

웹 캐싱을 위한 P2P 협력적인 캐시 교체 정책

김상훈^o 류영석 양성봉
연세대학교 컴퓨터과학과
{huniky^o, ryu, yang}@cs.yonsei.ac.kr

P2P Cooperative Cache Replacement Policy for Web Caching

SangHoon Kim^o, YoungSuk Ryu, SungBong Yang
Dept. of Computer Science, Yonsei University

요 약

P2P 웹 캐싱 기술은 자신의 로컬 캐시에서 원하는 웹 객체를 찾지 못한 경우, 중앙 서버나 프락시 서버로부터 웹 객체를 받기 전에 P2P 네트워크에 참여하는 이웃으로부터 웹 객체를 받음으로써 기존의 서버-클라이언트 방식을 탈피하여 중앙 서버의 고장에 영향을 적게 받고, 중앙 서버로의 과도한 부하를 막아준다. 기존의 많은 P2P 웹 캐싱 시스템은 피어의 로컬 캐시 교체 정책으로 LRU기법을 사용한다. 하지만 이는 P2P 환경을 고려하지 않는 단순한 정책이다. 따라서 본 논문에서는 P2P 환경을 고려한 디렉토리 기반의 웹 캐싱 시스템을 통해서 P2P 협력적인 캐시 교체 정책을 제시하고 실제 웹 로그 트래이스로 시뮬레이션을 하여, 제안하는 교체 정책이 효과가 있음을 기술한다.

1. 서 론

P2P(peer-to-peer) 웹 캐싱(web caching) 기술은 자신의 로컬 캐시(local cache)에서 원하는 웹 객체(web object)를 찾지 못한 경우, 중앙 서버(central server)나 프락시 서버(proxy server)에서 웹 객체를 받기 전에 P2P 네트워크에 참여하는 다른 피어, 즉 이웃(neighbor)으로부터 웹 객체를 받음으로써 기존의 서버-클라이언트 방식을 탈피하여 중앙 서버의 고장에 영향을 적게 받고, 중앙 서버로의 과도한 부하를 막아준다.

이웃을 탐색하는 방식에 따라 P2P 웹 캐싱 시스템은 크게 3가지로 나누어진다. 첫째, 디렉토리 기반(directory-based) [1],[2]의 방식이다. 이는 중앙 서버에서 각 웹 객체를 가진 클라이언트들을 등록해 놓고 웹 객체를 요청한 클라이언트에게 해당 웹 객체를 소유한 클라이언트 목록을 전달해 주는 시스템으로 단순하지만 중앙 서버의 고장에 영향을 받는다. 둘째, 플러딩 기반(flooding-based) [3]의 방식이다. 이는 자신과 인접한 이웃들을 통해서 원하는 웹 객체를 찾을 수는 있지만, 웹 객체를 찾는 과정에서 많은 룩업(lookup) 메시지로 인해 네트워크에 과도한 부하가 생기게 된다. 셋째, DHT 기반(DHT-based) [4]의 방식이다. 이 시스템은 원하는 웹 객체를 반드시 찾을 수 있지만, 각 웹 객체마다 하나의 이웃만이 존재하기 때문에 요청이 많은 웹 객체를 소유한 피어는 많은 부하를 가지게 된다.

본 논문에서는 디렉토리 기반의 P2P 웹 캐싱 시스템을 이용하여 P2P 환경에 적합한 피어의 로컬 캐시 교체 정책에 관해서 기술한다. 기존의 많은 P2P 웹 캐싱 시스템의 피어 로컬 캐시 교체 정책은 LRU기법을 사용하였다. 하지만 이는 P2P 환경을 고려하지 않는 정책으로 P2P를 고려한 피어의 로컬 캐시 교체 정책이 필요하다. 따라서 본 논문에서 제안하는 피어의 로컬 캐시 교체 정책이 P2P 웹 캐싱 환경에서 효과적인지, 실제 웹 로그 트래이스를 이용하여 시뮬레이션을 하였다.

2. 관련연구

웹 캐싱 기술은 클라이언트, 프락시 그리고 서버에 캐시 공

간을 동으로써 네트워크의 트래픽(traffic)과 서버의 부하를 줄이는 목적으로 사용되어진다. 효과적인 웹 캐싱에 관한 연구는 P2P 방식이 아닌 일반 서버-클라이언트 방식을 위한 프락시 서버에서의 연구 [5]가 많이 진행되어 왔다. 또한, 클라이언트의 캐시 토폴로지(topology)에 관한 연구 [6]도 진행되었다.

