/>
- [5] Lars Wirzenius , Kannel 1.0.2 User's Guide , Volume: 43, Page(s): 23-24
- [6] Daniel A. Menasce , Virgilio A.F. Almeida , Capacity Planning for Web performance , Page(s): 82-83