대용량 웹에서 RCS를 이용한 웹 히스토리 저장 시스템 설계

이무훈° 이민희 조성훈 장창복 김동혁 최의인
한남대학교 컴퓨터공학과
{mhlee°, mhl, shcho, chbjang, dhkim, eichoi}@dblab.hannam.ac.kr

A Design of Web History Archive System Using RCS in Large Scale Web

Moohun Lee° Minhee Lee, Sunghoon Cho, Changbok Jang, Donghyuk Kim, Euiin Choi
Dept. of Computer Engineering, HanNam University

요 약

웹의 급속한 성장에 따라 웹 정보는 시간적 · 공간적 제약을 받지 않고 널리 활용되어지고 있다. 하지만 기존에 유용하게 사용되던 정보가 어느 순간 삭제가 된다면 더 이상 웹 정보를 이용할 수 없게 된다는 문제점이 존재한다. 이러한 문제를 해결하기 위해 웹 아카이브 시스템에 대한 연구와 좀더 효율적으로 삭제된 웹 정보를 저장하기 위한 기법들이 제안되었다. 그러나 기존의 기법들은 단순히 웹 정보를 저장하는 것에만 초점을 두었기 때문에 저장 공간의 효율성 및 제약성을 전혀 고려하지 않는 단점을 가지고 있다. 따라서 본 논문에서는 WebBase를 기반으로 하여 레포지토리에서 갱신되는 웹 정보들을 효율적으로 저장하고 검색할 수 있는 웹 히스토리 저장 시스템을 설계하였다. 본 논문에서 제안한 기법은 웹 히스토리 저장 시스템 설계를 위해 별도의 Crawler를 두지 않고 WebBase를 활용함으로써 웹 정보 수집에 대한 오버헤드를 줄일 수 있고, 삭제되는 웹 정보를 RCS를 통하여 체계적이고 효율적으로 저장함으로써 중요한 웹 정보를 공유할 수 있도록 하였다.

1. 서 론

웹은 인터넷이라는 하부구조를 기반으로 급속한 성장을 이루어왔으며, 규모나 사용자의 의존도 측면에서 실생활에서 없어서는 안 될 중요한 정보원으로 자리 잡았다. 따라서 웹 정보에 대한 효율적인 관리의 필요성이 대두되었고, 이를 위해 다양한 연구 활동이 시작되었다. 특히, 웹 정보의 획득에 있어서 효율적이고 정확한 웹 정보를 제공하기 위한 많은 노력을 기울여 왔으나, 중요한 웹 정보의 보존에 대한 연구는 미비한 상태이다. 뿐만 아니라 기존에 중요한 정보원으로 활용되었던 웹 페이지들이 그 중요도와 상관없이 소멸되어져가고 있는 실정이다[1, 2]. 따라서 이렇게 소멸되어져가는 중요한 정보들을 수집 · 보존하기 위해서는 효율적인 저장 기법에 대한 연구와 이를 토대로 하는 웹 아카이브 시스템에 대한 연구가 필요하다[2, 3].

기존에 연구되었던 웹 아카이브 시스템(web archive system)은 저장 공간의 효율성 및 제약성을 전혀 고려하지 않고 단순히 소멸되는 웹 정보를 저장하는 것에만 초점을 두고 있다. 또한 웹 정보를 수집함에 있어 여러 개의 Crawler를 두기 때문에 대역폭의 낭비나 중복된 페이지의 수집과 같은 문제점이 발생한다.

본 논문에서는 이러한 기존 웹 아카이브 시스템의 문제점을 해결하기 위해 Stanford WebBase 기반으로 웹 페이지를 수집함으로써 Crawling을 위한 오버헤드를 줄일 수 있고, RCS (Revison Control System)를 활용하여 저장 공간 및 검색의 효율성을 극대화할 수 있는 웹 히스토리 저장 시스템을 제안하였다.

본 논문의 구성은 다음과 같다. 2장에서 관련연구에 대해 이야기하고, 3장에서는 본 논문에서 제안한 웹 히스토리 저장 시스템의 구조와 각 모듈의 처리 절차를 기술하며, 4장에서는 기존의 웹 아카이브 시스템과 비교분석한 후, 마지막으로 5장에서 결론을 맺는다.