웹 캐싱을 위한 많은 캐시 교체 정책이 있다. 첫째, 최신 기반(recency-based)의 방식이다. 이는 웹 객체를 참조한 시간을 바탕으로 참조된 지 가장 오래된 웹 객체가 캐시에서 교체될 대상이 된다. 대표적인 예로 LRU를 들 수 있고, 웹 캐싱의 성능 향상을 위해서 LRU를 변형한 캐시 교체 정책 [7]이 존재한다. 둘째, 빈도 기반(frequency-based)의 방식이다. 이는 웹 객체를 요청한 횟수를 바탕으로 가장 적게 참조된 웹 객체가 교체 대상이 된다. 대표적인 예로 LFU를 들 수 있다. 셋째, 함수 기반(function-based)의 방식이다. 이는 각 웹 객체에 대해서 여러 요소를 이용하여 각 요소에 가중치를 두고 수식을 이용해서 가장 적은 값을 갖는 웹 객체가 교체 대상이 된다. 대표적인 예로 GD-Size를 들 수 있다.

본 논문에서는 효과적인 웹 캐싱을 위한 중요한 요소인 인기도(popularity)를 이용하여 P2P 환경에 적합한 클라이언트의 캐시 교체 정책을 제안하고 기존의 웹 캐싱에서 사용된 캐시 교체 정책과 시뮬레이션을 통해서 비교하며 제안한 캐시 교체 정책이 효과적임을 기술한다.

3. LPU (Least Popularly Used)

3.1 피어의 상태

피어는 일정 주기 T 동안 원하는 웹 객체들을 찾기 위해서 총 N 번의 로컬 캐시를 액세스(access)하게 된다. 각 피어는 자신의 로컬 캐시를 모니터링(monitring)함으로써 자신의 상태를 감지한다.

$$\frac{N}{T} = \frac{N_0}{T} + \frac{N_P}{T}$$

T 는 일정 주기, N 은 주기 T 동안 로컬 캐시를 액세스한 총 횟수, N_0 은 주기 T 동안 자신이 요청한 웹 객체를 찾기 위해서 로컬 캐시를 액세스한 횟수, N_P 은 주기 T 동안 다른 피어

가 요청한 웹 객체를 찾기 위해서 로컬 캐시를 액세스한 횟수이다.

피어의 상태는 임계값 α, β, γ 에 의해서 결정되어진다.

1. $\alpha \leq \frac{N_p}{T}$ 이면, 피어는 very high 상태이다.
2. $\beta \leq \frac{N_p}{T} < \alpha$ 이면, 피어는 high 상태이다.
3. $\gamma \leq \frac{N_p}{T} < \beta$ 이면, 피어는 low 상태이다.
4. $\frac{N_p}{T} < \gamma$ 이면, 피어는 very low 상태이다.

피어의 상태는 자신이 원하는 웹 객체를 찾기 위해서 로컬 캐시를 얼마나 액세스했는가에 달려 있다. 이는 로컬 캐시 내에 웹 객체를 교체하는데 있어서 중요한 요소가 된다.

3.2 웹 객체의 타입

P2P에 참여하는 피어들 간의 협력(cooperation)을 증대시키기 위해서 로컬 캐시에 저장되는 웹 객체는 다음과 같이 4가지 종류로 나누어진다. 이는 로컬 캐시 내에 웹 객체가 가득 찬 경우, 효율적으로 웹 객체들을 교체하여 로컬 캐시 공간을 확보하도록 하기 위함이다.

표 1 웹 객체의 종류

웹 객체의 타입	설명
1	자신에게 인기 있는(popular) 웹 객체, 다른 피어에게 인기 있는 웹 객체
2	자신에게 인기 있는 웹 객체, 다른 피어에게 인기 없는(unpopular) 웹 객체
3	자신에게 인기 없는 웹 객체, 다른 피어에게 인기 있는 웹 객체
4	자신에게 인기 없는 웹 객체, 다른 피어에게 인기 없는 웹 객체

웹 객체에 대한 자신과 다른 피어의 인기 여부는 주기 T 동안 해당 웹 객체에 대한 요청 횟수의 임계값 ψ, ω 에 의해서 각각 결정되어진다.

3.3 캐시 교체 정책

로컬 캐시 내의 웹 객체들은 표1에서 정의한 웹 객체 타입 중 한 가지 타입을 갖게 되고 로컬 캐시가 가득 찬 경우, 교체될 웹 객체를 선택하기 위해서 피어의 상태와 다른 피어의 요청률, $\frac{N_p}{T}$ 의 임계값 ϵ 에 의해서 교체될 웹 객체를 결정하게 된다.

