웹 필터링 시스템을 위한 URL 탐색 기법

김창섭[U] 정경진 김성조
중앙대학교 컴퓨터공학과
{cskim, neutrino, sjkim}@konan.cse.cau.ac.kr

A URL Search Technique for Web Filtering System

Chang-Sup Kim Kyung-Jin Jung Sung-Jo Kim
Dept. of Computer Science & Engineering, Chung-Ang University

요 약

현대 생활에서 필수가 된 웹은 많은 정보를 쉽게 얻을 수 있다는 장점이 있지만 음란물의 유혹과 업무시간의 주식 투자 등의 부작용들도 속속 나타나고 있다. 이러한 문제를 해결하기 위한 방안으로서 웹 필터링 시스템이 활용되고 있다. 본 논문에서는 시스템의 성능 저하와 사용자의 웹 접속 지연 시간을 최소화하면서 차단 여부를 신속히 판별할 수 있는 URL 탐색기법을 제안한다.

1. 서론

전세계의 네트워크들이 서로 연결되어 있는 인터넷은 이미 초기의 군사적인 목적과 학술적인 연구 차원을 벗어나 일상 생활에 깊숙이 파고들었다. 이제 인터넷은 현대인에게 필수가 되었으며 이를 통해 많은 정보를 서로 공유하며, 의견을 나눌 수 있게 되었다. 이러한 잇점을 바탕으로 기업에서는 업무에 필요한 정보의 바른 획득과 공유를 위해 회사의 네트웍을 인터넷에 연결해서 사용하고 있다. IDC의 연구에 따르면 2000년 현재 전세계 고용자 대략 1억 2000만 명이 인터넷에 연결되어 있고, 2003년까지 대략 2억 7200만 명이 인터넷을 사용할 것으로 예상되고 있다[1]. 하지만, 기업 내에서의 인터넷의 이용은 새로운 부작용을 야기하고 있다. 너무나 쉽게 접근할 수 있는 음란물의 유혹과 주식투자 등의 업무에 지장을 주는 환경으로 인한 업무로의 집중력 결여와 업무 시간에 사적인 목적으로 인터넷을 사용하여 실제 업무에 필요한 네트웍의 대역폭보다 훨씬 많은 부하로 인한 만성적인 대역폭 부족문제와 보안 위협과 같은 문제점 또한 발생하게 되었다.

이러한 문제점을 극복하기 위해 많은 웹 필터링 프로그램들이 등장하게 되었다. 이러한 프로그램들은 대부분 차단하고자 하는 웹 사이트의 URL들을 미리 저장해 두고, 요청한 사이트가 미리 저장되어있는 차단 목록 내의 사이트와 일치하면 해당 요청을 차단한다. 모든 인터넷 접근에 대하여 차단 목록과 요청 URL을 비교하는 것은 시스템의 성능 저하 및 웹 접속 지연을 유발하여 사용자의 불만을 가중시키고 있다. 따라서 필터링 소프트웨어에서는 접속 지연시간을 최소화하기 위한 빠른 탐색 방법이 강구되어야 한다. 그리고 대부분의 상용 프로그램들은 유지하고 있는 차단 목록이 공개되는 것을 방지하기 위해 데이터의 인코딩을 주로 사용하고 있다. 본 논문에서는 인터넷 필터링을 위해 인코딩을 지원하면서, 효과적으로 탐색을 할 수 있는 기법을 제안하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 인터넷 접근의 차단을 위한 공개 필터링 프로그램에 사용된 탐색 기법을 살펴본다. 3장에서는 2장에서 제기된 문제점을 해결하기 위한 IP 인덱싱과 사전식 경로명 탐색 트리를 이용한 URL 탐색기법에 대해 기술한다. 마지막 4장에서는 결론과 향후 연구 과제에 대해 논의한다.

2. 관련 연구

웹에 대한 접근을 차단하기 위한 프로그램들이 그동안 많이 등장하였지만 대부분의 상용 프로그램들은 그 탐색 및 차단 알고리즘이 공개되어 있지 않기 때문에 본 논문에서는 공개 프로그램들을 대상으로 비교하겠다.

2.1. SquidGuard

SquidGuard는 공개 라이브러리인 Berkely DB를 사용하여 URL의 비교를 수행한다. Berkely DB는 데이터베이스로의 접근 방법으로 B+트리와 확장형 선형 해시나 고정/가변 길이 레코드등의 방법을 제공한다[2]. 하지만 SquidGruad는 Berkely DB에서 제공하는 접근 방법 대신에 자체적으로 B-트리를 이용하며 탐색을 수행한다[3]. 또한 호스트명과 경로명에 따라 차단할 수 있으며

정규식(Regular expression)을 통하여 파일 종류에 따른 차단이 가능하다. 하지만 SquidGuard는 단순히 URL들로만 차단이 가능하며 IP 주소로는 차단이 불가능하다. 또한 IP 주소를 통한 차단 자체를 지원하지 않으므로 동일 IP에 여러 개의 호스트명이 존재하는 경우에는 모든 호스트명을 차단목록으로 관리해야 한다. 또한 전체 URL을 B-트리로 구현함에 따라 메모리 요구량이 너무 많다는 문제가 있다.

