메타정보를 이용한 웹에서의 효율적인 캐쉬 관리 기법의 설계

한지영*, 윤성대**
부경대학교 전산교육학과*
부경대학교 전자계산학과**

A Design of Efficient Cache Management Scheme Using Meta Information in the Web

Ji-Young Han*, Sung-Dae Youn**
Dept. of Computer Science Education, PuKyong National University*
Dept. of Computer Science, PuKyong National University**

요 약

웹 정보의 급격한 양적 팽창은 네트워크 병목 현상과 사용자의 지연시간 증가 및 웹 서버의 과부하 등의 문제를 야기하고 있다. 이를 완화시키기 위한 방법으로 웹 캐싱이 이용되는데, 전통적인 캐싱과는 달리 문서의 종류와 크기가 가변적이며 많은 사용자의 요구를 처리해야하는 특성이 있다. 따라서 본 논문에서는 동적인 웹 환경과 한정된 크기의 웹 캐쉬 공간의 사용 효율을 향상시켜 캐쉬 적중률을 증가시키기 위한 방법으로, 서비스되는 각 파일의 메타정보를 Main Server의 캐쉬에 리스트 형태로 유지하는 CRBM(Client Request Buffer Manager)을 제안한다.

1. 서론

최근 인터넷 사용의 급격한 증가로 인해 웹 정보가 폭발적인 양적 팽창을 함으로써 보다 빠른 처리를 위해 서버의 처리 속도가 빠르게 증가하였다. 그럼에도 불구하고 많은 사용자가 이용하는 웹 서버의 경우에는 자주 과부하가 걸려 서비스가 중단되곤 한다. 이러한 과부하로 인한 웹서비스의 중단은 서비스 제공자와 사용자에게 큰 손실을 가져온다. 뿐만 아니라 네트워크 병목·현상과 같은 문제가 발생하며, 사용자의 웹 서비스 지연시간이 증가된다. 웹 서버 클러스터 시스템에서는 이런 문제들을 완화시키기 위한 방법으로 캐싱을 사용할 수 있다[1].

웹 캐싱이란 웹 사용자에 의해 빈번히 요청되는 웹 객체를 캐쉬에 보관하여 빠른 서비스를 가능하게 하는 기법으로, 기존의 페이징 방법과는 달리 가변 길이의 문서를 다루며 많은 사용자들로부터 요청이 이루어진다는 특징이 있다. 웹 캐싱을 사용함으로써 디스크 접근 횟수를 줄여 서버의 부하를 줄이고, 많은 웹 문서의 요청들로 인해 발생하는 네트워크 트래픽을 완화시킬 수 있으며, 또한 캐싱된 문서는 사용자에게 빠른 응답시간을 제공할 수도 있다. 기존의 접근 방법에는 클라이언트 캐쉬, 서버 캐쉬, 프락시 캐쉬, 계층적 캐쉬, 분할 캐쉬, 협력 캐쉬[2] 등이 있다. 이들 접근 방법에서는 '적중률을 향상시키기 위해 웹 캐쉬의 한정된 공간을 어떻게 효율적으로 사용할 것인가'가 공통적인 문제로 다루어지며, 일반적으로 하나의 URL을 구성하는 파일 단위로 캐싱을 한다. 하지만 멀티미디어 파일에 있어서 파일 단위로 캐싱을 할 경우, 파일의 크기와 캐쉬의 크기를 고려해볼 때 캐쉬의 사용성에 있어 효율이 저하되는 단점이 있다[1,3].

본 논문에서는 웹 서버 클러스터 시스템에서 사용자의 요구를 받아들이는 Main Server의 캐쉬에, 각 파일의 메타 정보를 리스트 형태로 유지함으로써 캐쉬 공간의 사용 효율을 향상시키고, 캐쉬 적중률을 높이며, 메타 정보를 이용한 파일 블록 접근으로 사용자의 서비스 지연시간을 줄이고자 한다.

본 논문의 구성은 다음과 같다 2장에서는 웹 캐쉬와 관련된 기존 연구에 대해 살펴보고, 3장에서는 CRBM(Clint Request Buffer Manager)을 제안하고, 마지막 4장에서는 결론 및 앞으로의 연구 과제를 제

시한다.