2. 관련 연구

본 논문에서 제안하고 있는 웹 히스토리 저장 시스템은 Stanford WebBase Project에 기반을 두고 있다. WebBase는 Stanford 대학에서 개발 중인 웹 검색 엔진 프로젝트로써 효율적인 웹 페이지의 수집, 관리, 인덱스 구성, 검색에 대한 연구가 진행 중이다. WebBase의 초기 버전은 Google 검색 엔진의 Crawler, Repository 및 전반적인 부분에서 사용되었다. WebBase는 Crawler, Storage Manager, Metadata & Indexing, Multicast, Query Engine의 5 가지 모듈로 구성되어져 있다[4, 5]. Crawler 모듈은 웹으로부터 페이지들을 수집하여 Storage Manager 모듈 전송한다. 전송된 페이지들은 WebBase의 레포지토리(repository)에 저장된다. Metadata & Indexing 모듈은 저장된 페이지와 메타데이터에 대한 인덱싱을 수행하고, Query Engine과 Multicast 모듈은 레포지토리에 저장된 contents의 접근을 제공한다.

RCS는 텍스트, 일반 문서, 소스코드, 테스트 데이터 등 각종 파일의 버전을 관리하는 시스템이다. 즉, RCS는 다양한 파일의 리비전(revision)을 저장하고, 저장된 임의의 버전을 선택적으로 판독하는 기능을 제공한다. 그리고 전체 파일의 버전을 버전 트리(version tree)의 형태로 구성하고 각각 파일의 버전을 하나의 노드로써 구성함으로써 검색의 효율성을 향상시킬 수 있다. 또한 각 버전들 사이의 변화값(delta)을 저장함으로써 저장 공간의 낭비를 막을 수 있고, 가장 최신의 파일은 원본 그대로 유지하여 최근 파일의 접근시간을 최소화 할 수 있다[6, 7].

WayBack Machine은 비영리 집단인 Internet Archive와 Alexa Internet이 공동으로 개발한 시스템으로써 웹 정보를 디지털 도서관의 형태로 보존하려는 최초의 시도였다. 이 시스템의 연구가 시작된 1996년 이후에 지금까지 300억만 개 이상의 웹 페이지들이 수집되어 있으며, 각 웹 페이지들은 다양한 버

전으로 저장 · 관리되어 웹을 통해 공유하고 있다. 하지만, WayBack Machine의 경우 다양한 웹 정보를 수집하기 위해 별도의 Crawler를 구성하고 있으며 웹 정보를 보존하는 저장소의 효율성은 전혀 고려하지 않고 있다. 이 시스템은 실제적으로 2001년 U.C.Berkeley의 Bancroft Library에서 이용하였다 [8].

3. 웹 히스토리 저장 시스템

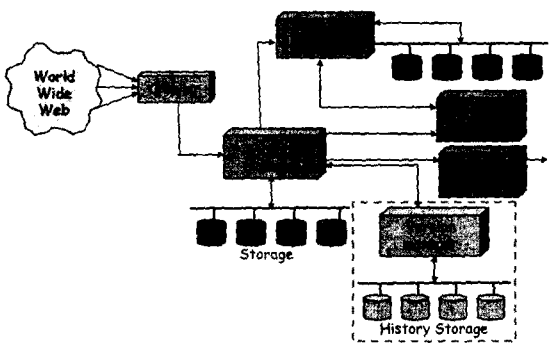
본 논문은 Stanford WebBase Project에 기반을 두고 시스템의 레포지토리에서 삭제되어지는 웹 페이지들을 수집하여 버전을 통해 저장할 수 있는 Version Manager를 설계 하였다.

3.1 웹 히스토리 저장 시스템 구조

현재 웹 상에 존재하는 수많은 정보에 대한 획득과 관리는 비교적 효율적으로 이루어지고 있지만, 이러한 정보들이 최신의 것으로 갱신될 경우, 그 이전의 정보에 대해서는 관리가 미비하다는 문제점을 가지고 있다. 즉, 웹 상의 모든 정보는 정보가 저장되어 있는 서버의 관리자들에 의해 지속적으로 갱신 및 삭제되기 때문에 그것의 중요성 여부와 관계없이 대다수의 정보가 소멸되어 간다는 단점을 가지고 있다.

따라서 본 논문에서는 웹 검색 엔진인 WebBase를 통해 갱신되어 삭제되기 이전의 모든 정보들을 수집하고 이를 히스토리 저장소 내에 체계적으로 저장함으로써 삭제되어지는 중요한 웹 정보 효율적으로 공유할 수 있게 하였다.

본 논문에서 제안한 시스템은 WebBase의 Crawler를 통해 페이지들을 수집하고 Storage Manager에 의해 페이지를 레포지토리에서 갱신한다. 이 때, 저장소 내에 저장되어 있던 기존 페이지가 삭제되면, 삭제된 페이지(history page)를 Version Manager를 통해 History Stroage에 저장 및 관리할 수 있도록 하였다. Version Manager에서는 여러 개의 처리모듈을 두어 버전을 관리함으로써 수집된 페이지의 체계적인 관리가 가능하도록 하였다. 또한 Version Manager를 통하여 처리된 히스토리 페이지는 History Storage에 저장하게 된다. [그림 1]은 본 논문에서 제안하는 웹 히스토리 저장 시스템의 구조를 보여주고 있다.



