

P2P 방식의 웹 캐싱 시스템에 대한 효용성 분석

장재완[○] 김영석* 홍정우** 맹승렬* 김진수*
*한국과학기술원 전자전산학과 전산학전공
**한국과학기술정보연구원 슈퍼컴퓨팅센터

{jwjang[○], kimys}@calab.kaist.ac.kr jwhong@hpcnet.ne.kr {maeng, jinsoo}@cs.kaist.ac.kr

On the Effectiveness of P2P-based web caching system

Jae-Wan Jang[○]* Youngseok Kim* Jungwoo Hong** Seung Ryoul Maeng* Jin-Soo Kim*
*Div. of Computer Science, Dept. of EECS., KAIST
**KISTI Supercomputing Center

요 약

웹의 발전과 더불어 웹 캐시 시스템의 중요성이 증대되면서, 다양한 구조의 웹 캐시 시스템이 제안되고 있으며 최근에는 Squirrel 과 같은 P2P 방식을 이용한 웹 캐시 시스템도 출현하였다. 본 논문에서는 Squirrel과 같은 일반적인 P2P 웹 캐시를 서버넷에 국한시킨 경우의 P2P 웹 캐시의 효용성을 분석한다. 이를 위하여, P2P 웹 캐시를 이용하는 사용자들의 웹 접근 성향을 웹 로그 트레이스로 시뮬레이션 하여, P2P 웹 캐시가 서버넷에서도 웹 캐시로서의 효용성이 있음을 보인다. 또한 현실적인 접근법으로써 중앙 프락시 서버를 두고 P2P 웹 캐시를 사용하는 경우에도 성능상의 효용성이 있음을 기술한다.

1. 서론

웹은 현재의 인터넷을 존재하게 한 가장 큰 주역으로 인정을 받고 있으며, 현재도 끊임없이 발전을 거듭하고 있다. 웹이 인터넷의 가장 중요한 요소라고 본다면, 사용자들이 웹을 사용하는 데 있어서 불편을 느끼지 않게 해 주는 가장 중요한 기술 중의 하나로 웹 캐시를 생각할 수 있다. 웹 캐시를 이용함으로써 사용자의 입장에서는 좀 더 신속하게 웹의 객체들을 접근할 수 있으며, 웹 서비스를 해 주는 기관의 입장에서는 외부 인터넷으로 나가야 하는 데이터의 양이 줄어들게 되는 이점을 가지게 된다.

웹 캐시가 처음 제안 되었을 때에는 사용자들의 컴퓨터에 일정한 양의 저장 공간을 할당해서 접근된 객체를 저장하는 방식이었다. 하지만 웹 캐시 기술이 점점 발달함에 따라 하나의 기관이나 사이트 전체에서 캐싱 서비스를 전담하는 중앙 프락시 캐시 서버 등을 두게 되었다. 최근에는 좀 더 효율적인 객체 캐싱을 위해서 프락시 서버들이 계층적인 구조를 형성하는 방식이 많이 연구되고 있다 [1].

한편, 웹이 기존의 인터넷 서비스에 새로운 지평을 열었듯이, 현재는 P2P(peer-to-peer) 기술이 인터넷 서비스의 새로운 패러다임을 제시하고 있다. P2P 기술은 기존의 서버-클라이언트 방식의 인터넷 서비스 방식에서 탈피하여 각각의 노드가 서버인 동시에 클라이언트의 역할을 하도록 되어 있다. 그러므로 서버-클라이언트 방식과는 달리 중앙 노드의 고장에 영향을 받지 않으며, 중앙 노드에 과도한 부하가 걸리는 일도 없다. 이에 따라 웹 캐싱 구조에서도 계층적인 서버-클라이언트 방식에서 벗어난 P2P의 장점을 살린 웹 캐싱 구조가 제안되고 있다.

본 논문에서는 서버넷에 속하는 한 노드에 존재하는 지역(local) 캐시 공간을 같은 서버넷에 존재하는 다른 노드의 지역 캐시 공간과 공유하는 방식의 효용성에 대해서 기술한다. 이와 같이 P2P 방식으로 한 노드가 다른 노드와 지역 캐싱 공간을 공유하는 방식을 본 논문에서는

P2P 웹 캐싱이라고 하기로 한다. P2P 웹 캐싱은 하나의 노드 입장에서는 자신의 지역 캐시 공간과 그 서버넷을 담당하는 중앙 캐시 서버의 중간에 존재하는 논리적인 캐싱 공간이라고 볼 수 있다. 본 논문에서는 P2P 웹 캐싱 방식이 효과적인지, 실제 웹 로그 트레이스를 이용해서 시뮬레이션하였으며, P2P 웹 캐싱이 특정 상황에서 효용성이 있음을 기술한다.

