

모바일 응용 서버에서 컨텐츠 변환을 위한 캐쉬 관리자의 설계

장철수, 김성훈, 변영철, 장민수
한국전자통신연구원 전자거래연구부
e-mail : {jangcs, saint, ycb, minsu}@etri.re.kr

A Design of Cache Manager for Contents Transcoding in Mobile Application Server

Choul-Soo Jang, Seong-Hoon Kim, Yeong-Cheol Byeon, Min-Su Jang
Electronic Commerce Department
Electronics and Telecommunications Research Institute

요 약

무선 인터넷 환경에서 무선 인터넷 브라우저를 지원하는 휴대폰 단말기는 각 단말기 종류별로 표시 가능한 줄 수, 화면 넓이, 칼라 지원 여부 등과 같이 다양한 표현 능력을 갖고 있음에 따라, 휴대폰 단말기를 통해 클라이언트에 의해 요청된 해당 콘텐츠는 각 단말기의 표현 능력에 따라 응용 서버에 의해 적절하게 변환되어 클라이언트에게 제공되어야 한다. 그러나 변환 작업은 처리 시간이 매우 많이 소모되는 작업이므로 캐쉬를 이용하는 방법이 효율적이다. 그러나, 무선 인터넷 환경에서는 다양한 종류의 휴대폰 단말기가 존재하므로 변환된 콘텐츠는 오직 그 콘텐츠를 요청한 단말기에만 유효하도록 캐쉬에 저장되기 때문에 캐쉬에 대한 적중률이 매우 떨어질 수 있는 문제점이 있다. 본 논문에서는 무선 인터넷 환경에서의 이러한 캐쉬의 약점을 극복하기 위해, 모바일 환경의 응용 서버에서 다양한 단말기를 위한 콘텐츠 변환을 지원하면서도 캐쉬의 적중률을 높일 수 있는 방법을 제안하며 이에 대한 설계를 논한다

1. 서론

최근 이동통신 기술의 발전에 따라 개인용 컴퓨터 뿐만 아니라 다양한 유형의 모바일 단말기까지 포함하는 유무선 통합 인터넷 서비스 환경으로의 변화가 급격히 일어나고 있다. 이러한 무선 인터넷 환경에서의 캐쉬 관리자는 대용량 모바일 응용 서버에서 사용자에게 보다 빠른 응답 시간을 보장하기 위해 한번 처리된 페이지에 대한 정보를 캐쉬에 저장하여 재요구 시에는 캐쉬에 저장된 정보를 전송한다. 또한 변환 규칙에 따라 변환기에 의해 이미 변환 완료된 이미지 데이터 등에 대해서는 불필요한 콘텐츠 변환 과정을 생략하여 전체 페이지의 생성 시간을 줄여 응답 속도를 개선할 수 있다. 그러나, 무선 인터넷 환경에서는 다양한 종류의 단말기가 존재하므로 각 단말기 종류마다 변환된 콘텐츠를 캐쉬에 저장해 놓는다면 캐쉬의 적중률(hit rate)을 떨어뜨려 캐쉬의 효율성이 매우

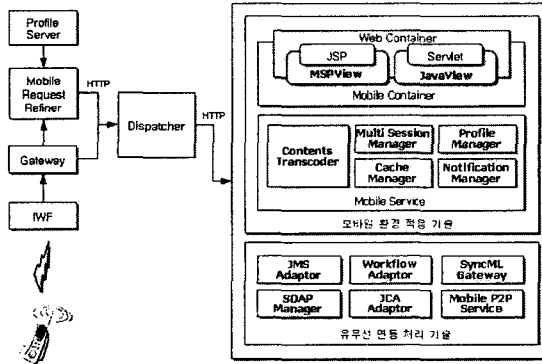
떨어진다. 본 논문에서는 이러한 무선 인터넷 환경에서 다양한 단말기를 위한 변환을 지원하면서도 캐쉬의 적중률을 높일 수 있는 방법을 제안하며 이에 대한 설계 내용을 살펴본다. 이 논문의 구성은 다음과 같다. 2 장에서는 유무선 통합의 모바일 응용 서버의 개념에 대해서 설명하고, 3 장에서는 모바일 응용 서버에서의 콘텐츠 변환에 대해 살펴봄, 4 장에서는 캐쉬 관리자의 설계 내용을 설명하고, 5 장에서 결론을 맺는다.