표 2 로컬 캐시 내 웹 객체 교체 순서

피어의 상태	웹 객체 교체 순서	
very high	$\frac{N_p}{T} \geq \epsilon$	4 → 3 → 2 = 1
	$\frac{N_p}{T} < \epsilon$	4 = 3 → 2 = 1
high	$\frac{N_p}{T} \geq \epsilon$	4 → 3 → 2 → 1
	$\frac{N_p}{T} < \epsilon$	4 = 3 → 2 = 1
low		4 → 2 → 3 → 1

	$\frac{N_p}{T} \geq \epsilon$	
	$\frac{N_p}{T} < \epsilon$	4 → 3 = 2 = 1
very low	$\frac{N_p}{T} \geq \epsilon$	4 → 2 → 3 = 1
	$\frac{N_p}{T} < \epsilon$	4 → 3 = 2 = 1

LPU는 각 웹 객체의 타입을 정의하여 피어의 상태와 다른 피어의 요청률에 따라서 교체될 웹 객체를 선택한다. 예를 들어, 피어의 상태가 very high이고 $\frac{N_p}{T} \geq \epsilon$ 인 경우, 교체될 웹 객체를 찾는 순서는 4 → 3 → 2 = 1 이다. 새롭게 삽입될 웹 객체의 공간이 확보될 때까지, 웹 객체의 타입 순서대로 교체할 웹 객체를 선택하게 된다. 2=1과 같은 경우에는 두 웹 객체 타입에 대해서 LRU를 적용하여 교체될 웹 객체를 선별하게 된다.

4. 실험 결과 및 분석

4.1 실험 환경

시뮬레이션에 사용된 웹 로그 트레이스는 WorldCup'98 데이터셋(dataset) [8]이다. WorldCup'98 데이터셋은 타임스탬프, 클라이언트 아이디, 웹 객체 아이디, 웹 객체 사이즈로 구성되어 있다. 시뮬레이션을 위해서 임의로 10일의 로그를 선택하였고 또한 각 10일의 로그에 대해서 임의로 100명의 클라이언트들을 1시간씩 선택하였다.

웹 객체의 일관성을 유지하기 위해서, 웹 객체의 사이즈가 변경된 경우, 웹 객체는 업데이트된 것으로 간주하였다.

표 3 WorldCup'98 데이터셋

WorldCup'98	
총 요청 수	1,352,804,107
기간	1998/4/30 - 1998/7/26
총 클라이언트 수	2,770,108
총 웹 객체 수	89,997
평균 웹 객체 크기	3962.683846 byte

시뮬레이션에서 순수한 P2P 웹 캐싱을 위하여 중앙 서버와 피어 사이에 존재하는 프락시 서버는 두지 않았으며, 각 피어는 동시에 여러 피어들에게 웹 객체를 제공해 줄 수 있다고 가정하였다. 또한, 시뮬레이션에서 피어들은 정적인 참여(static participation)를 하는 것을 간주하였고, 중앙 서버는 각 웹 객체에 대해서 최대 5명의 이웃을 보유한다.

표 4 시뮬레이션 기본 파라미터

파라미터	값
T	300 초
α	1.0
β	0.7
γ	0.4
ψ	0.02
ω	0.02
ϵ	0.4

시뮬레이션의 결과는 10일 로그에 대한 평균값으로 표시하였다.

4.2 적중률

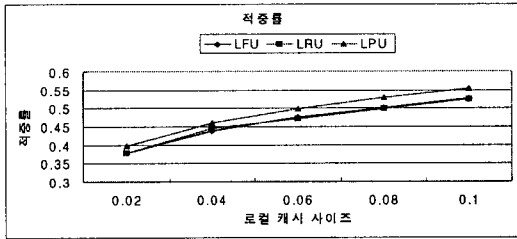


그림 1 로컬 캐시 사이즈에 따른 적중률

4.3 바이트 적중률

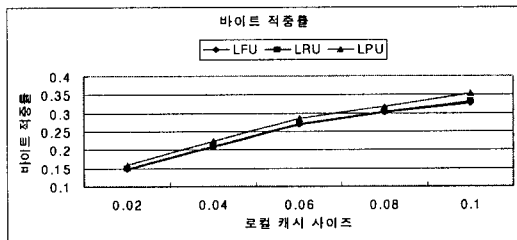


그림 2 로컬 캐시 사이즈에 따른 바이트 적중률

그림 1과 그림 2를 통해서 LPU가 다양한 캐시 사이즈에 대해서 LRU와 LRU보다 더 좋은 성능을 보여준다.