2. 관련연구

2.1 분할캐싱

웹에서 인식되는 객체 크기의 다양성을 고려하기 위해 캐쉬 저장 공간을 관리하기 위한 방법인 분할 캐쉬가 있으며[1,9], 분할 캐쉬란 모든 문서를 저장하는 단일 캐쉬를 가지는 대신, 캐쉬 저장 공간을 여러 개의 분할 영역으로 나누어서 문서들의 크기나 타입에 따라 각각의 문서들을 저장하는 방법이다. 캐쉬를 분할하는 기본적인 이유는 각각의 분할 영역에서 참조 지역성을 보존함으로써 전체 캐쉬 공간의 적중률을 높이고자 하는데 있다. 그러나 기존 연구에서는 캐싱 대상을 하나의 URL을 구성하는 파일 블록만을 고려하였기 때문에 멀티미디어 파일에 있어서는 파일의 크기와 캐쉬의 크기를 고려할 때 캐쉬 사용공간에 있어 효율성이 저하되는 단점이 있다.

2.2 협력캐싱

협력 캐쉬는 기존 요구 처리 단계 중 클라이언트의 요구가 자신의 캐쉬에서 처리되지 않을 경우, 서버에 요청하기 전 그 블록을 캐쉬하고 있는 다른 클라이언트에게 그 블록에 대한 요청을 하여 자신의 요구를 처리하는 방법이다. 협력 캐싱은 요구한 데이터 블록의 위치에 따라 로컬 캐쉬 탐색, 원격 캐쉬 탐색, 디스크 탐색의 세 단계로 이루어지며, 원격 캐쉬는 다른 클라이언트의 캐쉬 블록 정보를 유지해야 한다.

이러한 협력 캐쉬에 대한 많은 기존 연구 중 Dahlin[5]은 캐쉬 블록들의 관리를 위하여 N-chance 알고리즘을 제안하였고, Feeley[6]는 GMS(Global Memory Service)상에서 효율적인 캐쉬 블록 알고리즘을 제안하였다. 그리고 Sarkar[2,4]는 기존 협력 캐쉬에서 정확한 클라이언트 캐싱 정보를 가지고 있던 것을 단순한 힌트에 의해 블록의 캐싱 정보를 유지함으로써 캐싱 정보를 유지하는데 필요한 부하를 줄이는 방법을 제안하였다. 힌트는 디스크에서 읽어 캐쉬에 저장된 블록의 위치를 나타내며, 각 노드는 중앙에서 관리자 없이 힌트를 확인하여 캐쉬에서 원하는 블록의 위치를 찾아낼 수 있다.

N-chance 알고리즘과 GMS 알고리즘은 관리자를 통해 블록의 정보를 구하여 해당 파일 블록을 얻는 방법을 사용함으로써, 다수의 클라이언트가 동시에 블록을 요구할 때 관리자의 병목현상을 야기할 수 있다. 힌트 기반의 협력 캐쉬 알고리즘은 각 클라이언트가 전체 블록에 대한 정보를 가지고 있어 관리자를 거치지 않고 파일 블록을 검색할 수 있도록 함으로써, N-chance 알고리즘과 GMS 알고리즘에서의 관리자 병목현상을 해결하였지만, 각 클라이언트가 가지는 캐쉬 정보의 부정확성으로 인한 불필요한 작업이 야기된다.

이러한 협력 캐싱에 관한 연구들은 데이터 블록들이 여러 호스트에 중복해서 존재하기 때문에 데이터 일관성 유지가 어렵다는 단점이 있다. 또한 각 알고리즘의 적용은 디스크 접근 속도 보다 원격지 호스트의 메모리를 접근하는 속도가 훨씬 더 빠르다는 가정 하에 가능한 것으로, 이와 같은 연구는 ATM, Fast Switched LAN, Fiber Channel과 같은 고속의 네트워크를 지원하는 환경에서만 사용이 가능하다는 한계점이 있다.

2.3 부하 분배 알고리즘

웹 서버 클러스터의 부하 분배기는 제공되는 서비스에 따라 적절한 스케줄링 알고리즘을 통해 부하를 분산시키는 역할을 하며, 여러 노드가 하나의 가상서버로 보이게 하는 역할을 한다. 부하 분배 알고리즘은 서버의 성능에 많은 영향을 미치며 노드 선택을 위해 간단하고 빨리 계산될 수 있어야 한다는 것이다.

최근 부하 분배에 대한 연구들이 많이 이루어졌으나 잦은 디스크 접근으로 인한 전체 성능 저하라는 문제점을 안고 있다. 이러한 문제를 해결하기 위해 제시된 것이 내용 기반(Content-Based) 부하 분배 알고리즘과 캐싱 정책이다.