2. 유무선 통합 모바일 응용 서버의 개념

이동통신 기술의 발전에 따라 비교적 넓은 대역폭을 갖는 셀룰러 폰, PDA, Post PC 등과 같은 모바일 단말기들이 광범위하게 보급되어 활용되고 있다. 2003년 까지 인터넷 접근의 50%가 PC가 아닌 휴대용 단말기를 이용하여 접속할 것으로 예상되며[2], 2005년까

지 전 세계적으로 10 억의 사용자가 모바일 단말기를 이용하여 인터넷에 접속할 것으로 보인다[3]. 모바일 단말기를 중심으로 한 인터넷 서비스 환경의 변화가 일고 있는 것이다. 이처럼, 개인용 컴퓨터를 이용한 네트워크 서비스 환경에서 개인용 컴퓨터뿐만 아니라 다양한 유형의 모바일 단말기까지 포함하는 유무선 통합 인터넷 서비스 환경으로 변화가 일어나고 있다.

이러한, 환경 변화에 따른 새로운 형태의 모바일 서비스의 등장은 필연적이다. 이를 반영이라도 하듯, 굿모닝 증권에 따르면 국내 무선 인터넷 서비스 업체의 수는 2001년 2400 개에서 2003년 13 만 개로 기하급수적으로 증가하는 추세에 있으며, 이처럼 모바일 서비스에 대한 요구가 증가함에 따라 특정 기능 위주의 모바일 지원 기술이 개발되고 있지만, 무선 인터넷 서비스를 효과적으로 제공하려면 기존의 유선 인터넷을 기반으로 구축되어 운영중인 서비스를 재개발하지 않더라도 모바일 환경에 쉽게 적용되도록 하기 위한 유무선 통합 모바일 지원 기술 개발이 절대적으로 필요하다. 모바일 비즈니스 서비스는 기존의 인터넷 환경에서 기업 내부 비즈니스나 기업간 비즈니스 등 유선 인터넷 서비스를 포함하며, 무선 환경에서의 효과적인 서비스를 위해 무선 단말기의 제약성과 이동 통신망의 간헐적 단절성을 극복할 수 있는 비즈니스 서비스를 제공할 수 있어야 한다[1].



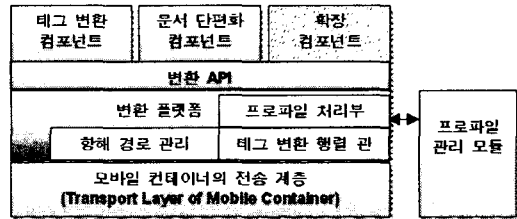
<그림 1> 유무선 통합 모바일 응용서버 구성도

3. 콘텐츠 변환

무선 인터넷 환경에서 무선 인터넷 브라우저를 지원하는 휴대폰 단말기는 각 단말기 종류별로 표시 가능한 줄 수, 화면 넓이, 칼라 지원 여부 등과 같이 다양한 표현 능력을 갖고 있음에 따라, 휴대폰 단말기를 통해 클라이언트에 의해 요청된 해당 콘텐츠는 각 단말기의 표현 능력에 따라 응용 서버에 의해 적절하게 변환되어 클라이언트에게 제공되어야 한다.

