네트워크 응답시간을 고려한 웹 캐시 교체 정책

신은희[O] 서진모 신승훈 박승규
아주대학교 정보통신전문대학원
{blueseh[O], seo007, sihnsh, sparky }@ajou.ac.kr

A Web Cache Replacement Policy in Consideration of Netwok Response Time

Eunhee Shin[O] Jinmo Seo Seunghun Sihn Seungkyu Park
Graduate School of Information and Communication, Ajou University

요 약

최근 인터넷 기반 서비스 환경의 질적개선과 이에 따른 이용자 증가에 따라, 오디오나 동영상과 같은 규모가 크고 동적 특성을 가진 웹 컨텐츠의 수가 증가하는 추세이다. 이에 따라 'hit ratio' 뿐만 아니라 오브젝트의 크기를 기반으로 하는 'byte hit ratio' 또한 캐시 성능을 평가하는 중요한 측정 요소가 되었다. 이러한 측정기준을 대상으로 할 때, 기존의 웹 캐시 교체 정책중 HP연구소의 GDSF (Greedy-Dual-Size with Frequency)알고리즘과 LFU-DA(LFU with Dynamic Aging)알고리즘이 우수한 성능을 보이는 것으로 알려져있는데, 이러한 기존의 웹 캐시 교체 정책은 서버와의 네트워크 상태를 고려하지 않은 정책이고, 이에 따라 네트워크 상태에 따른 전송비용의 차이를 반영하지 못하고 있다. 따라서 본 논문에서는 서버와 웹 캐시 간의 네트워크 상태를 반영할 수 있는 캐시 교체 정책을 제안하고, 이에 대한 실험을 수행하였으며, 그 결과 사용자의 요구에 대한 응답시간의 감소 효과를 얻을 수 있었다.

1. 서론

최근 초고속 인터넷의 성장으로 인하여 개인 사용자간, 혹은 기업간에 전송되는 데이터중 멀티미디어 데이터가 차지하는 비율이 더욱 증가하고 있으며, 특히 기업체는 사내 응용 프로그램을 인트라넷 환경을 기반으로 한 브라우징 프로그램으로 대체하고 있다. 이러한 추세는 네트워크 하부 구조에서의 트래픽 부하 가중 및 웹 접근 속도저하현상 등의 현상을 초래하였다. 이러한 문제를 해결하기 위해 등장한 것이 웹 캐싱 시스템이다.

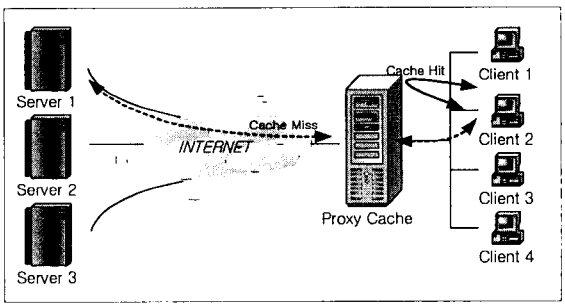

그림 1 웹 캐싱 시스템 구조

웹 캐시는 요청 빈도가 높은 웹 컨텐츠를 사용자에 가까운 로컬 캐시 서버에 저장하고 이를 사용자에게 서비스함을 통해, 사용자가 부하가 집중되고, 전송비용이 높은 원격지의 웹 서버에 접속하여 데이터를 가져오는 확률을 낮춤으로써, 서버의 부하와 사용자에 대한 응답시간을 줄이고, 서버-사용자간 네트워크의 효율성을 증가시키기 위한 목적으로 이용된다. 일반적으로 웹 캐싱 시스템의 성능은 캐시 교체 정책이 보이는 'hit ratio'를 주 대상으로 한다. 그러나 인터넷 환경의 질적 개선에 따른 오디오 및 동영상과 같은 규모가 큰 동적 웹 컨텐츠의 증가에 따라 기존의 웹 캐시 교체 정책의 주요 목표였던 hit ratio의 재고 뿐만 아니라 오브젝트의 크기을 기반으로 캐시 교체 정책이 가지는 이득을 측정하는 'byte hit ratio'도 중요한 측정요소가 되었다.[2]