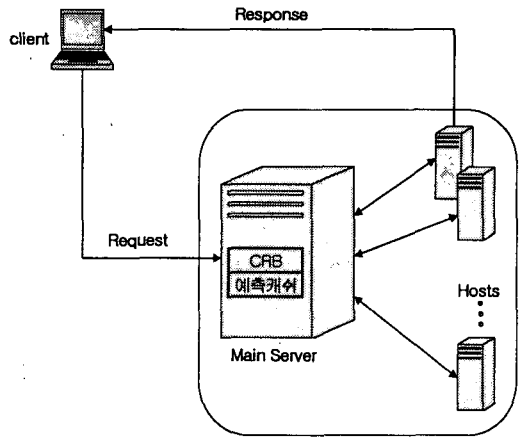
내용 기반 부하분배 알고리즘의 대표적인 것으로는 LARD(Locality-Aware Request Distribution) 방식이 있다[8]. 이 방식은 같은 내용의 요구를 같은 서버에 보냄으로써 캐쉬의 지역성을 최대한 이용하여 디스크 접근을 줄이는 방법이다. 본 논문에서는 같은 내용의 요구를 같은 서버가 처리할 수 있도록 하기 위해, 일종의 부하 분배기 역할을 Main Server가 하도록 하며, 각각의 호스트로 부하를 분배하는 알고리즘을 위해 각 파일의 메타정보를 이용하도록 하였다.

3. CRBM 기법의 제안

본 논문에서는 사용자들이 요청한 파일의 메타 정보를 Main Server의 캐쉬에 리스트 형태로 유지하는 CRBM(Client Request Buffer Manager)을 제안한다.

그림 1은 CRBM 시스템의 구조를 보여준다. 부하 분산 클러스터 시스템은 사용자들은 하나의 컴퓨터,

즉 Main Server로부터 모든 서비스를 제공받는 것으로 여기지만 실제 서비스의 제공은 Main Server에 연결되어 있는 여러 호스트들에 의해서 이루어지도록 하는 시스템이다.



(그림 1) CRBM 시스템의 구조

본 논문에서는 Main Server의 캐쉬를 분할하여 파일의 메타 정보를 관리하는 CRB(Client Request Buffer)와 최근에 자주 참조되었던 파일 블록을 선반입(pre-fetch) 하는 예측 캐쉬로 사용함으로써, 캐쉬 공간의 사용 효율을 높인다. 분할비는 Main Server 캐쉬 크기의 30%를 CRB에 할당하고 나머지 70%를 예측 캐쉬에 할당하도록 한다. 하지만 가변적인 웹 환경을 고려하여, 분할비를 주기적으로 변경할 수 있게 한다.

CRB는 하나의 파일에 대해 이름, 파일이 존재하는 호스트, 접근 시간, 접근 횟수로 구성된 메타 정보를 리스트 형태로 관리하며, 구조는 아래의 표 1과 같다.

<표 1> 메타정보 구조

File_name	Host	Access_time	Access_count
파일명	호스트	접근시간	접근횟수
256byte	1byte	10byte	3byte

CRB는 각 호스트가 사용자의 요구를 서비스할 때, Main Server로 보내는 해당 파일의 메타 정보로 구성된다. CRB에서는 메타 정보를 참조횟수에 따라 내림차순으로 정렬하며, 참조횟수가 동일할 경우는 최근에 참조된 파일의 순위를 높여준다. 이는 사용자의 요구를 처리할 때 발생하는 메타정보를 검색 시간을 줄이기 위한 것으로, 이전에 빈번하게 접속된 문서들은 앞으로도 다시 요청될 가능성이 크다는 문서의 편중된 요청 패턴에 의존한다[8].

Main Server의 캐쉬를 구성하는 두 번째 그룹인 예측 캐쉬는 CRB의 정보를 이용하여 최근에 가장 많이 참조된, 즉 CRB의 상위에 존재하는 파일 블록을 선반입(pre-fetch) 하여 사용자들의 요청빈도가 많은 파일 블록에 대해 적중률을 높여 준다. 예측 캐쉬는 일정한 시간 주기(약 12시간)로 교체하여 동적인 사용자들의 요구에 대해 최대한 높은 적중률이 발생할 수 있도록 유도한다.

본 시스템에서 사용자들의 요청은 Main Server에 전달된다. Main Server는 사용자가 요청한 파일 블록이 예측 캐쉬 내에 존재하면 Main Server가 바로 사용자에게 서비스한다. 그렇지 않을 경우는 CRB를 이용하는데, 요청된 파일의 메타정보가 존재하면 Main Server는 이를 이용하여, 사용자의 요청을 해당 호스트에 전달한다. 호스트는 요청 받은 데이터를 자신의 메모리나 혹은 디스크로부터 가져오며, Main Server를 거치지 않고 바로 사용자에게 응답을 보내는 직접 라우팅 방식을 이용한다.