이러한 변환 작업을 위해서 W3C 에서의 장치 독립성과 관련하여 다양한 권고안을 만들어내고 있다. 그 중 대표적인 것이 CC/PP(Composite Capability/Preference Profile)이다[4]. CC/PP 는 장치의 기능 특성과 사용자의 콘텐츠 표시 선호 정보를 표현하기 위한 프로파일(Profile) 기술 언어이다. 여기서 프로파일이란

장치, 네트워크, 사용자 등 웹 콘텐츠의 전송에 관련된 요소들에 대한 정보 모음을 가리킨다. 콘텐츠를 클라이언트 장치에 적합한 형태로 변환하기 위해서는 프로파일 정보가 필수적이다. WAP 진영에서는 이미 CC/PP 를 채용하여 WAP 장치의 특정 정보를 기술하는 표준안을 UAProf(User Agent Profile)이란 이름으로 제정한 바 있으며, 앞으로 장치 독립성 분야에서 CC/PP 의 채용은 더욱 보편화될 전망이다. 모바일 응용 서버의 변환 모듈은 프로파일 정보 처리를 위해 CC/PP 를 채용함으로써 장래 웹 표준 환경에 대응할 수 있도록 하고 있다. 모바일 응용 서버에서는 클라이언트로부터 전달된 요청 메시지의 헤더를 분석하여 CC/PP 를 생성하여 콘텐츠 변환에 사용하며, 이를 기반으로 본 논문에서 설계된 콘텐츠 변환의 구성도는 <그림 2>와 같다.



<그림 2> 콘텐츠 변환 모듈의 구성도

4. 캐시 관리자의 설계

CC/PP 와 같은 프로파일 정보를 이용하여 콘텐츠를 변환하는 방법은 변환 시점에 따라 크게 2 가지로 분류할 수 있다.

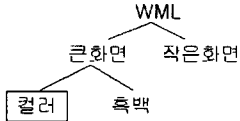
그 첫번째는 응용 서버에서 페이지를 생성하기 위해 실행되는 JSP 나 서블릿과 같은 컴포넌트에서 직접 CC/PP 의 내용을 참조하여 원본 콘텐츠를 생성하는 시점에 콘텐츠의 내용을 변경하게 하여 컴포넌트의 실행 결과가 바로 타겟 콘텐츠가 되는 방법이 있고, 두번째 방법은 페이지를 생성하기 위한 컴포넌트에서는 CC/PP 의 내용을 참조하지 않고 xHTML 과 같이 장치 독립적인 중간 단계의 원본 콘텐츠를 생성하고 이후에 생성된 원본 콘텐츠를 변환기를 통해 타겟 콘텐츠로 변환하도록 하는 방법이 있다.

첫번째 방법은 구현의 간단한 측면이 있으나, 변환 코드 자체가 수 많은 페이지생성 컴포넌트에 하드 코딩되므로 기술의 발달로 새로운 종류의 단말기가 출시되는 경우 수 많은 컴포넌트의 하드 코딩을 일일이 수정해야 하는 등 확장성이 떨어져서 관리의 어려움이 예상되며, 또한 변환 처리가 페이지를 생성하는 JSP 와 같은 컴포넌트내에서 개발자 임의대로 사용되기 때문에 페이지 생성 컴포넌트에 의해서 생성된 콘텐츠는 오직 해당 프로파일의 단말기에만 사용될 수 있다. 즉, 프로파일 내의 어떤 정보를 이용해서 콘텐츠를 생성하였는지 응용 서버가 알 수 있는 방법이 없으므로 비슷한 형식의 다른 단말기에도 전혀 사용될 수 없으며, 이와 같은 변환 방법에 캐시를 적용하

게 되면 캐쉬의 적중률(hit rate)이 매우 낮을 수 밖에 없어 캐쉬의 기능이 유명무실해 질 수 있다.

여러가지 디바이스 프로파일일지라도 보통은 몇개의 프로파일 범주내에 속하게 된다. 따라서, 같은 프로파일 범주의 다른 디바이스에 의해 요청되어 변환된 타겟 콘텐츠를 캐쉬하고 있다가 전달해 주는 것이 효율적일 것이다.

따라서, 본 논문에서는 원본 콘텐츠의 생성과 타겟 콘텐츠의 생성을 분리하도록 하여, JSP 나 서버릿과 같은 페이지 생성 컴포넌트에서는 XHTML 과 같은 장치 독립적인 중간 형태의 콘텐츠를 생성하도록 하며, CC/PP 의 내용을 이용하여 디바이스 프로파일을 <그림 3>처럼 계층적인 구조로 프로파일 그룹을 관리하여, 각 프로파일 그룹마다 변환 규칙을 작성하고, 해당 그룹에 속한 단말기에서 요청된 콘텐츠는 해당 그룹의 변환 규칙이 적용되어 타겟 콘텐츠를 생성하도록 한다.



