

피어의 동적 참여 환경에서 오브젝트 라이프타임 기반 피어-투-피어 웹 캐싱

류영석*, 양성봉*

*연세대학교 컴퓨터과학과

e-mail:ryu@cs.yonsei.ac.kr

Object Lifetime-based P2P Web Caching under Dynamic Participation of Peers

Young-Suk Ryu*, Sung-Bong Yang*

*Dept of Computer Science, Yonsei University

요 약

P2P(peer-to-peer) 웹 캐싱 모델은 서버 쪽에 집중되는 트래픽을 완화시킴으로써 전통적인 웹 캐싱 모델을 보완할 수 있다는 측면에서 최근에 활발히 연구되어 왔다. P2P 웹 캐싱은 클라이언트들의 로컬 캐쉬를 활용하여 부가적인 인프라의 추가없이 캐쉬 공간이 확대되는 효과를 얻을 수 있지만, 각 클라이언트들은 독립된 피어로서의 자율성(autonomy)을 가지므로 이러한 자율성의 제한을 최소화해야 한다. 본 논문에서는 피어의 자율적인 동적 참여와 로컬 캐싱 전략을 보장하여 시스템의 실행 가능성(feasibility)을 높인 환경에서 효율적인 디렉토리 기반 P2P 웹 캐싱 시스템을 제안하였다. 제안하는 P2P 웹 캐싱 시스템은 동적인 P2P 네트워크 상에서의 오브젝트의 lifetime을 예상하여 이를 저장 공간 관리(storage management)에 적용하였다. 시스템의 성능 평가를 위하여 클라이언트의 http 요청 로그 데이터셋을 이용하여 트레이스 기반(trace-driven) 실험을 수행하고, 제안하는 시스템이 기존의 시스템에 비하여 주어진 동일한 환경에서 더 높은 정확성을 가짐을 확인하였다.

1. 서론

P2P (peer-to-peer) 웹 캐싱에서는 클라이언트 각각이 피어로서 P2P 네트워크를 형성하며, 클라이언트가 웹 오브젝트를 요청했을 때 자신의 로컬 캐쉬에 해당 오브젝트가 없다면 해당 오브젝트는 P2P 네트워크 상에서 찾은 이웃(neighbor) - 해당 오브젝트를 과거에 이미 전송받은 클라이언트 -으로부터 전송받게 된다. 따라서 P2P 웹 캐싱은 부가적인 인프라의 추가 없이도 피어들의 로컬 캐쉬들을 활용함으로써 캐쉬 공간의 확대와 트래픽의 분산 효과를 얻을 수 있다.

그러나 피어의 로컬 캐쉬는 피어 자신을 위해 우선적으로 쓰여져야하기 때문에 기존의 협력 캐싱(cooperative caching) 모델에서의 전용(dedicated) 캐쉬처럼 사용되기는 힘들다. 즉, 피어의 자율성(autonomy)으로 인하여 우선, 피어에게 P2P 네트워크에 일정기간 머물러야하는 의무, 피어가 직접 요청하지 않은 오브젝트를 강제적으로 피어의 로컬 캐

쉬에 저장하도록 하는 등의 부가적인 부담(load)을 지우는 것을 최소화해야 한다. 또한, 네트워크 상에 항상 상주하는 전용 캐쉬와 달리, 피어는 자신의 의지를 가지고 동적으로 P2P 네트워크로의 참여/탈퇴(dynamic join/leave)를 반복하게 된다. 따라서 이러한 동적 참여/탈퇴로 인한 시스템 운영 오버헤드나 성능 저하를 최소화해야 한다.

본 논문에서는 피어의 자율적인 동적 참여와 로컬 캐싱 전략을 보장하여 시스템의 실행 가능성(feasibility)을 높인 환경에서 효율적인 디렉토리 기반 피어-투-피어 웹 캐싱 시스템을 제안하였다. 우리는 P2P 네트워크 상에 존재하는 오브젝트의 라이프타임(lifetime)을 예상하고, 이를 제안하는 시스템 프레임워크의 저장 공간 관리(storage management) 단계에서 활용하여 시스템의 성능 향상을 꾀하였으며, 제안하는 시스템을 OLP(Object Lifetime-based P2P web caching system)로 명명하였다.

