

이동 컴퓨팅 환경에서 요구 패턴 분석에 기반한 캐쉬 대체 전략

이윤장^o 신동천 김도일
 중앙대학교 정보시스템학과

willy@ms.cau.ac.kr, dcshin@cau.ac.kr, dikim@cau.ac.kr

A Caching Strategy based on Analysis of Access Pattern in Mobile Computing Environments

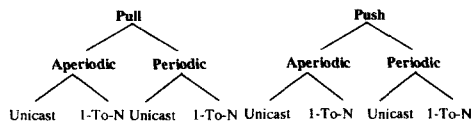
Yoon-Jang Lee^o Dong-Cheon Shin Do-il Kim
 Dept. of Information System, Management Chung-Ang University

요 약

이동 컴퓨팅 환경에서 캐쉬 대체 전략은 시스템의 성능에 많은 영향을 준다. 과거의 캐쉬 대체 전략은 주로 푸쉬 환경에서 방송 디스크(Broadcast Disk) 기법을 사용하는 스케줄링 기법을 기반으로 제안되었다. 따라서 다양한 사용자 요구를 반영하는 풀 환경에서는 좋은 성능을 보여주기 어렵다. 본 논문에서는 사용자의 요구 패턴 분석을 기반으로 풀 기반에 적용할 수 있는 효율적인 캐쉬 대체 전략을 제안한다. 제안한 대체 전략은 히트율 뿐만 아니라 미스 비용도 고려한 전략이다.

1. 서론

지난 몇 년 동안 인터넷의 급속한 성장은 컴퓨팅 환경 뿐만 아니라 통신 환경까지도 급속도로 변화시켰다. 모든 응용 프로그램은 인터넷으로 통합되고 있으며, 인터넷을 통한 클라이언트/서버 환경은 유선 환경을 넘어 무선 환경까지 적용 범위가 확대되고 있다. 이러한 통신망의 성장은 데이터 분산 기반 시스템(data dissemination-based system)과 같은 다수의 클라이언트 요구를 수용할 수 있도록 하기 위한 시스템에 대한 연구를 촉진시키고 있다. 이러한 분산 기반 시스템의 특징은 대규모화(tremendous scale), 데이터 요구의 높은 집중도, 통신환경의 비대칭(asymmetry) 등을 들 수 있다. [그림 1]은 시스템의 환경에 따라 취할 수 있는 전송 기법을 보여주며, 크게 푸쉬 기반과 풀 기반으로 나눌 수 있다[2].



[그림 1] 데이터 전송 기법

푸쉬 기법은 클라이언트의 요구 없이 데이터를 보내는 방식으로 대역의 낭비 없이 보낼 수 있는 장점이 있다.

풀 기법은 클라이언트의 직접적인 요청이나, 대기 시간, 데드라인 등을 이용하여 보내는 방식으로 대역의 낭비는 있으나, 사용자의 요구를 정확히 수용할 수 있는 장점을 가진다[6]. 유선 환경에서의 분산 기반 시스템에 관한 연구는 초기에 대역의 제약으로 푸쉬 기반 응용이 대부분이었으나, 네트워크 인프라와 통신기술의 발전으로 NOD(News-On-Demand), VOD(Video-On-Demand), Interactive TV와 같은 풀 기반 응용에 관한 연구가 진행되고 있다.

대체할 페이지(victim) 선정 기준¹⁾ 중에 어떠한 것에 중점을 두느냐에 따라 캐쉬 대체 전략을 푸쉬나 풀 기반에 적용할 수 있는지가 결정된다. 히트율에 중점을 둔 캐쉬 대체 전략은 풀 기반에 적합한 방법이다. 페이지가 캐쉬에 없을 경우 직접 페이지를 요청하여 짧은 시간 내에 가져올 수 있기 때문에 미스 비용에 대한 고려는 필요가 없다[1, 4]. 반면에, 미스 비용을 고려한 캐쉬 대체 전략은 푸쉬 기반에 적합한 방법이다. 페이지가 캐쉬에 없을 경우 서버에 요청하여 받는 것이 아니라, 해당 페이지를 서버가 보내줄 때까지 기다려야 하기 때문

1) · 히트율(Hit Ratio) : 요구하는 페이지가 캐쉬에 있을 확률
 · 미스비용(Miss Cost) : 요구하는 페이지가 캐쉬에 없을 경우, 페이지를 사용하기 위해서 필요한 비용 - 통신비용, 대기시간 등.