2. 관련연구

P2P 방식이 아닌 서버-클라이언트 방식의 협력 캐싱에 대해서는 이미 많은 연구가 진행되어 왔다 [1]. 협력 캐싱은 주로 중앙 프락시 웹 캐시들 사이에서 캐싱 공간을 공유하는 방식에 초점을 두고 있다. 일반적으로 하위에 존재하는 프락시 웹 캐시에서 캐싱 실패가 되는 경우 상위 프락시 웹 캐시의 캐싱 객체를 사용하는 계층적인 구조를 가지는 경우가 많다. 그리고 현재의 연구들은 이런 계층적인 구조를 가지는 서로 다른 웹 캐시들 사이에 일관성 문제 및 효율적인 통신 프로토콜의 고안에 많은 비중을 두고 수행되고 있다 [2][3].

P2P 방식으로 웹 캐싱을 하는 경우 객체의 위치와 같은 메타 정보를 중앙 집중적으로 저장하는 공간이 존재하지 않기 때문에, 사용하고자 하는 객체를 찾는 오버헤드가 매우 크다는 점이 문제가 된다. 이와 같은 문제점을 해결하기 위해서, Pastry [4], Chord [5] 와 같은 P2P 객체 위치 탐색 서비스 프로토콜들이 제안되었다. Pastry P2P 객체 위치 탐색 서비스 프로토콜을 이용해서 웹 캐시 분야에서도 Squirrel [6] 과 같은 P2P를 이용한 웹 캐시 구조가 제안되었다. 또한 Mangosoft [7] 에서도 이와 유사한 Cachelink [8] 같은 시스템을 제안하였다.

본 논문에서는 Squirrel 같은 일반적인 P2P 방식의 웹 캐시를 서버넷으로 한정을 하였을 때에도 효과적으로 적용이 되는지를 기술한다. 또한 Mangosoft가 이미 근거리 통신망(LAN)에서 P2P를 이용한 웹 캐

시를 개발하였지만, 공개된 정보가 거의 없기 때문에, 본 논문에서는 캐시의 범위를 서브넷으로 한정된 P2P 웹 캐시에 초점을 두며, 향상된 캐싱 구조의 가능성을 기술한다.

3. 캐싱 방식

하나의 서브넷을 기준으로 웹 객체를 캐싱할 수 있는 물리적인 공간은 크게 두 가지로 나뉘볼 수 있다. 본 논문에서는 실제 웹 서비스를 사용하는 노드에 존재하는 지역 캐시와 서브넷 전체를 위해서 객체들을 캐싱하는 중앙 캐시를 가정한다. 그리고 논리적인 캐싱 공간인 P2P 웹 캐시가 존재한다. P2P 웹 캐시를 이용하면 지역 캐시에 존재하지 않는 객체를 인접한 노드의 지역 캐시에서 서비스를 받음으로써, 실제로는 인접한 노드의 캐시를 이용하지만 논리적으로는 자신의 캐시로 사용하는 공간이 형성된다.

P2P 웹 캐싱이 효용성이 있는지 살펴보기 위해서 다음과 같이 크게 3개의 설정을 하고 시뮬레이션을 수행하였다.

- Conf.1) 지역 캐시와 중앙 캐시만이 존재하는 경우
- Conf.2) 지역 캐시와 P2P 웹 캐시만이 존재하는 경우
- Conf.3) 지역 캐시, P2P 웹 캐시 그리고 중앙 캐시가 모두 존재하는 경우

Conf.1은 일반적인 캐싱 방식으로, 지역 캐시에서 캐시 실패인 경우, 중앙 캐시에서 서비스를 받으며, 중앙 캐시에서도 서비스가 불가능한 경우에는 실제 웹 사이트로부터 요청한 객체를 받아오는 방식이다.