4.3 로컬 적중률

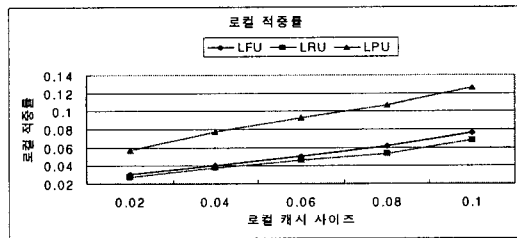


그림 3 로컬 캐시 사이즈에 따른 로컬 적중률

4.4 로컬 바이트 적중률

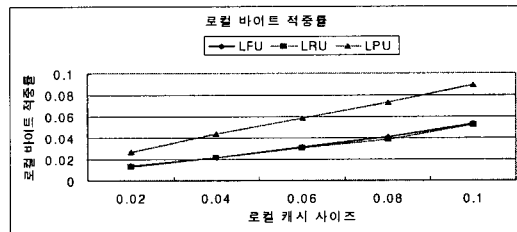


그림 4 로컬 캐시 사이즈에 따른 로컬 바이트 적중률

그림 3과 그림 4를 통해서 LPU가 전체 캐시 적중률 중 로컬에서의 캐시 적중률이 LRU와 LRU보다 더 높은 것을 볼 수 있다.

5. 결론 및 향후 연구

본 논문에서는 디렉토리 기반의 순수 P2P 웹 캐싱을 위한 효율적인 피어의 로컬 캐시 교체 정책을 제안하였다. 로컬 캐시 사이즈가 커짐에 따라 제안한 캐시 교체 정책인 LPU가 LRU와 LRU보다 더 좋은 성능을 보여주는 것을 알 수 있다. 본 논문에서 제시한 캐시 교체 정책은 자신과 다른 피어에게 인기 있는 웹 객체를 캐시 내에 오랫동안 가지고 있기 때문에 자신과 다른 피어의 캐시 적중률을 높여 준다. 따라서 전체 캐시 적중률 또한 높아진다. 즉, LPU는 자신에게 인기 있는 웹 객체 뿐 아니라 다른 피어에게 인기 있는 웹 객체도 로컬 캐시 내에 가지고 있음으로써 협력적인 P2P 환경을 고려하여 전체 캐시 적중률을 높여 준다는 것을 보여주었다.

본 논문에서 제시한 LPU를 모바일 환경에 적용, 실험 중이다. 모바일 환경은 일반 웹 환경과는 다르게 각 클라이언트가 각기 다른 속도와 방향으로 움직인다. 또한 클라이언트가 웹 객체를 전송할 수 있는 범위가 제한되어 있기 때문에 P2P 간의 웹 캐싱을 위한 효율적인 캐시 교체 정책이 필요하다. 따라서 본 논문에서 제시한 LPU를 바탕으로 모바일 환경에서의 성능을 실험 중이다.

6. 참고 문헌

- [1] K. Kong and D. Ghosal, "Mitigating server-side congestion in the Internet through pseudoserving," IEEE/ACM Transactions on Networking, vol.7, no.4, pp.530-544, 1999.
- [2] V.N. Padmanabhan and K. Sripanidkulchai, "The case for cooperative networking," Proc. International Workshop on Peer-to-Peer Systems, pp.178-190, 2002.
- [3] A. Stavrou, D. Rubenstein, and S. Sahu, "A lightweight, robust P2P system to handle flash crowds," IEEE Journal on Selected Areas In Communications, vol.22, no.1, pp.6-16, 2004.
- [4] S. Iyer, A. Rowstron, and P. Druschel, "Squirrel: A decentralized peer-to-peer Web cache," Proc. Symposium on Principles of Distributed Computing, pp.213-222, 2002.
- [5] Shudong Jin and Azer Bestavros, "Popularity-Aware GreedyDual-size Web Proxy Caching Algorithms", Proc. of the 20th International Conference on Distributed Computing Systems, pp.254-261, 2000.
- [6] Zhiyong Xu, Yiming Hu, Laxmi Bhuyan, "Exploiting Client Cache: A Scalable and Efficient Approach to Build Large Web Cache", Proc. of the 18th International Parallel and Distributed Processing Symposium, p.55, 2004.
- [7] Kai Cheng and Yahiko Kambayashi, "LRU-SP : A Size-Adjusted and Popularity-Aware LRU Replacement Algorithm for Web Caching", Proc. of the 24th International Conference. IEEE Computer Society, Piscataway, NJ, pp.48-53, 2000.
- [8] <http://ita.ee.lbl.gov>