에 페이지에 따라 대기시간이 길어질 수 있으므로, 미스 비용의 고려가 필요하다[5, 7, 8].

다수의 클라이언트를 가진 이동 컴퓨팅 환경에서는 대역의 제약을 극복하기 위해 푸쉬 기법을 활용한 연구가 진행되어 왔다[5, 6]. 특히 [5]에서 제안한 방송 디스크 기법은 주기적이며 1-to-n의 특징으로 미스 비용의 계산이 용이하여, 캐쉬 대체 전략과 프리페칭(prefetching) 전략에 자주 이용된다[7].

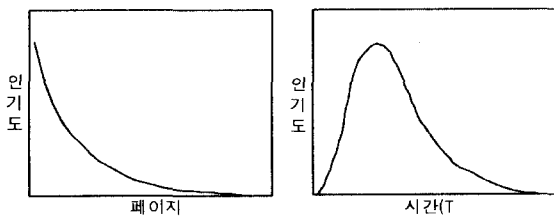
본 연구에서 제안하는 캐쉬 대체 전략은 사용자 요구 패턴 분석에 기반하여 대체할 페이지를 선택한다. 따라서, 요구패턴 변화가 심한 풀 환경에 보다 적합하고, 클라이언트 전체 응답 시간을 개선할 수 있다.

2. 요구패턴분석

이동 컴퓨팅 환경에 적합한 기존의 캐쉬 대체 전략은 방송 디스크 기법을 이용하여 푸쉬 기반에 적합하도록 만들어졌다. 풀 기반의 시스템에서는 사용자의 페이지에 대한 응답 시간을 최소화하기 위해서 사용자의 요구 패턴에 기초한 캐쉬 대체 전략이 필요하다. 사용자의 요구 패턴을 분석하기 위해 인기도(popularity)와 대기시간(waiting time) 등을 고려할 수 있다.

2.1 인기도(Popularity)

인기도란 사용자의 관심에 따라 특정 페이지가 접근되는 빈도를 수치화 한 값이다. 인기도는 크게 단기(short-term) 인기도와 장기(long-term) 인기도로 나눌 수 있다. 단기 인기도는 시간의 흐름에 따른 클라이언트의 관심 변화를 배제하고, 특정 시점에 전체 페이지에서 특정 페이지가 얼마나 요구되는지를 확률로 표현하며, Zipf분포(Zipf Distribution)[3]와 마찬가지로 특정 페이지에 요구가 집중된다([그림 2] 참고). 반면에, 장기 인기도는 특정 페이지에 대한 사용자의 시간에 따른 관심의 변화 정도를 확률로 표현한다([그림 3] 참고).