2.2. squirm

squirm은 선형탐색을 통해 차단 여부를 결정한다. 하지만 순수한 선형탐색은 많은 탐색 시간이 소요되므로 URL 전체 중에서 우선 비슷한 파일 이름을 가진 일부를 먼저 비교하는 accelerator 스트링 기법을 사용한다 [4]. 하지만 squirm 역시 IP 주소만의 탐색을 지원하지 않는 문제점을 가지고 있다.

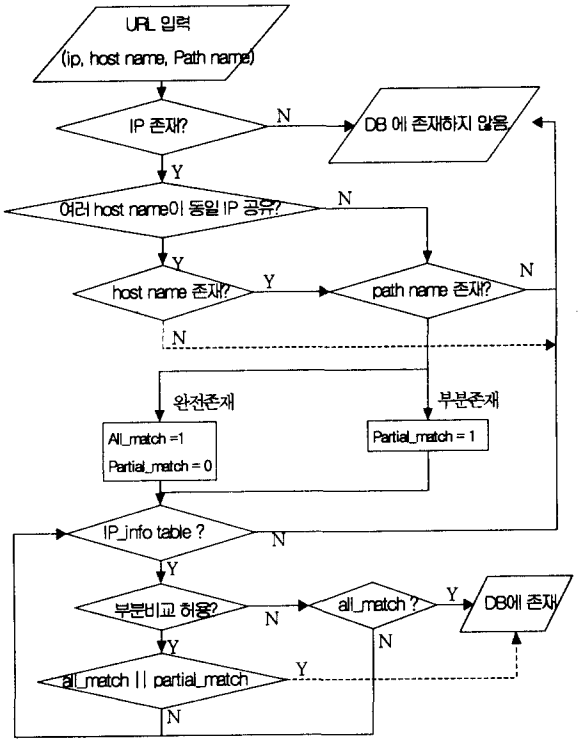
또한 위에서 살펴본 두 프로그램은 모두 차단 목록이 평이한 텍스트로 작성되어 노출이 되기 쉽다.

3. URL 탐색 기법

본 논문에서 제안하는 탐색 기법의 구현과 빠른 탐색을 위해서는 다음과 같은 필터링 시스템의 조건이 필요하다.

- ▷ URL 탐색을 위해 모든 차단 목록은 메인 메모리 상에 상주되어야 한다.
- ▷ 요청 URL은 캐싱 서버나 프록시 서버로부터 입력받는다.
- ▷ 입력 URL은 IP, 호스트명, 경로명, 포트번호 그리고 프로트콜 등으로 분리되어 필터링 시스템으로 전달된다

필터링 시스템에서 필터링이 수행되는 과정은 (그림 1)과 같다. URL 탐색을 위해서는 두 가지 탐색이 사용된다. 첫 번째는 IP 인덱싱으로, 입력받은 URL의 IP 주소가 차단목록에 존재하는 지를 확인하게 된다. 두 번째는 사전식 경로명 탐색 트리 방법으로 경로명을 디렉토리 단위로 분리하여 저장하여 탐색하는 방법이다. 이 과정에서 만일 이미 저장하고있는 URL 데이터베이스 내의 IP가 여러 개의 호스트명을 가지고 있으면 호스트명의 탐색 후 경로명의 탐색을 실시한다. 그리고 경로명이 완전히 일치하지 않는 경우에도 부분 비교 비트를 사용하여 디렉토리 단위의 차단이 가능하다. 또한 하나의 IP에 여러 개의 경로명이 존재 할 수 있으므로 해당 IP에 더 이상의 IP 정보데이터가 존재하지 않을 때까지 비교를 수행한다.



(그림 1) 필터링 흐름도

3.1. IP 인덱싱

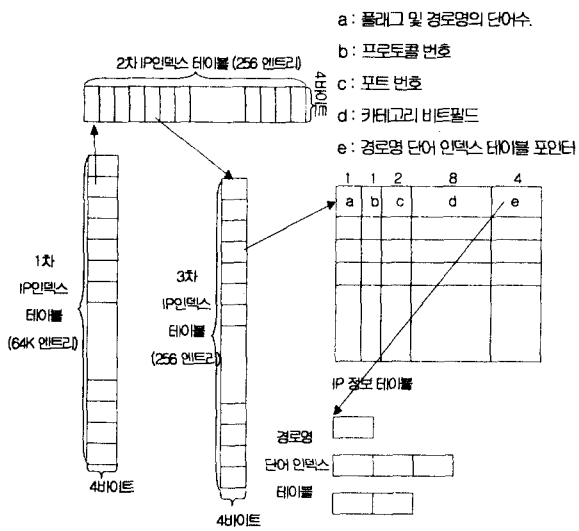
본 시스템에서는 2장에서 이미 언급한 바와 같이 빠른 탐색을 위해 기존의 B-트리를 이용한 방법이나 순차적인 방법을 이용하는 대신 IP를 이용한 인덱싱을 사용한다. (그림 2)에서 보는 바와 같이 우선 IP를 비교하기 위해 다음과 같은 3번에 걸친 인덱싱을 수행한다.