시스템의 성능 평가를 위하여 클라이언트의 http

요청 로그 데이터셋을 이용하여 트레이스 기반(trace-driven) 실험을 수행하였다. 실험의 결과를 통하여 제안하는 시스템이 기존의 디렉토리 기반 P2P 웹 캐싱 시스템에 비하여 주어진 동일한 환경에서 더 높은 정확성을 가짐을 확인한다.

논문의 구성은 다음과 같다. 2장에서는 관련 연구를 소개하며, 3장에서는 본 연구에서 제안하는 P2P 웹 캐싱 시스템을 기술한다. 4장에서는 실험을 통한 성능 평가 결과를 설명하고, 5장에서는 결론 및 향후 연구 진행 방향을 제시한다.

2. 관련 연구

Pseudoserving [1]은 선구적인 P2P 웹 캐싱 시스템의 하나이며 서버의 트래픽 병목 현상을 완화시키는데 초점을 맞추고 있다. 그러나 피어의 동적 참여에 따른 P2P 네트워크 관리의 어려움을 해결하기 위하여 피어에게 의무적으로 네트워크에 머물러야 하는 시간을 부여한 문제점이 있다. CoopNet [2]은 디렉토리 기반 P2P 웹 캐싱 시스템으로서 Pseudoserving과 유사하게 서버의 트래픽 병목 현상을 완화시키는데 초점을 맞추고 있으며, 서버로부터의 리디렉션 메시지 전송 자체의 트래픽까지 고려하여 이를 최소화하려고 하였다. 그러나 CoopNet은 개개 피어들의 로컬 캐쉬 운영은 고려하지 않았다.

Squirrel [3]과 Backslash [4]는 DHT 기반 P2P 웹 캐싱 시스템으로서, Squirrel의 경우 Pastry [5]의 탐색 방법에, Backslash의 경우 CAN [6]의 탐색 방법에 기반하였다. DHT 기반 방식은 오브젝트를 비교적 소수의 유한한 홉 수만에 찾을 수 있다는 장점을 가지지만, 피어의 동적참여가 활발한 경우에는 오버레이 유지 비용이 많이 드는 문제가 있다.

PROOF [7]는 플러딩 기반 P2P 웹 캐싱 시스템으로, 서버로 클라이언트들의 요청들이 한꺼번에 몰릴 경우에도 오브젝트를 효과적으로 전송하는데 초점을 맞추고 있다. 그러나 P2P 네트워크 상에 혼하지 않은 오브젝트를 탐색할 경우에는 네트워크 상으로 전파되어 나가는 룩업 메시지들이 과도한 네트워크 트래픽을 야기하게 된다.

본 논문에서는 피어의 자율적인 동적 참여가 존중되고 오브젝트 탐색 비용이 과도하지 않은 실행 가능한 환경에서 효과적인 P2P 웹 캐싱의 제안에 초점을 맞추고 있다. 이러한 차별화된 목표 환경에서는 기존의 DHT 기반 방식은 피어의 동적 참여로 인한 P2P 오버레이의 유지 비용이 아주 크기 때문

에 적합하지 않다. 기존의 플러딩 기반 방식 또한 레어 오브젝트를 탐색할 시 과도한 룩업 비용이 필요할 수 있다는 점에서 목표 환경에 적합하지 않다고 할 수 있다. 따라서, 제안하는 시스템은 디렉토리 기반 P2P 웹 캐싱에 기반하고 있다.

3. 제안하는 P2P 웹 캐싱 시스템

3.1. 기본 오퍼레이션

OLP는 웹 서버와 웹 서버를 방문하는 클라이언트들로 P2P 네트워크를 구성한다. 클라이언트는 수퍼피어인 웹 서버를 떠나 P2P 네트워크에서 탈퇴하기 전까지 피어로서 P2P 네트워크에 존재하게 된다. 클라이언트가 서버로 특정 오브젝트를 요청했을 때 OLP의 기본 동작 과정을 정리하면 다음과 같다.

