제19회 한국정보처리학회 춘계학술발표대회 논문집 제10권 제1호 (2003.5)

비트 벡터를 기반으로 하는 데이터 방송 시스템을 위한 캐쉬 대체 전략

신동천*
*중앙대학교 정보시스템학과
e-mail:dcshin@cau.ac.kr

A Page Replacement Strategy for Data Broadcast Systems Based on Bit Vector

Dong Cheon, Shin*
*Dept of Information Systems, Chung-Ang University

요 약

다수의 클라이언트에게 효과적으로 데이터를 전달하는 방법중의 하나가 데이터 방송이다. 방송된 데이터에 대한 효과성 정보를 클라이언트로부터 전달받기 위해 비트 벡터를 사용할 수 있다. 본 논문에서는 비트 벡터를 사용하는 방송 알고리즘의 특성을 고려한 효율적인 캐쉬 대체 전략을 제안한다. 제안한 캐쉬 대체 전략은 클라이언트의 요청 회수, 최근 요청 정도, 방송 횟수등을 복합적으로 고려한다.

1. 서론

통신 기술의 급속한 발달은 이동 컴퓨팅 환경을 보편화시키고 있다. 이동 컴퓨팅 환경의 특징은 크게 무선과 이동성으로 나누어 볼 수 있다[1, 4, 7]. 무선과 이동성이라는 특징으로 인하여 기존의 컴퓨팅 환경에서 제안된 여러 가지 기법이나 알고리즘들은 이동 컴퓨팅 환경에서 적용될 수 없거나 낮은 성능을 보이게 된다[5, 6]. 뿐만 아니라, 무선과 이동성이라는 특징은 클라이언트와 서버사이에 비대칭성(asymmetry)을 초래하며 다수의 클라이언트가 셀 안에 존재하게 된다. 따라서, 제한된 대여폭에서 다수의 클라이언트에게 데이터를 전송하는 방법으로 데이터 방송기법이 관심을 끌어 왔다.

데이터 방송 기법이 갖는 가장 큰 장점중의 하나는 클라이언트의 수에 관계없이 데이터 서비스를 할 수 있다는 점이다. 그러나 방송 알고리즘이 클라이언트의 데이터 요구에 대한 변화를 반영하여 방송 프로그램을 생성하지 못하였다[2]. [8]에서는 다중 디스크 방송 기법[2, 3]을 확장하여 클라이언트의 요구 변화를 반영하여 방송하는 혼합형태 기반의 방송 알고리즘을 제안하였다. [8]에서 제안한 방송 알고리즘은 비트 벡터(bit vector)와 클라이언트의 데이터 요청을 토대로 다음 방송 프로그램에 클라이언트의 요구 변화를 반영시키고 있다.

한편, 데이터 방송 환경에서는 낮은 대여폭으로 인하여 클라이언트에 캐쉬를 두어 필요한 데이터가 방송되기를 기다리지 않고 캐쉬를 액세스하여 효율성을 높일 수 있다. 캐쉬를 고려하는 방법에서 해결해야 될 중요한 문제 중 하나는 캐쉬내의 데이터 페이지를 새로운 페이지와 대체시키는 페이지 대체 전략이다.

이동 컴퓨팅 시스템에서 데이터 방송 환경을 고려한 많은 전략들이 제안되었다[3]. 이러한 전략들의 대부분은 데이터 방송 환경에서는 캐쉬의 히트율을 높이는 것뿐만 아니라 캐쉬 미스가 발생하는 경우 서버로부터 요구하는 데이터가 방송될 때까지 기다려야 하므로 미스에 따른 비용까지 고려해야 한다는 관점에서 제안되었다.

그러나, 궁극적으로 캐쉬 전략은 방송 알고리즘과 긴밀한 상호 의존성을 갖는다. 예를 들어, 자주 방송되는 데이터는 상대적으로 드물게 방송되는 데이터보다 캐쉬에 존재할 가치가 떨어지며 캐쉬에 존재하지 않는 데이터는 존재하는 데이터보다 상대적으로 자주 방송해야 대기 시간을 줄일 수 있다. 지금까지 제안된 캐쉬 전략의 대부분은 방송 알고리즘과 독립적으로 제안되었거나 방송 알고리즘의 기본 원칙을 효율적으로 반영하지 못하였다고 할 수 있다.