▷ 1차 인덱스 테이블
- 64K(256×256)개의 엔트리로 구성
- IP 주소의 앞 두 자리 값을 인덱스로 사용
- 2차 인덱스 테이블의 포인터를 값으로 사용

▷ 2차 인덱스 테이블
- 256개의 엔트리로 구성
- IP 주소의 세 번째 자리 값을 인덱스로 사용
- 3차 인덱스 테이블의 포인터를 값으로 사용

▷ 3차 IP인덱스 테이블
- 256개의 엔트리로 구성
- IP 주소의 네 번째 자리 값을 인덱스로 사용
- IP정보 테이블(차단목록에 속한 URL에 관한 정보를 저장)의 포인터를 값으로 사용

이렇든 3개의 인덱스의 테이블을 사용하므로 항상 3번의 비교만으로 요청한 URL의 IP가 차단목록에 존재하는지 여부를 확인할 수 있으므로 정해진 시간 내에 빠른 탐색을 할 수 있다. 비교 횟수를 줄이기 위해 2개의 인덱스 테이블을 사용하지 않은 이유는 메모리의 낭비를 막기 위함이다. 2번의 인덱싱으로 IP 탐색을 마치려면 두 번째 인덱스 테이블은 64K의 개수를 가진 테이블로 구성되어야 한다. 이는 과다한 메모리의 사용의 여지가 있으므로 본 시스템에서는 메모리의 효율적인 관리를 위해 3번의 인덱싱을 수행한다.



(그림 2) IP 인덱싱

3.2. 사전식 경로명 탐색 트리

IP를 통한 탐색을 마친 후 동일 IP에 여러 개의 호스트명을 가진 URL은 차단 목록에 각 호스트명을 dot(.)로 구별하여 여러 개의 단어를 생성한다. 경로명의 경우에는 /으로 구별하여 여러 개의 단어를 생성한다. 이러한 단어들로 B+-트리 구조를 이용한 사전을 생성한 후 각 단어의 인덱스를 차단목록에 저장한다. 이러한 구조를 가짐으로서 다음과 같은 장점을 얻을 수 있다. 중복적으로 데이터베이스에 나타나는 많은 단어들, 예를 들면 성인 사이트의 경우 'adult' 같은 단어들은 대부분 중복적으로 사용되는데 이를 사전식 경로명 탐색 트리를 이용하면 단 한번의 저장이 필요하므로 메모리의 사용량을 줄일 수 있다. 또한 대부분의 상용 프로그램은 차단 사이트 URL 데이터베이스의 내용을 회사의 자산으로 여겨 직접 노출시키지 않는다. 그런데 본 탐색 기법에서 제시한 경로명 탐색 트리를 이용하여 인덱스의 시작 위치에 임의의 값을 더하여 적용시킴으로써 차단 URL 데이터베이스의 내용을 읽기 위해 메모리 덤프 등의 방법을 사용하는 것을 막을 수 있다.

3.3. 성능

본 논문에서 제시한 탐색 방안을 이용하면 다음과 같은 장점이 있다. 3.2절에서 언급한 바와 같이 본 시스템은 중복되는 단어를 하나의 기억장소에 보관하므로 B-트리에 비해 메모리의 사용량이 적다. 그리고 모든 데이터를 메모리 상에 상주 시켜 비교를 위한 디스크 입출력이 불필요하다. 또한 호스트명 이외의 경로가 없는 경우는 단순한 IP 비교만으로 차단 여부를 가릴 수 있기 때문에 B-트리를 사용하는 경우 보다 빠른 탐색이 가능하다. 그리고 짧은 경로명을 가진 URL의 경우에도 전체 URL을 비교하는 B-트리를 이용하거나 선형탐색을 이용하는 다른 시스템 보다 빠른 탐색속도를 얻을 수 있다. 하지만 URL의 경로명이 긴 경우는 탐색 시간이 늘어나는 단점이 있다.

4. 결론 및 향후 연구과제

본 논문에서는 제시하는 탐색 기법은 2단계로 구성되어 있다. 첫 번째 단계에서는 IP 인덱싱을 이용해 탐색을 수행하고 두 번째 단계에서는 사전식 경로명 탐색 트리를 이용한다. 이러한 기법의 사용으로 인해 빠른 탐색이 가능해지고 차단 사이트의 URL 데이터베이스의 노출을 막을 수 있다.

향후 연구과제로는 사전식 경로명 탐색 트리에서의 URL의 경로명이 길어지는 경우에 나타나는 탐색시간의 증가를 최소화하기 위한 알고리즘이 연구되어야 하며, 또한 다른 웹 필터링 시스템과의 성능 비교가 필요하다.

5. 참고 문헌

[1] Christian A. Chrstiansen *"Employee Internet Management"* IDC, 2000.

[2] http://www.squidguard.org/intro

[3] http://www.sleepycat.com

[4] http://www.senet.com.au/squirm