- i. 만약 오브젝트 요청이 클라이언트의 로컬 캐쉬에서 히트(hit)한다면, 모든 과정을 종료한다.
- ii. 서버는 리디렉션 디렉토리를 참조해 최근에 오브젝트를 이미 전송받아간 N 명의 이웃을 선택하여, 이들의 주소 목록을 포함하는 리디렉션 메시지를 클라이언트에게 보낸다. 만약 리디렉션 디렉토리에서 오브젝트의 이웃을 찾을 수 없다면, 서버는 오브젝트를 클라이언트에게 직접 전송하고 리디렉션 디렉토리를 갱신 한 후 모든 과정을 종료한다.
- iii. 클라이언트는 서버로부터의 리디렉션 응답을 받은 후, 이웃 목록의 각 이웃에게 오브젝트를 요청한다.
- iv. 만약 오브젝트 요청이 특정 이웃의 로컬 캐쉬에서 히트한다면, 오브젝트는 해당 이웃으로부터 클라이언트로 제공된다. N 명의 이웃 모두의 로컬 캐쉬에서 미스(miss)인 경우, 서버는 오브젝트를 클라이언트에게 직접 전송하고 리디렉션 디렉토리를 갱신 한 후 모든 과정을 종료한다.
- v. 오브젝트가 이웃으로부터 전송 완료되고 사용된 후, 클라이언트의 로컬 에이전트는 오브젝트를 자신의 로컬 캐쉬에 저장할지 여부를 **저장 공간 관리 단계**를 통해 결정한다.
- vi. 클라이언트는 서버로 갱신 메시지를 보낸다. 서버는 필요하다면 리디렉션 디렉토리를 갱신한다.

3.2 주요 인자

3.2.1 P2P 네트워크 환경에서의 오브젝트 라이프타임

일반적인 웹 캐싱에서의 오브젝트 라이프타임 개념과는 달리, P2P 웹 캐싱에서의 오브젝트 라이프타임의 경우에는 피어 자체도 P2P 네트워크를 동적으로 참여/탈퇴하기 때문에, 피어의 로컬 캐쉬 내 오브젝트의 라이프타임 뿐만 아니라 해당 오브젝트를 소유하고 있는 피어 자체의 라이프타임도 고려하여야 한다. 따라서, 우리는 P2P 네트워크에서 오브젝트 라이프타임을 정의하고 이를 OLP 프레임워크에 활용하였다.

LT_A 를 피어 A 에 저장되어 있는 오브젝트의 라이

프타임이라고 하면, LT_A 는 식 (1)과 같이 피어 A 의 평균 활동 기간과 피어 A 의 평균 오브젝트 replacement 시간 중 최소값으로 예측될 수 있다. 식 (1)에서 p_A 는 피어 A 의 총 탈퇴 횟수이고, 피어 A 의 i 번째 참여 시각은 T_j^i 이며 i 번째 탈퇴 시각은 T_i^i 이다. q_A 는 피어 A 의 로컬 캐쉬에서 오브젝트가 삭제된 총 횟수이며, 피어 A 가 자신의 로컬 캐쉬에 저장한 k 번째 오브젝트의 저장 시각은 T_s^k 이고 삭제 시각은 T_r^k 이다.

$$LT_A = \min \left(\frac{\sum_{i=1}^{p_A} (T_i^i - T_j^i)}{p_A}, \frac{\sum_{k=1}^{q_A} (T_r^k - T_s^k)}{q_A} \right) \quad (1)$$

그러므로, 우리는 $T_j^{p_A+1}$ 이후 - 피어 A 가 $p_A + 1$ 번째 참여한 시간 이후 - 의 시간 t 에 피어 A 의 로컬 캐쉬에 저장되는 오브젝트가 P2P 네트워크 상에서 소멸되는 시각을 예측하였다. DT_A^t 를 시간 t 에 피어 A 에 저장된 오브젝트의 소멸 시간이라고 하면, DT_A^t 는 식 (2)와 같이 예측될 수 있다.

$$DT_A^t = \min \left(T_j^{p_A+1} + \frac{\sum_{i=1}^{p_A} (T_i^i - T_j^i)}{p_A}, t + \frac{\sum_{k=1}^{q_A} (T_r^k - T_s^k)}{q_A} \right) \quad (2)$$

3.2.2 오브젝트 요청 빈도

오브젝트 별 요청 빈도를 통하여 해당 오브젝트가 다음 번에 요청될 시간을 예상할 수 있는데, 현재 시간 t' 에 웹 문서 x 에 대한 다음 요청 예상 시간 $RT_x^{t'}$ 는 다음 식 (3)과 같다.

$$RT_x^{t'} = t' + \frac{1}{\text{the request rate of } x} \quad (3)$$

3.3 저장 공간 관리(storage management)