여기에 추가적으로 오브젝트의 크기는 네트워크의 전송 비용에 큰 영향을 미치므로, 서버와 웹 캐시간 네트워크의 상태는 반드시 고려해야 하는 중요한 요소라고 할 수 있다. 따라서 본 논문에서는 기존의 주요 캐시 교체 정책인 GDSF(Greedy-Dual-Size with Frequency) 알고리즘과 LFU-DA(LFU with Dynamic Aging)알고리즘을 변형하여, 네트워크의 상태를 파악하여 평균 응답시간을 최소화 할 수 있도록 하는 GNA(GDSF with Network Awareness)알고리즘과 LNA(LFU-DA with Network Awareness)알고리즘을 제안하고, 이에 대한 평가를 수행한다.

2. 웹 캐싱 시스템의 기능

웹 캐시는 서버와 사용자 사이에 위치하여, 사용자의 요청을 받아 서버에 전달하고, 서버로부터 받은 데이터를 해당 사용자에게 전달하는 중계 역할을 수행한다. 즉, 그림 2에 제시된 것처럼 사용자의 요청이 발생하면, 해당 HTTP 오브젝트가 캐시 내에 존재하는지 확인하고, 저장되어 있는 경우라면 서버에 접속하지 않고 웹

본 연구는 정보통신부지원 대학기초연구지원사업 (정보통신기초기술연구지원사업)의 지원을 받아 진행됨

캐시가 해당 오브젝트를 바로 클라이언트에게 제공한다. 반대로 캐시 내에 저장되어 있지 않는 경우에는, 해당 HTTP 오브젝트를 서버에 요청하고, 서버로부터 전달받은 오브젝트를 캐시에 저장한 후, 클라이언트에게 제공한다. 만약 캐시에 해당 HTTP 오브젝트를 저장할 여유 공간이 없다면, 캐시 교체 알고리즘에 따라 삭제할 오브젝트를 결정하여 이를 삭제하고, 새로운 오브젝트를 저장된다.

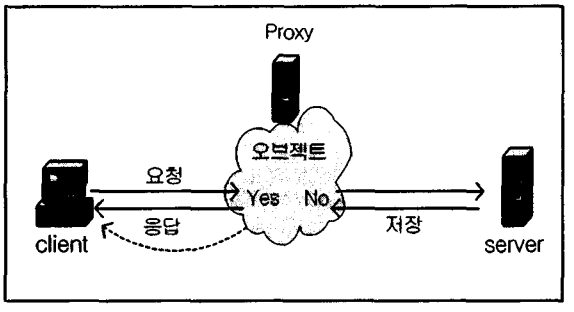

그림 2 웹 캐싱 시스템의 기능

3. 관련연구

3.1 Greedy-Dual 정책

Greedy-Dual 알고리즘은 캐시 또는 메모리에서 동일한 크기를 갖는 오브젝트들에 대해 다른 저장 장소로부터 오브젝트를 반입하는 비용이 각기 다른 경우를 적용하기위해 소개되었다. 캐시된 모든 오브젝트들의 우선순위 값을 갖고 있으며, 이 우선순위 값이 가장 작은 오브젝트를 교체하는 방법이다. 우선순위 값은 오브젝트를 캐시로 가져올 때의 비용으로 정의하고 캐시로 가져오는 비용이 적을수록 더 낮은 우선순위를 갖게 된다.[5]

3.2 GD-Size

Greedy-Dual 알고리즘을 더 보완하기 위해 우선순위 값을 *cost/size*로 재정의하였다. 이 알고리즘에서 비용함수는 캐시의 목적에 따라 선택적으로 다르게 선택할 수 있도록 하였다. 이 정책에서 캐시 미스율을 최소화하기위한 GD-Size(1), 캐시 미스로 인한 네트워크의 트래픽을 최소화 하기위한 GD-Size(packets)가 만들어졌다.[5]

3.3 GDSF & LFUDA

HP(Hewlett Packard)연구소는 GD-Size 정책에 오브젝트에 대한 접근 빈도를 추가로 반영한 GDSF (Greedy Dual-Size with Frequency)정책과 LFU에 Aging요소를 반영한 LFU-DA(Least Frequently Used with Dynamic Aging)정책을 제안하였다. GDSF정책은 GD-Size(1)알고리즘을 확장한 것이며, 최고의 hit ratio를 보이며, LFU-DA는 byte hit ratio값이 높은 LFU알고리즘을 확장한 것이다.[3][4]

3.5 기존 정책의 문제점

HP연구소에서 제안한 GDSF와 LFU-DA정책은 기존에 제안되었던 알고리즘들의 주요 요소들의 장점을 취합하여 완성되었다고 할 수 있다.