<그림 3> 계층적인 프로파일 그룹의 예제

각 그룹 노드의 변환 규칙은 부모 노드의 변환 규칙을 상속받도록 하여 변환규칙의 작성 노력을 줄이도록 하고 있다.

무선 인터넷 환경에서 모바일 응용 서버의 캐쉬 기능은 크게 공유 기능과 문서 단편화 기능으로 분류할 수 있다.

공유 기능이란 임의의 클라이언트에게 제공했던 문서를 캐쉬에 저장하고, 다른 클라이언트가 같은 문서를 요청하는 경우 캐쉬에 저장되어 있는 문서를 돌려주어 성능을 향상시키는 기능이다. 이를 위해 프로파일 그룹을 관리하며 각 그룹 마다 변환 규칙을 제공한다. 그러나, 모든 콘텐츠를 공유하는 것은 매우 위험하다. 왜냐하면, 특정 클라이언트에 개인화되어 있거나 데이터베이스 등에 WRITE 행위가 있는 동적문서 (이하 Isolated 동적 문서라 함)의 타겟 콘텐츠는 요청을 보낸 클라이언트에만 유효한 문서이므로 공유기능은 효용가치가 없을 뿐만 아니라, 만약 Isolated 동적 문서를 공유할 경우 마땅히 실행되어야 할 DB Write 행위가 이루어지지 않을 수 있다. 따라서 모바일 응용 서버에서는 Isolated 동적 문서에 대한 공유기능을 지원하지 않으며, 동적문서에 대해서는 개인화되어 있지 않으며 READ-ONLY 인 동적문서 (이하 Shared 동적 문서라 함)에 대해서만 공유 기능을 제공한다. Shared 동적 문서의 대표적인 예로는 신문 사이트의 메인 페이지가 해당한다. READ-ONLY 인 Shared 동적 문서의 경우에도 데이터베이스의 내용이 변경되면 캐쉬하고 있던 콘텐츠를 캐쉬로부터 제거할 수 있는 기능이 제공되어야 한다. 공유기능에서는 파일 및 이미지와 같은 정적 문서에 대해서 타겟 콘텐츠를 캐쉬하고 있으며, 파일의 변경에 따라 캐쉬의 생명주기

를 관리한다.

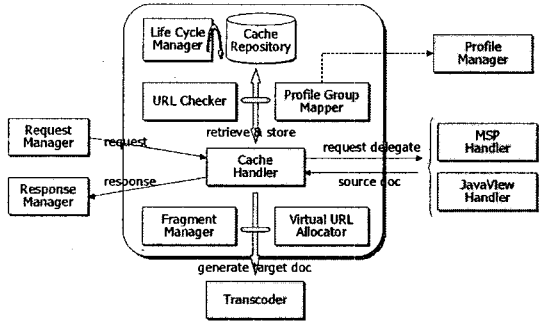
모바일 응용 서버의 캐쉬에서 또 다른 기능인 단편화 기능이란, 캐쉬 관리자는 각 모바일 단말기에 대한 메모리 용량 및 화면 사이즈 등의 제약 정보를 프로파일 그룹으로부터 얻어 와서 콘텐츠 열람 편의성과 장치의 콘텐츠 수용 능력을 고려하여 요청된 원본 문서에 대해 지능적인 문서 단편화 기능을 수행하며, 클라이언트의 요청을 여러 개의 Sub 요청으로 나누고 각 Sub 요청 마다 해당 조각(fragment)을 보내줄 수 있어야 하는 기능이다. 그리고, Isolated 동적 문서에 대해서 공유 기능을 지원하지 않는다고 하여도 문서 단편화 기능을 제공하기 위해서는 캐쉬에 Isolated 동적 문서에 대한 fragment 를 저장하고 있다. 이러한 단편화 기능을 통해서 클라이언트 단말기의 메모리 폭주를 막을 수 있을 뿐만 아니라, 데이터 전송 속도가 늦은 네트워크에서 최적의 속도로 데이터를 전송할 수 있다