효율적인 저장 공간 관리를 통한 캐싱의 성능 향상을 위하여, 피어가 직접 요청한 오브젝트에 대해, 이를 이웃으로부터 전송받았을 경우, 저장 공간 관리 단계에서 오브젝트를 피어 자신의 로컬 캐쉬에 저장할지 여부를 결정하게 된다. 피어의 DT 가 이웃의 DT 와 오브젝트의 RT 보다 더 나중이라면 가치있는 - 저장될 경우, 충분한 시간동안 존재하여 타 피어로부터 요청받아 제공되어질 가능성이 높은 - 오브젝트로 간주하여 해당 오브젝트는 저장되게 된다. 그림 2는 피어 A 가 오브젝트 x 를 시간 t' 에 요청하여 시간 t 에 전송완료 받았을 때 pseudo 코드이다.

4. 실험 및 성능 평가

4.1 실험 환경

실험을 수행하기위해 사용한 웹 로그는 98년 월드컵 데이터셋으로, 클라이언트들의 월드컵 웹 사이트로의 1,352,804,107 개 요청 정보들로 이루어져 있다. 본 실험은 로그 파일 중 임의의 10일의 데이터를 선택하고 각각에 대해 임의의 300명의 클라이언트를 선정 후 이들의 웹 문서 요청을 로그 파일에 따라 모사하여 수행하였다. 클라이언트들은 각자 자신의 로컬 캐쉬 공간을 가지며, 결과 값은 각 경우를 평균 내어 계산한 값이다.

네트워크 모델은 서버는 T1, 클라이언트들은 [8]에 따라 14.4Kbps부터 T3까지의 대역폭을 가진다. 기본적으로 각 피어는 동시에 한 개의 다른 피어에게만 오브젝트를 제공할 수 있다. 클라이언트 로컬 캐쉬의 replacement 전략은 LRU (Least Recently Used)이다. 피어의 동적인 참여를 모사하기 위해 피어의 최근 요청 후 활동 공백 임계값 δ 초만큼 후에도 다음 요청을 하지 않으면 해당 피어는 P2P 네트워크에서 탈퇴한 것이며, 그 다음 요청이 일어나는 순간 다시 P2P 네트워크에 참여하는 것으로 하였다. 임계값 δ 는 60초부터 600초까지 60초 간격으로 변경해 가며 모든 실험을 수행하였다.

성능 비교를 위해서, 일반 디렉토리 기반 P2P 웹 캐싱, 즉 저장 공간 관리 단계 시에 받은 오브젝트를 항상 로컬 캐쉬 공간 내에 저장하는 시스템 (GCD)과, 제안한 오브젝트 라이프타임 기반 저장 공간 관리를 통해 GCD를 개선한 시스템(OLP_SM)을 모사하여 그 성능을 비교 분석하였다.

```

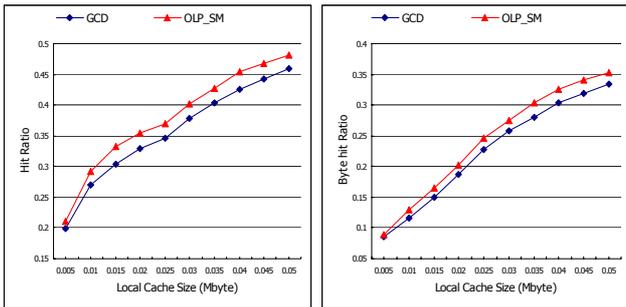
IF (A receives x from the server)
    x is stored in the local cache of A;
ELSE IF (A receives x from B who is one of the neighbors
in the neighbor list)
    IF (there is enough space for x in the local cache)
        x is stored in the local cache of A;
    ELSE
        A calculates  $DT_A^{t'}$ ;
        A reads  $DT_B^{t''}$  from the neighbor list;
        //  $t''$  is the time when the x was stored in B
        A reads the request rate for x from the neighbor list
        and calculates  $RT_x^{t'}$ ;
    IF ( $DT_A^{t'} \geq DT_B^{t''}$  AND  $DT_A^{t'} \geq RT_x^{t'}$ )
        x is stored in the local cache of A;
    ELSE
        After shown to A, x is not stored in the local cache
of A;
  
```

(그림 1) 저장 공간 관리 단계

4.2 성능 평가 기준 (Performance Metrics)

캐싱의 성능을 평가하기 위해 사용한 기준은 *hit ratio* 와 *byte hit ratio* 의 두 가지이다. *hit ratio* 는 전체 요청 회수에서 로컬 캐쉬에서의 탐색 성공이거나 이웃 피어 캐쉬에서의 탐색 성공인 회수가 차지하는 비율이며, *byte hit ratio* 는 전체 요청의 대상 오브젝트들의 총 사이즈에서 로컬 캐쉬에서의 탐색 성공이거나 이웃 피어 캐쉬에서의 탐색 성공인 오브젝트들의 사이즈가 차지하는 비율이다.