Conf.2는 P2P 방식의 웹 캐시를 이용하는 경우로써, 지역 캐시에서 캐시 실패인 경우, 인접한 노드의 지역 캐시에서 요청한 객체를 서비스 받는다. P2P 웹 캐시에서 서비스를 못 받는 경우, 즉 인접한 캐시들에서도 요청한 객체를 저장하고 있지 않는 경우에는 실제 웹 사이트로부터 요청한 객체를 받는다.

Conf.3은 Conf.1과 Conf.2의 방식을 혼합한 방식으로, 지역 캐시에서 캐시 실패이면, P2P 웹 캐시에서 서비스를 받으며, P2P 웹 캐시에서도 서비스가 불가능한 경우에는 중앙 캐시로부터 서비스를 받는다. 마찬가지로, 중앙 캐시에서도 캐시 실패인 경우에는 요청한 객체를 실제 웹 사이트로부터 서비스를 받는다.

4. 시뮬레이션 결과 및 분석

4.1 실험 환경

시뮬레이션에 사용한 웹 로그 트race는 한국과학기술원(KAIST)의 중앙 프락시 웹 캐시에서 나온 웹 접근 로그이며, 트race 중에서 가장 전송량이 많은 두 개의 서브넷을 대상으로 시뮬레이션하였다. 트race의 특성은 표1에 요약하였다.

서브넷에 존재하는 모든 노드는 항상 사용 가능하다고 가정하였으며, 캐싱하는 객체는 정적인 객체만을 대상으로 한다. 동적인 객체는 웹 캐시들의 특성에 따라서, 캐싱을 하기도 하고 하지 않기도 하기 때문에 모두 캐싱 대상에서 제외하였다. 객체의 이름에 php, asp, cgi, cfm, jsp, ?, =, &, @ 등이 포함된 경우에는 모두 동적인 객체로 처리하도록 하였다. 동적인 객체는 이 논문의 논의 대상이 아니므로, 캐싱 성공률은 캐시 가능한 객체들을 대상으로 계산하였다.

또한 시뮬레이션에서 사용되는 모든 지역 캐시, 중앙 캐시는 캐싱 공간이 가득한 경우에 모두 LRU 교체 알고리즘을 이용해서 캐싱 공간을 확보하도록 하였다.

표 1 시뮬레이션에 사용된 웹 로그의 특성

	Subnet A	Subnet B
총 객체 크기	12.3GB	20.8GB
기간	167시간	
객체 요청 수	2294396	2198523
활성 노드 수	172	158
캐시 성공률	83.35%	84.42%

4.2 Conf.1과 Conf.2의 비교

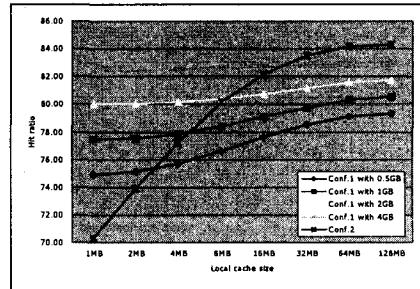


그림 1 서브넷 B에서 Conf.1 과 Conf.2의 비교

그림1은 B 서브넷에서, 지역 캐시 크기를 다르게 하는 경우에 중앙 캐시(Conf.1)와 P2P 웹 캐시(Conf.2)의 캐시 성공률을 비교한 것이다. 중앙 캐시의 경우, 0.5GB에서 4GB까지 변화를 주면서, 성능을 시뮬레이션하였다.

지역 캐시 공간이 부족하여 전체 캐싱 공간이 부족한 경우에는 Conf.1이 Conf.2보다 더 높은 캐시 성공률을 보여준다. 하지만, P2P 웹 캐시의 경우, 지역 캐시 공간이 증가하여 전체 캐싱 공간이 충분히 확보되면 중앙 캐시를 사용하는 경우와 비슷한 수준의 캐시 성공률을 보여준다.¹⁾

좀 더 정량적으로, 서브넷 전체를 담당하는 캐싱 공간을 기준으로 비교하면, 그림 2와 같다.