[그림 1] 웹 히스토리 저장 시스템의 기본 구조

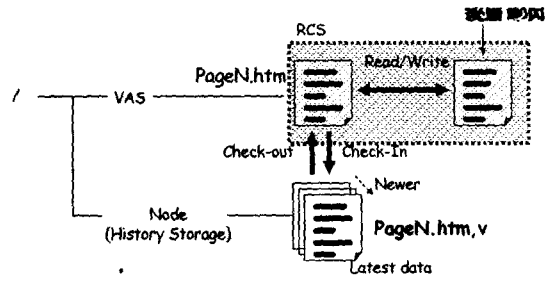
본 논문에서 제안한 시스템은 기본적으로 WebBase의 각 모듈들을 활용하고 있으며, 히스토리 페이지를 별도로 저장하기 위한 Version Manager와 History Storage를 추가하여 설계하였다. Version Manager는 VCS(Version Control System)와 각 노드와 연결되는 VAS(Version Assignment System)들로 구성하였고, History Storage는 여러 개의 노드로 분산된 형태로 구성하였다.

3.2 웹 히스토리 저장 시스템의 처리 절차

가. RCS 처리 절차

수많은 웹 페이지를 저장해야 하는 웹 히스토리 저장 시스템에서 RCS는 히스토리 페이지를 효과적으로 압축하고 저장하기 위한 시스템으로써, 페이지에 대해 버전 작업을 처리하는 부분이다. History Storage 내에 각 페이지별 리비전 그룹을 생성하고, 생성된 리비전 그룹을 Check-in/out 연산을 통하여 History Storage 내부의 각 노드에 저장하는 역할을 수행한다. 또한 RCS는 버전 수행 과정에서 현재 페이지와 이전 페이지의 변화값만을 저장함으로써 대용량의 웹 페이지를 저장하는데 있어서 저장 공간을 효율적으로 운용할 수 있도록 해준다.

[그림 2]는 RCS를 이용하여 히스토리 페이지들을 처리하는 과정을 나타낸 것이다. 먼저 VCS를 거쳐 VAS로 삽입된 데이터는 NIT(Node Information Table) 내에 페이지가 존재하는 지를 검색하게 된다. 존재할 경우 PageN.htm,v를 Check-out 한 후 갱신한 다음, 리비전 그룹으로 Check-in하는 과정을 거쳐 처리된다. 이때 History Storage 내부에서 PageN.htm은 PageN.htm,v라는 리비전 그룹의 형태로 저장되어지며, 이때 이전 페이지와 현재 저장되어지는 페이지의 변화값을 계산하여 저장한다.

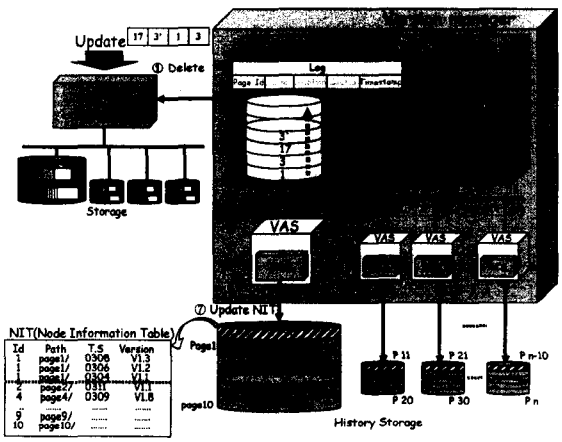


[그림 2] RCS의 처리 과정

나. 히스토리 페이지 처리 절차

본 논문에서 제안하고 있는 웹 히스토리 저장 시스템의 히스토리 페이지 처리 절차는 다음 [그림 3]과 같다. 우선 Crawler에 의해 페이지를 수집하고, Storage Manager에 의해 페이지의 갱신이 일어나게 되면 Storage Manager가 Storage 내부의 페이지들에 대한 압축과정을 수행하게 된다. 압축과정이 진행되면서 삭제되어지는 페이지는 Storage Manager를 통해 Version Manager의 VCS로 전송된다. VCS는 가장 오래된 데이터의 우선적 처리를 위하여 타임스탬프를 기준으로 데이터의