본 논문에서는 비트 벡터를 기반으로 하는 데이터 방송 시스템 환경에서 방송 알고리즘의 기본 특성을 고려한 새로운 페이지 대체 전략을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 비트 벡터를 기반으로 하는 방송 알고리즘을 간략히 소개한다. 3장에서는 새로운 페이지 대체 전략을 제시하고 4장에서는 결론을 맺는다.

2. 비트 벡터 기반의 방송 알고리즘[8]

이 알고리즘은 [2]에서 제안된 방송 디스크 알고리즘 중에서 혼합방송 형태의 알고리즘을 확장한 것이다. 따라서, 푸쉬 형태로 방송되는 데이터와 풀 형태로 방송되는 데이터 사이에 대여폭을 효율적으로 할당하여 두 형태의 방송 데이터를 모두 지원한다. 즉, 클라이언트의 데이터 요청은 큐에 저장되어 FIFO 형식으로 일반 방송 프로그램의 데이터와 할당된 대여폭 범위 내에서 번갈아 가며 방송된다. 큐가 만원일 때 클라이언트의 요청은 거절되며 이미 큐에 있는 데이터에 대한 요청은 앞서 요청된 데이터 방송으로 만족될 수 있으므로 큐에 저장되지 않는다.

각 방송 주기 초에 방송 스케줄 즉, 데이터 식별자와 방송 주기 버전번호를 먼저 방송한다. 각 클라이언트는 수신한 방송 스케줄 정보를 토대로 필요한 데이터가 캐쉬에 없는 경우 이번 방송 주기에 방송되면 기다리고 그렇지 않으면 데이터 요청을 서버에게 한다.

방송된 데이터에 대한 클라이언트의 액세스 정보를 얻기 위해 각 클라이언트는 비트 벡터[10]를 유지한다. 비트 벡터의 각 비트는 현재 방송 프로그램의 각 데이터 페이지를 나타낸다. 새로운 방송 프로그램을 수신하면 클라이언트는 버전 번호를 갱신하고 비트 벡터를 모두 '0'으로 클리어 한다. 요구한 데이터가 방송되는 데이터에 있으면(air hit) 해당 데이터 비트는 '1'로 변경된다. 캐쉬에 있는 경우나 요청하여 방송된 데이터인 경우에는 비트 벡터에 영향을 주지 않는다. 필요한 데이터가 캐쉬나 방송 프로그램에 없는 경우 클라이언트는 서버에게 데이터 요청을 한다. 이때, 비트 벡터도 함께 피기배킹(piggybacking)된다. 즉, 데이터 요청 메시지는 데이터 식별자, 주기 버전번호, 비트 벡터로 구성된다.

서버는 각 데이터에 대해 MFA(most frequently accessed) 값을 유지한다. 서버는 버전번호를 확인하고 데이터가 클라이언트에 의해 요청되거나 해당 비트 벡터가 '1'인 경우 MFA 값을 '1' 증가시킨다. MFA 값이 높은 데이터들을 방송 디스크 기법에 따라 방송하게 된다. 이렇게 함으로써 클라이언트의 데이터 액세스 정보를 다음 방송 프로그램 생성에 반영할 수 가 있다.

3. 캐쉬 대체 전략

3.1 동기

이동 컴퓨팅 환경에서 캐쉬의 사용은 낮은 대여폭을 극복하여 클라이언트에게 빠른 응답시간을 제공할 수 있는 보편적인 방법중의 하나이다. 캐쉬 사용에서 빠른 응답을 제공하기 위해서는 효율적인 캐쉬 대체 전략이 필요하다.

전통적인 시스템에서 대부분의 캐쉬 대체 전략은 히트율을 향상시키기 위한 전략들이었다. 그러나 데이터 방송 환경에서는 캐쉬 미스가 발생할 경우 전통적인 시스템 환경과는 달리 상대적으로 낮은 대여폭을 통하여 필요한 데이터를 서버로부터 제공받기 때문에 방송 알고리즘에 따라 미스에 따른 비용이 커지며 아울러 그 비용을 예측하기 어렵게 된다. 따라서, 히트율을 높이는 전략뿐만 아니라 미스 비용을 줄일 수 있는 전략이 되어야 한다. 방송 프로그램의 내용을 완전히 아는 경우에는 미스 비용을 최소화할 수 있음은 당연한 사실이다. 그러나 동적인 방송 환경에서는 방송 주기마다 방송 프로그램이 다를 수 있으므로 방송 프로그램을 생성하는 알고리즘의 기본 원칙을 고려하여 대체 전략을 수립하여야 미스

비용을 가능한 최소화 할 수 있을 것이다.