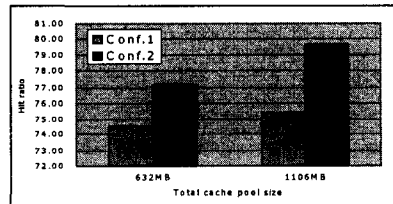


그림 2 전체 캐싱 공간이 동일한 크기인 경우

전체 캐싱 공간은 다음과 같은 식으로 계산된다.

$$\text{전체 캐싱 공간} = \text{중앙 캐시 크기} + \text{각 노드의 지역 캐시} \times \text{서브넷의 활성 노드 수}$$

전체 캐시 크기 632MB의 경우에, Conf.1은 중앙 캐시 474MB, 각 노드의 지역 캐시 1MB로 시뮬레이션 하였으며, Conf.2는 중앙 캐시는 없으며, 각 노드의 지역 캐시 4MB로 하였다. 1106MB에서는 중앙 캐시 474MB, 지역 캐시 4MB의 Conf.1과, 지역 캐시 7MB의 Conf.2를 비교하였다.

같은 크기의 전체 캐싱 공간의 경우 P2P 웹 캐시를 사용하게 되면 캐시 성공률이 중앙 캐시를 사용하는 경우보다 조금 더 높게 나타나는 것을 볼 수 있다. 이는 좁은 범위의 서브넷을 대상으로 한 P2P 웹 캐

1) A 서브넷도 거의 동일한 양상을 보였으나, 지면 관계상 생략한다.

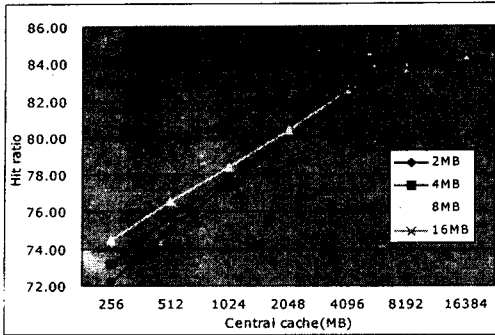


그림 3 Conf.1

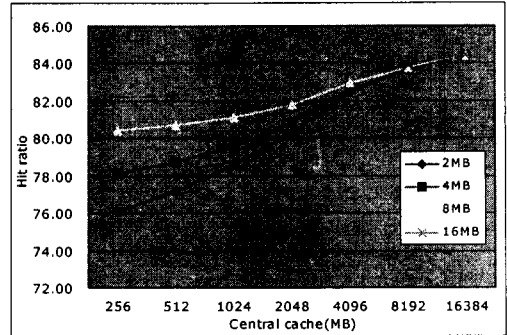


그림 4 Conf.3

시도 좀 더 넓은 범위의 일반적인 LAN을 대상으로 한 P2P 웹 캐시인 Squirrel과 비슷한 성능을 보여줌으로써, 그 효용성을 보여준다. 하지만 일반적으로 P2P를 이용한 인터넷 서비스의 경우, 공유하는 객체들을 찾는 데 드는 비용이 커서, 서버-클라이언트 방식보다 성능이 떨어지는 단점이 존재하기도 한다. 마찬가지로, P2P 웹 캐싱의 경우에도 캐시 객체를 찾을 때에 중앙 캐시에서 캐시 객체를 가져오는 것보다 큰 오버헤드가 예상되므로, 캐시 성공률은 조금 더 좋다고 하더라도 실제로는 더 낮은 성능을 보일 것으로 추측된다.

4.3 Conf.1와 Conf.3의 비교

P2P 웹 캐싱만을 사용하는 경우에는 객체 위치를 찾는 데 오버헤드가 크다고 예상되고, 현실적으로 P2P 웹 캐시는 개개인의 협조가 있어야 사용이 가능하므로, 중앙 캐시를 돕는 보조적인 수단으로 사용하는 경우의 P2P 웹 캐시의 효용성을 분석하였다.

그림3, 그림4는 서버넷 B에서 지역 캐시 크기와 중앙 캐시 크기에 따라 변하는 Conf.1과 Conf.3의 캐시 성공률을 비교한 것이다. Conf.1과 Conf.3을 시뮬레이션하는 환경은 동일한 환경이다.²⁾

Conf.3은 중앙 캐시와 이웃 캐시를 모두 사용하므로, Conf.1보다 더 높은 캐시 성공률을 보여주고 있다. 전체 캐싱 공간이 작을 때는, Conf.3이 6% 정도 더 높은 캐시 성공률을 보이며, 중앙 캐시 크기가 더 커질수록 B 서버넷의 캐싱 한계인 84.42%에 이르는 것을 알 수 있다.