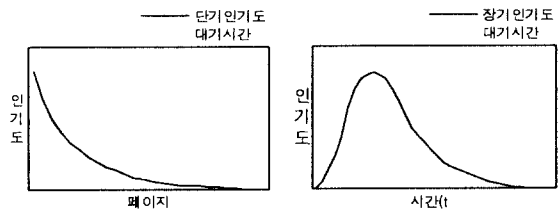
[그림 2] 단기 인기도 패턴 [그림 3] 장기 인기도 패턴

인기도의 유지는 사용자의 요구 패턴의 변화에 따라 인기도가 [그림 2]와 [그림 3]의 패턴에 따르도록 하는

것이다. 처음 페이지가 요청되면, 인기도는 1로 초기화된다. 페이지가 요청될 때마다 인기도를 1씩 증가시켜 자주 요청되는 페이지와 그렇지 못한 페이지간의 인기도 차이를 반영할 수 있다. 반대로, 시간의 흐름에 따른 클라이언트 관심의 변화를 반영하기 위해 인기도를 감소시킨다. 인기도가 감소하는 것은 해당 페이지에 대한 요청의 빈도가 이전에 비해 줄었음을 나타내므로, 시간 T 간격마다 각 페이지의 요청 여부를 확인하여 요청되지 않은 페이지의 인기도를 α 만큼 줄일 수 있다. α 값과 시간 T의 값을 조절하여 단기 인기도와 장기 인기도를 보다 민감하게 반영할 수 있다. 한편, 구해진 인기도는 각 클라이언트의 페이지 접근 패턴을 보다 정확하게 반영하여 계산된 결과이므로 페이지 히트율을 높이는 데 이용될 수 있다.

2.2 대기 시간(Waiting Time)

대기 시간은 페이지에 대한 요구가 있을 후 실제 서버로부터 받을 때까지 걸린 시간을 의미한다. [그림 4]는 특정 페이지의 단기 인기도 혹은 접근 확률에 따른 대기시간의 변화 패턴을 보여준다. 일반적인 푸쉬 환경에서 서버는 접근 빈도나 인기도가 높은 페이지를 그렇지 않은 페이지보다 자주 보내고, 풀 환경에서도 그것은 마찬가지다. [그림 5]는 특정 페이지에 대해 시간의 변화에 따라 인기도가 높아진다면, 다른 조건이 동일할 때, 대기시간이 감소하는 모습을 보여준다.



[그림 4] 단기 인기도와 대기시간의 상관관계 [그림 5] 장기 인기도와 대기시간의 상관관계

본 연구에서는 [그림 4]와 [그림 5]의 대기시간을 반영하기 위해 식(1)과 같이 가중 평균을 사용한다.

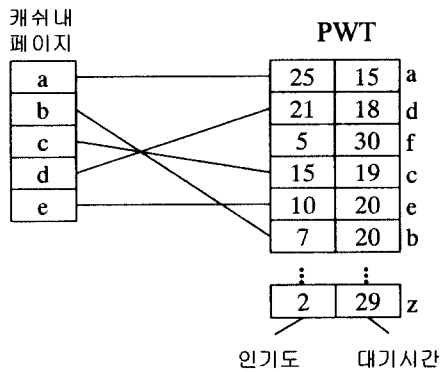
$$A(P_i, W_{x+1}) = \frac{A(P_i, W_x) \times \beta}{100} + \frac{P_i, W_{x+1} \times (100 - \beta)}{100} \dots (1)$$

식 (1)에서 P_i, W_i 는 특정 페이지 i 에 대한 최초 요구시 대기 시간을 의미하고, P_i, W_x 는 x 번째 요구시 대기 시간을 의미한다. $A(P_i, W_{x+1})$ 은 $x+1$ 요구까지의 평균 대기 시간을 의미하며, P_i, W_x 까지의 평균 대기 시간을 β (%) 반영하고, P_i, W_{x+1} 의 대기 시간을 $100 - \beta$ (%) 반영하여 구한다. 이렇게 하므로써, 두 가지 인기도의 변화에 따른

대기 시간의 변화를 반영할 수 있으며, 비율 β 을 조정하여 인기도의 변화에 따른 대기 시간의 변화를 좀 더 민감하게 반영할 수 있게 된다.

3. 캐쉬 대체 전략

이동 컴퓨팅 환경에서 제안된 캐쉬 대체 전략은 미스 비용이 무시될 정도로 작지 않기 때문에 일반적으로 고려된다[5]. 본 연구에서 제안하는 캐쉬 대체 전략에서 고려할 수 있는 대체 기준은 히트율을 높일 수 있는 인기도와 미스 비용을 줄일 수 있는 대기 시간이다. 이 중에서 인기도만을 고려할 경우에는 히트율은 높일 수 있지만, 요청할 가능성이 캐쉬에 있는 페이지보다 높지 않지만, 대기시간이 높은 페이지 때문에 성능 저하가 발생할 수 있다. 대기 시간만을 고려할 경우에는 미스 비용을 줄일 수 있지만, 인기도가 높은 페이지가 계속 미스되어 마찬가지로 성능에 문제가 발생할 수 있다.