만일 Main Server로부터 요청 받은 호스트에 사용자가 요구한 파일이 존재하지 않으면, Main Server는 다른 호스트에게 서비스를 요청하여, 파일 블록을 보유하고 있는 호스트가 서비스를 해줄 수 있도록 한다. 이때 서비스를 수행하는 호스트는 해당 파일의 메타 정보를 전달하여 CRB에 존재하는 파일의 이전 정보를 갱신하도록 한다.

제안하는 CRBM은 사용자가 요청한 각 파일의 메타정보만을 포함하므로 파일의 크기나 형식이 캐싱에 미치는 영향이 크지 않아 캐싱 공간 사용의 효율성을 높일 수 있다. 또한 이전의 협력 캐싱 알고리즘의 데이터 일관성 문제를 해결한다.

그리고 CRB의 정보를 이용하여 최근에 자주 참조되는 파일을 예측 캐쉬에 선반입함으로써, 사용자에 의해 빈번히 요청되는 파일에 대해 빠른 서비스가 가능하다. 또한 사용자의 요청을 해당 호스트가 서비스할 때, CRB의 리스트에는 메타정보가 추가되며, 해당 호스트의 캐쉬에는 서비스된 파일 블록이 존재하게 되므로, 이후 같은 요청에 대해 Main Server의CRB, 혹은 예측 캐쉬 내에서 뿐만 아니라, 각 호스트 내에서도 높은 캐쉬 적중률을 유지할 수 있다.

4. 결론 및 향후 과제

본 연구에서는 부하 분산 웹 서버 클러스터 시스템을 기반으로 하여 부하를 분배해주는 Main Server에 메타 정보만을 유지하는 CRBM기법을 제안하였다. 제안한 기법에서는 캐쉬를 CRB와 CRB를 이용하여 자주 참조되는 파일을 선반입하는 예측 캐쉬로 분할하여 운영함으로써, 사용자에게 빠른 응답시간을 보장해줄 수 있고, 모든 서비스를 Main Server가 처리하는 것이 아니라 각 호스트가 직접 라우팅 방식을 통해 사용자에게 바로 서비스 할 수 있도록 함으로써 Main Server의 과부하를 막을 수 있다.

향후 연구 과제로는 본 논문에서 제안된 시스템의 구현과 CRB와 예측 캐쉬를 갱신하는 최적의 주기를 도출하는 것이다.

[참고문헌]

[1] Jia Wang, "A survey of web caching schemes for the Internet". AGM Computer Communication Review, October 1999

[2] Prasenjit Sarkar, John Hartman, "Hint-Based Cooperative Caching", University of Arizona, February 1998

[3] Pei Cao, Sandy Irani, "Cost-Aware WWW Proxy Caching Algorithms", Proceedings of the USENIX Symposium on Internet Technologies and Systems Monterey, California, December 1997

[4] W.Ahn, S.Park, D.Park, "Efficient Cooperative Caching for File Systems in Cluster-Based Web Servers", Cluster Computing, 2000. Proceedings. IEEE International Conference on , 28 Nov.-1 Dec. 2000 Page(s): 326 -334

[5] Thmas Anderson, Michael Dahlin, Jeanna Neefe, David Patterson, Drew Roselli, Ran- dolph Wang, "Serverless Network File Sys- tems", In Proceedings of the 15th Sympo- sium on Operating System Principles, 1995

[6] Michael J.Feeley, William E. Morgan, Frederic H. Pighin, Anna R. Karlin, Henry M. Levy, "Implementing Global Memory Management in a Workstation Cluster", DEC Systems Research Center, 1995

[7] V. S. Pai, M. Aron, G. Bang, M. Svendsen, P. Druschel, W. Zwaenepoel, and E. Nahum, "Locality-Aware Request Server," 8th ACM Conference on Architectural Support for Programming Languages and Operating Systems, Vol.33, No. 11, pp. 205-216, 1998

[8] D. Lee, J. Choi, J. Kim, S. Noh, S. Min, Y. Cho, C. Kim, "LRFU : A Spectrum of Policies that Subsumes the Least Recently Used and Least Frequently Used Policies", IEEE Transactions on computers, VOL. 50, NO. 12, DECEMBER 2001

[9] Stephen Williams, Marc Abrams, Charles R. Standridge, Ghaleb Abdulla, and Edward A. Fox, "Removal policies in network caches for World-Wid Web documents", In Proceedings of the ACM SIGCOMM '96 Conference, Stanford University, CA, August, 1996

[10] 정지영, 김성수 " 웹 서버 클러스터에서 사용자 응답시간 개선을 위한 메모리 관리", 한국정보과학회 논문지 제 28권 제 9호, 2001, 10

[11] 신범주, 김경배, 김창수, 김명준, "네트워크 저장장치를 위한 클러스터 파일 시스템 개발", 한국정보처리 학회지 8권 제 4호, 2001, 7