그리고 Conf.1와 Conf.3에서 같은 캐시 성공률을 얻기 위해서 필요한 중앙 캐시의 크기는 매우 큰 차이를 보여준다. 예를 들어서, 지역 캐시의 크기가 8MB인 경우에, 81%의 캐시 성공률을 얻기 위해서, Conf.3에서는 약 256MB의 중앙 캐시가 필요하지만, Conf.1에서는 중앙 캐시가 8배인 2GB가 필요하다.

이와 같이 전체 캐싱 공간의 크기가 작은 환경에서, P2P 웹 캐시를 중앙 캐시와 함께 사용하는 경우, 같은 캐시 성공률을 얻는데 필요한 중앙 캐시 크기가 매우 줄어들음을 볼 수 있다. 실질적으로는 P2P 웹 캐시를 사용하기 때문에 줄어든 캐시 공간에 다른 캐시 객체를 저장할 수 있는 여지를 가지고 있으므로 논리적으로 더 큰 전체 캐싱 공간을 확보할 수 있음을 예상할 수 있다.

5. 결론 및 향후 연구

본 논문에서는 서버넷에서의 P2P 웹 캐싱 방식의 효용성을 기술하였다. P2P 웹 캐싱 자체로는 중앙 캐시와 비슷한 정도의 캐시 성공률을 보여주지만, 객체 위치 탐색 오버헤드가 존재한다. 또한 현실적으로

P2P 웹 캐시를 모든 사용자들이 사용하게 하는 것이 쉽지 않으므로, 완전하게 중앙 캐시를 대체하기는 어려울 것으로 보이며, 대신 중앙 캐시의 보조적인 수단으로는 효과가 있을 것으로 예상할 수 있다. 중앙 캐시와 함께 P2P 웹 캐싱을 이용하면, 같은 캐시 성공률을 얻기 위해서 필요한 중앙 캐시의 크기의 상당량을 줄일 수 있으며, 줄어든 만큼의 공간이 다른 캐싱 객체를 저장할 수 있는 여지가 되므로, 여기에서 P2P 웹 캐싱의 효용성을 생각할 수 있다.

하지만 여전히 P2P 웹 캐싱은 P2P 구조의 가장 큰 문제점인 객체 위치 탐색에 오버헤드를 가지고 있으므로, 효율적인 객체 위치 탐색 알고리즘의 개발이 필요하다. 이미 P2P 객체 위치 탐색 프로토콜은 많이 연구되고 있으나, 일반적인 프로토콜에 중점을 두고 있으므로, 본 논문에서 초점을 두고 있는 서버넷의 특성을 이용한 P2P 객체 위치 탐색 프로토콜을 찾고 적용시킬 필요가 있다.

또한 본 논문에서는 단일 웹 프락시 로그를 이용해서 시뮬레이션 하였으므로 일반적인 서버넷의 특성을 보여준다고 일반화하기는 어려우므로, 좀 더 다양한 웹 프락시 로그들에 대해서 시뮬레이션 할 필요가 있다.

6. 참고문헌

- [1] M. Raunak, A Survey of Cooperative Caching, 1999
- [2] L. Fan, P. Cao, J. Almeida, and A. Broder, Summary Cache: A Scalable Wide-Area Web Cache Sharing Protocol, In *Proc. SIGCOMM'98*, 1998
- [3] Alex Rousskov and Duane Wessels, Cache Digests, In *Proc. 3rd Intl. Web Caching Workshop*, 1998
- [4] A. Rowstron and P. Druschel, Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems, In *Proc. 18th IFIP/ACM Middleware2001*, 2001
- [5] I. Stoica, R. Morris, D. Karger, M. Kaashoek, and H. Balakrishnan, Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications, In *Proc. SIGCOMM'01*, 2001
- [6] S. Lyer, A. Rowstron, and P. Druschel, Squirrel: A decentralized peer-to-peer web cache, In *Proc. ACM Symposium on PODC'02*, 2002
- [7] Mangosoft <http://www.mangosoft.com>
- [8] Mangosoft's Cachelink <http://www.mangosoft.com/products/cachelink/>

2) 서버넷 A도 서버넷 B와 비슷한 양상을 가지고 있으나, 지면관계상 서버넷 A의 실험결과는 첨부하지 않는다.