하지만 GDSF와 LFU-DA정책은 웹 캐시-서버간 네트워크 상태의 다양성 및 동일 서버 내에서 시간 흐름에 따른 네트워크의 상태의 가변성 등, 네트워크의 상태의 불균일성에서 오는 영향을 고려하지 않았기 때문에 시간과 공간에 따라 사용자의 요청 환경이 변화하는 상황에 적용하는데 어려움이 있으며, 사용자에 대한 응답 시간 또한 상당한 편차를 보인다. 또한 캐시된 오브젝트에 부여된 우선 순위 값도 해당 오브젝트가 삭제되기 이전까지 동일한 값으로 유지되므로, 현재의 정확한 네트워크 상태를 반영한 값이라 볼 수 없다.

따라서 본 논문에서는 기존 교체 정책이 가지는 제한성의 축소를 위해, 네크워크 환경에 대한 고려를 추가한 캐시 교체 정책을 제안한다.

4. GNA정책 및 LNA정책

기존 GDSF정책과 LFU-DA정책이 캐싱된 오브젝트에 우선순위를 부여하는 방법은 아래의 수식과 같다.

$$Pr(f) = Clock + Fr(f) \times \frac{Cost(f)}{Size(f)} \quad \text{--- ①}$$

$$Pr(f) = Clock + Fr(f) \times Cost(f) \quad \text{--- ②}$$

위의 수식 ①, ②에서 $Size(f)$는 오브젝트 f의 크기(byte)이고, $Cost(f)$는 오브젝트 f를 웹 서버로부터 캐시로 가져오는 비용을 나타낸다. $Cost(f)$의 경우, hit ratio를 높이고자할 때는 1로, byte hit ratio를 높이고자할 때는 $2 + size/536$으로 정의하여 사용하며, 캐시 미스 처리 시에 요구되는 전송 패킷 수를 예측하는 함수이다. 이는 TCP의 한 세그먼트의 크기를 536바이트로 전제하고, 요청과 응답에 대해 각각 하나의 패킷이 필요하게 된다는 가정을 바탕으로 도입된 요소이다. 따라서 식 ①과 ②를 통해 GDSF와 LFU-DA는 서버로부터 전송되는 오브젝트의 사이즈를 고려한 정책임을 알 수 있다.

그러나, 네트워크 상태의 편차를 고려할 때, 서버로부터 오브젝트를 전달받는데 소요되는 지연 시간은 오브젝트의 사이즈가 캐쉬 교체 정책에 미치는 영향보다 크다고 할 수 있다.[1] 따라서, 네트워크의 상태를 고려하여 서버와 웹 캐시 사이의 HTTP 연결시간(CSTime)을 캐시 교체정책에 반영하여 아래와 같은 교체정책을 제안한다.

$$Pr(f) = Clock + Fr(f) \times \frac{CSTime}{Size(f)}$$

$$Pr(f) = Clock + Fr(f) \times CSTime$$

5. 시뮬레이션

5.1 시뮬레이션 환경

시뮬레이션에서는 동영상과 같은 규모가 큰 오브젝트에 대한 요청이 전체 요청의 20%라고 가정하였다. 웹 캐시와 각 서버의 HTTP 연결시간은 표1과 같이 최소, 최대 범위를 두고 시간에 따라 변하는 값을 갖게 하였으며, 실험의 편이성을 위해 프락시와 사용자간의 연결시간은 10*msec*로 동일하다고 가정하였다.

server	min	max	ave
A	132	808	470
B	235	1019	627
C	456	1432	944
D	773	1967	1370
E	908	2374	1641
F	1495	3624	2560
G	3442	6459	4951
H	4188	7976	6082
I	6586	8965	7776
J	8910	9970	9440

표 1 웹 서버별 http연결시간 (단위:*msec*)

5.2 시뮬레이션 결과

그림3과 그림4는 전체 요청의 20%가 미디어 스트림 오브젝트에 대한 요청이고, 전체 데이터의 사이즈에 대한 캐쉬 가능 사이즈 비율을 다르게 한 경우의, Hit Ratio 및 누적응답시간에 대한 시뮬레이션 결과를 나타내고 있다.