- fragment 요청 처리: fragment 를 요구하는 클라이언트 request 에 대해서 해당 fragment 를 제공할 수 있어야 함.
- fragment 영속성 관리 : fragment 들을 저장장치 (DBMS, 파일시스템)등에 저장하고 추출할 수 있어야 함.
- fragment 생명주기 관리 : 캐쉬에 저장되어 있는 fragment 들에 대해서 일정 기간동안 사용되지 않으면 폐기할 수 있어야 함.
- fragment URL 부여 : fragment 들에 가상 URL 을 부여할 수 있어야 함.
- fragment URL 연결 기능 : fragment 간의 가상 URL 연결기능을 제공하여야 함

문서 단편화 기능은 타겟 문서를 여러 개의 fragment 로 나누는 것이다. 그런데, 캐쉬에 저장되는 단위에 따라 타겟 문서 전체를 대상으로 하는 경우와 fragment 들을 저장하는 경우의 2 가지 방법이 있을 수 있으며 각각에 대한 장단점이 있다. 즉, 타겟 문서를 저장하는 경우에는 fragment URL 부여 및 연결 기능은 각각의 sub 요청에 대한 처리 runtime 시에 이루어지며, 타겟 문서 하나에 대해서만 관리를 하므로 타겟 문서에 대한 영속성 및 생명주기 관리가 용이하며, 또한 적은 용량의 캐쉬가 적용 가능하다. 그러나, 매번의 sub 요청마다 변환을 하여야 하므로 성능이 떨어지는 단점이 있다. 반대로 캐쉬에 저장하는 단위를 fragment 로 하는 경우에는 변환된 문서를 작은 조각으로 나누어 각 조각마다 가상 URL 을 부여하여 캐쉬에 저장하는데, fragment URL 부여 및 URL 간 연결 기능은 첫번째 요청 때에만 이루어지며, 그 후에는 변환기를 통한 fragment 작업을 할 필요가 없어서 빠른 응답 속도를 제공하지만, 많은 fragment 들을 관리하므로 영속성 및 생명주기 관리가 다소 어려우며, 또한 같은 타겟 문서라도 디바이스 프로파일 그룹 마다 단편화 작업을 하여야 하며 각 그룹 마다 수많은 fragment 들을 캐쉬가 저장하여야 하므로 캐쉬의 용량

이 커야 하는 단점이 있다.

이와 같이, 각각의 방법에 대해서는 캐쉬의 용량과 속도간의 trade-off 문제가 있지만, 캐쉬를 도입하는 대표적인 이유가 서버의 성능을 향상시키는데 큰 목표가 있으므로 본 개발에서는 fragment 를 저장하는 방법을 선택하였다.



<그림 4> 캐쉬 관리자의 구조

캐쉬 관리자의 개략적인 구조는 <그림 4>와 같은 모듈들로 구성되며, 각 모듈들의 기능은 다음과 같다.

- Cache Handler : 클라이언트의 요청을 가장 먼저 받아 캐쉬에서 해당 문서의 존재 여부를 판단하여 캐쉬에 없는 경우에만 페이지생성 컴포넌트를 실행시키는 핸들러에게 클라이언트 요청을 전달
- URL Checker : 클라이언트의 요청 URL 에 대해 해당 문서의 존재 여부를 결정
- Profile Group Mapper : 유사한 형태의 프로파일을 갖은 단말기들을 그룹화하여 관리
- Life Cycle Manager : 캐쉬 데이터의 생명주기를 관리
- Fragment Manager : 타겟 문서를 단말기 용량 등의 이유로 여러 개의 작은 조각으로 분할
- Virtual URL Allocator : 클라이언트의 요청을 여러 개의 Sub 요청으로 나누고 각 Sub 요청 마다 해당 Fragment 를 보내줄 수 있도록 각 Fragment 마다 가상의 URL 을 부여

생명 주기를 담당하는 Life Cycle Manager 에는 기능에 따라 , Auto Life Cycle Manager 와 Manual Life Cycle Manager 2 가지가 있으며, 각각의 기능은 다음과 같다.