정렬을 수행한다. 이때, 페이지의 Catalog에 있는 Page ID를 기준으로 각각의 노드에 연결된 VAS로 페이지를 할당한다. 할당된 페이지는 노드의 NIT와 비교하여 테이블 내에 동일 Page ID 존재 여부를 판별한 후 처리한다. 만일 NIT에 Page ID가 존재할 경우 테이블의 Path 정보를 이용하여 노드 내 리비전 그룹을 Check-out 한 후, Check-out 한 페이지에 대하여 갱신을 수행한다. 갱신된 페이지는 리비전 그룹으로 Check-in을 수행한다. 만일 NIT에 Page ID가 존재하지 않을 경우 페이지에 대한 새로운 리비전 그룹을 생성한다. 마지막으로 노드 내 존재하는 NIT의 Page ID, 타임스탬프, Version에 대한 정보의 갱신을 수행함으로써 히스토리 페이지에 대한 버젼을 완료한다.



[그림 3] 웹 히스토리 저장 시스템의 처리 절차

히스토리 페이지의 처리절차를 간략히 요약하면 다음과 같다.

- 단계 ① : Storage Manager에서 삭제된 페이지를 VCS로 전송
- 단계 ② : 삽입된 페이지에 대한 Catalog의 정보추출
- 단계 ③ : 타임스탬프를 기준으로 하여 페이지를 정렬
- 단계 ④ : 각 페이지에 해당하는 노드의 VAS로 전송
- 단계 ⑤ : 페이지와 각 노드의 NIT를 비교
- 단계 ⑥ : RCS의 Check-in/out 수행
- 단계 ⑦ : NIT 갱신

4. 비교분석

본 논문에서 제안한 시스템은 기존의 웹 아카이브 시스템에 비해 웹 히스토리 페이지를 효율적으로 저장, 관리하기 위한 체계적인 처리기법을 제시하고 있으며, 페이지의 변화값만 저장함으로써 저장공간의 낭비를 최소화하였다. 또한 페이지에 대한 버젼을 통하여 수집된 데이터에 대한 연관관계를 저장함으로써 효율적인 관리가 가능하다.

[표 1]은 기존의 WayBack Machine과 제안 시스템을 비교, 분석한 것이다.

[표 1] 비교분석

	[illegible]	제안 시스템
연관성	버전간 연관성 없음	버전간 연관성 생성
저장공간	저장공간 낭비	저장공간 최소화
추가비용	Crawler 필요	Crawler 불필요
확장성	검색엔진과 독립적	검색엔진과 종속적

5. 결 론

본 논문은 Version Manager를 제안함으로써 Storage Manager에서 삭제된 히스토리 페이지를 체계적이고 효율적으로 저장 할 수 있는 웹 히스토리 저장 시스템을 설계하였다. 제안된 웹 히스토리 저장 시스템은 기존의 웹 아카이브 시스템보다 히스토리 페이지를 체계적이고 효율적으로 저장, 관리 할 수 있고, 저장된 히스토리 페이지들 간에 버전 관리를 수행함으로써 페이지들 간의 연관관계를 생성할 수 있다. 또한 RCS를 적용하여 이전 버전 페이지와의 변화값만을 저장함으로써 저장공간의 낭비를 줄이고, 보다 효율적으로 History Storage를 운영할 수 있도록 하였다.

향후 연구 과제로는 저장된 히스토리 페이지를 활용할 수 있는 공유 방안에 대한 연구와 그에 따른 히스토리 페이지의 효율적인 검색, 인덱싱 기법에 관한 연구가 수행되어야 한다.

참고 문헌

[1] Arvind Arasu, Junghoo Cho, Hector Garcia- Molina, Andreas Paepcke, and Sriram Raghavan, Searching the web (invited paper). ACM Transactions on Internet Technology, 1(1), August 2001.

[2] Joao P. Campos, Versus: a Web Data Repository with Time Support, May 2003, http://www.di.fc.ul.pt /tech-reports.

[3] Burner, M. Crawling towards eternity: Building an archive of the World Wide Web. Web Techniques Magazine 2, 5 (May 1997).

[4] WebBase, http://www-diglib.stanford.edu/~testbed /doc2/ WebBase.

[5] Jun Hirai, Sriram Raghavan, Hector Garcia-Molina, and Andreas Paepcke. Webbase: A repository of web pages. In Proceedings of the International World-Wide Web Conference, pages 277-293, May 2000.

[6] Walter F.Tichy, RCS—A System for Version Control, Department of Computer Sciences Purdue University, 1991.

[7] Walter F.Tichy, Design, Implementation, and Evaluation of a Revision Control System, IEEE, 1982.

[8] Internet Archive, http://www.archive.org.