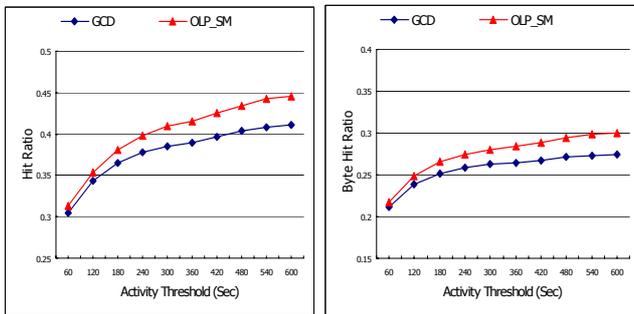
4.3 실험 결과 및 분석



(a) Hit ratio

(b) Byte hit ratio

(그림 2) 캐쉬 사이즈에 따른 캐싱 정확도 비교



(a) Hit ratio

(b) Byte hit ratio

(그림 3) 활동 공백 임계값에 따른 캐싱 정확도 비교

캐싱의 정확도를 로컬 캐쉬의 크기를 변경해가며 살펴보면 그림 2와 같다. 각 측정값은 활동 공백 임계값 δ 을 달리하며 계산한 성능의 평균값이다. 그림 2에서 각 시스템은 로컬 캐쉬의 크기가 커질수록 캐싱 성능이 향상되며, OLP_SM은 향상된 저장 공간 관리를 통해 GCD에 비해 일관적으로 좋은 성능을 보인다. 그림 3은 로컬 캐쉬의 크기를 고정시킨 상태에서 시스템들의 캐싱 성능을 임계값 δ 각각에 대하여 평가하여 비교한 것이다. OLP_SM이 피어의 다양한 동적 참여 환경에 상관없이 일관적으로 GCD에 비해 좋은 성능을 보인다. 그러므로

OLP_SM이 피어가 동적으로 참여/탈퇴하는 P2P 네트워크에 GCD에 비해 더 적합하다고 할 수 있다.

5. 결론

본 연구에서는 클라이언트들의 다양한 참여/탈퇴 패턴을 보장한 실행 가능성이 높은 환경에서 효율적인 디렉토리 기반 P2P 웹 캐싱 시스템을 제안하였다. 제안한 OLP 시스템은 P2P 네트워크의 오브젝트 라이프타임을 활용하여 저장 공간 관리 방안을 제안하였다. 실험을 통해 목표 환경에서 오브젝트 라이프타임 기반 접근 방식이 효율적이며 기존의 디렉토리 기반 P2P 웹 캐싱 시스템에 비해 정확도가 높음을 확인할 수 있었다. 향후에는 오브젝트 라이프타임을 활용한 접근 방식을 플러딩 기반 탐색 모델 등으로 확장해 적용하여 연구할 계획이다.

참고문헌

- [1] K. Kong and D. Ghosal, "Mitigating Server-Side Congestion in the Internet Through Pseudoserving," *Transactions on Networking*, Vol. 7, No. 4, pp. 530-544, 1999.
- [2] V. N. Padmanabhan and K. Sripanidkulchai, "The Case for Cooperative Networking," *IPTPS*, pp. 178-190, 2002.
- [3] 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.
- [4] T. Stading, P. Maniatis, and M. Baker, "Peer-to-Peer Caching Schemes to Address Flash Crowds," *IPTPS*, pp. 203-213, 2002.
- [5] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A Scalable Content-Addressable Network," *ACM SIGCOMM*, pp. 161-172, 2001.
- [6] W. Ng, B. Ooi, and K. Tan, "BestPeer: A Self-Configurable Peer-to-Peer System," *ICDE*, 2002.
- [7] 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.
- [8] S. Saroiu, P. K. Gummadi, and S. D. Gribble, "A Measurement Study of Peer-to-Peer File Sharing Systems," *Multimedia Computing and Networking*, pp. 407-418, 2002.