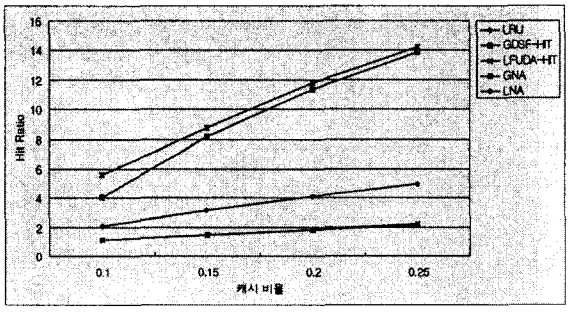

그림 3 Hit Ratio

GNA와 LNA정책은 우선순위에 네트워크 상태를 반영했고, 미디어 스트림에 대한 요구가 20%에 이른다고 가정했기 때문에 hit ratio가 기존의 정책에 비해 낮은 성능을 보이며[그림 4], 총 응답시간의 경우 기존의 GDSF와 LFU-DA정책보다 GNA와 LNA가 응답시간이 상당 부분 감소했음을 나타내고 있다.[그림 5] Hit ratio가 기존 캐쉬 교체 정책에 비해 낮음에도 불구하고, 누적응답시간이 감소한 것은 네트워크 응답시간을 고려함에 따라 캐쉬가 직접 서비스하는 오브젝트의 수는 줄었으나, 사용자에 대한 응답시간은 줄어들어 서비스에대한 품질이 향상되었음을 나타낸다.

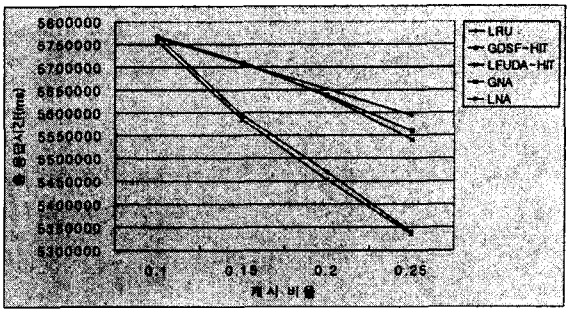

그림 4 누적응답시간

한편 별도로 수행한 실험에서는 미디어 스트림에 대한 요청이 0%인 경우 응답시간은 20%인 경우보다 현저한 감소를 보인다. 따라서 제안한 정책은 미디어 스트림에 대한 요구가 적은 경우 더욱 효율적인 결과를 보인다고 할 수 있다.

6. 결론

인터넷의 질적인 향상에 따른 대규모 미디어 오브젝트의 수가 급증하는 환경에서, 네트워크 전송비용은 반드시 고려해야 하는 중요한 요소이다.

본 논문에서는 오브젝트를 웹 서버로부터 캐시로 가져올 때, 오브젝트의 사이즈에 근거를 둔 비용측정 대신 네트워크의 상태에 근거하여 서버와 웹 캐시 사이의 HTTP 연결시간(CSTime)을 반영한 정책을 제안하였으며, 그 결과 사용자에 대한 응답시간의 감소로 서비스 품질의 향상을 보였다.

7. 참고문헌

[1] 서진모, "웹 프락시에서 네트워크 상태를 고려하는 캐시교체정책", 공학 석사 논문, 아주대학교, 2002

[2] M. Arlitt, L. Cherkasova, J. Dilley, R. Friedrich, and T. Jin, "Evaluating Content Management Techniques for Web Proxy Caches", HPL-98-173, April, 1999.

[3] L. Cherkasova. "Improving www Proxies Performance with Greedy-Dual-Size-Frequency Caching policy", Technical Report HPL-1998-69R1, HP Laboratories. November, 1998

[4] J. Dilley, M. Arlitt and S. Perret, "Enhancement and Validation of Squid's Cache Replacement Policy" Technical Report HPL-1999-69, Laboratories, May, 1999.

[5] P. Lorensetti, L. Rizzo and L. Vicisano. "Replacement Policies for Proxy Cache", Manuscript, 1997.