- Auto Life Cycle Manager : 캐쉬 데이터의 생명 주기를 관리하며, 캐쉬가 차면 자동으로 LRU(Least Recently Used) 교체기법을 사용해서 가장 오랫동안 사용되지 않은 문서를 교체한다.
- Manual Life Cycle Manager : 캐쉬에 저장된 콘텐츠에 대한 수동적인 생명 주기를 담당한다. 동적문서의 경우 데이터의 수정이 있는 경우 캐쉬된 문서가 제거되어야 하며, 이는 응용개발자에 제공되는 API 로 제어되도록 한다.

위와 같은 모듈들로 구성된 캐쉬 관리자의 동작을 보면, 모바일 응용 서버는 클라이언트로부터 요청이 들어 왔을 때 가장 먼저 디바이스 Profile Manager 로부터 해당 디바이스의 프로파일을 얻어와서, Profile Group Mapper 을 통해 디바이스가 어느 그룹에 속하는지를 결정하고, URL Checker 를 통해 해당 그룹의 임의의 단말기가 이전에 요청한 콘텐츠가 캐쉬에 저장되어 있는지를 URL Checker 를 통해 검사하여 만약 해당 콘텐츠가 존재하면 Response 객체를 통해 검색된 콘텐츠를 전달한다. 만약 캐쉬에 해당 콘텐츠가 없다면 페이지 생성을 위한 컴포넌트 핸들러에게 클라이언트의 요청 메시지를 전달하여 해당하는 페이지생성 컴포넌트가 XHTML 형식의 페이지를 생성하도록 한다. XHTML 의 페이지는 3 장에서 설명한 Transcoder 를 통해 프로파일 그룹에서 정의된 변환 규칙을 통해 타겟 콘텐츠로 변환된다. 변환이 완료된 타겟 콘텐츠는 문서 단편화 과정이 필요한 경우 Fragment Manager 를 통해 작은 조각으로 쪼개어지며 Virtual URL Allocation 에 의해 각각의 조각에 대해서 가상 URL 이 부여되며 조각들 간의 링크를 유지하도록 한다. 변환이 완료된 콘텐츠는 캐쉬에 저장되는데, 캐쉬에 콘텐츠를 저장하기 위해서는 파일이름, 수정시간, 정적 문서인지 동적 문서인지 구분하는 flag 등의 데이터가 필요하다. 또한, Isolated 인지 Shared 동적 문서인지 구분하기 위한 flag 를 두어 Isolated 문서인 경우 session 정보와 같이 fragment 들을 저장하여, 다른 클라이언트에게는 해당 세션과 연결된 fragment 를 전달하지 않도록 한다.

5. 결론

본 논문에서는 현재 개발 중인 모바일 응용 서버에 대해 간략히 살펴 보았으며, 그 중에서도 불필요한 콘텐츠 변환 과정을 생략하여 전체 페이지의 생성 시간을 줄여 응답 속도를 개선할 수 있는 캐쉬 관리자에 대해서 살펴보았다. 본 논문에서 제안하고 있는 캐쉬 관리자는 계층적인 구조로 프로파일 그룹을 관리하여, 각 프로파일 그룹마다 변환 규칙을 작성하고, 해당 그룹에 속한 단말기에서 요청된 콘텐츠는 해당 그룹의 변환 규칙이 적용되어 타겟 콘텐츠를 생성하도록 하여 캐쉬의 적중율을 높여 서버의 성능을 향상시킬 수 있도록 하고 있다.

참고문헌

[1] 김성훈, 장철수, 정승욱, 서범수, 노명찬, 박종기, 이경호, 김중배, 유무선 통합 모바일 응용 서버에 관한 연구, 정보과학회학술지, 제 20 권 1 호, 2002.
 [2] <http://www.metagroup.com/cgi-bin/inetcgi/jsp/home.do>
 [3] Geoff Johnson, "m-Commerce Scenario," Gartner Group, 2001.
 [4] <http://www.w3.org/Mobile/CCPP>, "Composite Capabilities / Preferences Profile Working Group Public Home Page"