[그림 6] PWT를 이용한 캐쉬대체 예제

[그림 6]에서 우측의 PWT 테이블은 PWT 값(PWT=인기도×대기시간)을 기준으로 내림차순으로 정렬된다. 특정 시점까지 요구를 수용한 결과 실제 캐쉬에 있는 페이지보다 PWT 값이 큰 페이지가 생기면, 캐쉬에 있는 가장 낮은 PWT 값을 가지는 페이지와 대체한다. [그림 6]의 예는 페이지 f가 캐쉬 내에 있는 페이지보다 PWT 값이 큰 경우를 보여준다. 이 경우 캐쉬에서 가장 낮은 PWT 값을 가지는 페이지 b가 대체할 페이지로 선정되며, 캐쉬에 있지 않은 페이지 중에서 가장 높은 PWT 값을 가지는 페이지 f가 캐쉬에 들어가게 된다.

4. 결론

이동 컴퓨팅 환경에서 캐쉬 대체 전략은 히트율 뿐만 아니라 미스 비용도 함께 고려하는 것이 사용자의 응답 속도를 향상시키기 위해서 필요하다. 본 논문에서는 사

용자 요구 패턴을 인기도로 수치화 하였다. 특정 페이지의 인기도가 변화하면, 대기 시간은 인기도의 영향을 받는다. 따라서 인기도와 대기 시간을 함께 고려하여, 풀 기반 환경에서 효율적인 캐쉬 대체 전략을 제안하였다.

본 연구는 몇 가지 최적화 문제가 남아 있다. 첫째, PWT 값의 유지와 관련하여 최적화된 PWT 테이블 크기 결정 문제가 있다. 둘째, 인기도 유지와 관련하여 요구 시간 T에 따른 인기도 감소치 α 를 최적화 하는 문제가 있다. 셋째, 대기 시간과 관련하여 대기 시간의 민감한 반응을 위한 β 의 결정 문제가 있다. 이를 위해 현재 시뮬레이션을 준비중이며, 향후 최적화된 PWT 대체 전략과 기존의 캐쉬 대체 전략과의 비교를 통해 성능 검증이 이어질 것이다.

참고 문헌

- [1] A. Tanenbaum, "Modern Operating System", Prentice Hall, Inc., 1992.
- [2] D. Aksoy, M. Altinel, R. Bose, U. Cetintemel, M. Franklin, J. Wang, and S. Zdonik, "Research in Data Broadcast and Dissemination", AMCP 1998, pp. 194 - 207.
- [3] D. Knuth, "The Art of Computer Programming", Vol II, Addison Wesley, 1981.
- [4] E. O'Neil, P. O'Neil, and G. Weikum, "The LRU-K Page Replacement Algorithm for Database Disk Buffering", Proc. of ACM SIGMOD, 1993.
- [5] S. Acharya, "Broadcast Disks:Dissemination-based Data Management for Asymmetric Communication Environments", Ph.D. Thesis, Dept. of Computer Science, Brown Univ., Sep. 1997.
- [6] S. Hameed and N. H. Vaidya, "Efficient Algorithms Scheduling Data Broadcast", ACM/Baltzer Wireless Network Journal, May 1999, pp. 183-193.
- [7] S. Khanna and V. Liberatore, "On Broadcast Disk Paging", Proc. of the 30th ACM Symp. on the Theory of Computing, 1998, pp. 634 - 643.
- [8] V. Liberatore, "Caching and Scheduling for Broadcast Disks Systems", TR 98-71, UMLACS, 1998.