[8]에서는 동적인 방송 알고리즘의 기본 원칙을 고려하지 않고 전통적인 시스템의 대체전략인 LRU-K[9]를 사용하고 있다. 그러나 이는 무엇보다도 비트 벡터를 이용하여 클라이언트의 액세스 패턴을 토대로 방송 프로그램을 생성하는 방송 알고리즘의 기본적인 특성을 고려하지 않은 것이라 할 수 있다. 예를 들어, [8]에서 제안된 방송 알고리즘에 따르면 캐쉬에 있는 데이터에 대한 방송 빈도수는 점차로 감소하게 되므로 캐쉬에 더 유지되어야 한다.

3.2 대체 전략

데이터 방송 환경에서 제안하는 효율적인 캐쉬 대체 전략은 다음과 같은 직관을 기반으로 한다.

I1) 클라이언트가 요구를 자주 하는 데이터 페이지는 캐쉬에 있어야 한다.
클라이언트가 필요로 하는 데이터는 캐쉬에 있거나(cache hit) 방송중이거나(air hit), 혹은 서버에게 요청을 하는(pull request) 3 가지뿐이다. air 히트인 경우 원하는 데이터가 방송되기를 기다려야 하며 요청하는 경우도 최소한 다음 방송까지 기다려야 한다. 따라서, 자주 요구되는 데이터는 캐쉬에 있어야 가장 빠른 응답시간을 얻을 수 있다.

I2) 클라이언트가 최근에 요구한 페이지일수록 캐쉬에 있어야 한다.
전통적인 시스템에서 가장 많이 고려한 것으로 프로그램의 속성상 참조 지역성에 따라 최근에 액세스된 데이터가 향후에도 많이 액세스될 것이라는 직관을 기반으로 하고 있다.

I3) 방송이 적게 되는 페이지일수록 캐쉬에 있어야 한다.
제안한 방송 알고리즘에 따르면 캐쉬에 데이터가 오래 있을수록 MFA 값이 서버측에서 감소되어 방송의 횟수가 줄어들게 된다. 방송의 횟수가 줄어들게 되면 클라이언트의 대기시간이 길어지게 되어 미스 비용이 증가하게 된다. 따라서, 방송 횟수가 적은 페이지일수록 미스 비용을 줄이기 위해 캐쉬에 존재하는 것이 바람직하다.

I1)을 반영하기 위해 각 데이터마다 클라이언트가 요청한 횟수(R_n)를 유지한다. 클라이언트가 데이터를 요구할 때마다 해당 데이터의 R_n 값을 1씩 증가시킨다. R_n 값은 초기에 0으로 설정된다.

I2)를 반영하기 위해 데이터 방송 주기 초에 서버로부터 받은 버전번호와 방송 데이터 식별자 정보를 이용하여 각 데이터마다 요구가 충족되는 시점의 방송 버전번호(R_v)를 유지한다. 클라이언트가 데이터를 요구할 때 해당 데이터의 R_v 값은 요구한 데이터가 서비스되는 시점의 방송 버전번호로 설정된다. 클라이언트가 요청하여 방송되는 데이터의 R_v 값은 요구한 시점의 버전보다 이후 버전번호를 갖게 됨을 주목하라. R_v 값은 데이터 요구나 캐쉬내의 존재 여부와 상관없이 지속적으로 유지된다.

I3)를 반영하기 위해 데이터 방송 주기 초에 서버로부터 받은 버전번호와 방송 데이터 식별자 정보를 이용하여 각 데이터마다 방송된 횟수(B_f)를 유지한다. 데이터가 캐쉬에 존재하는 동안 B_f는 변하지 않으며 캐쉬에서 희생자로 선택되어 더 이상 캐쉬에 존재하지 않게 되면 기본값 0으로 다시 초기화된다.

이상으로부터 캐쉬 대체를 위한 전략은 식 (1) 로부터 유도된 값(Value for Victim: VV)이 제일 작은 데이터 페이지를 희생자로 선택한다.

$$VV = R_n \cdot R_v \div B_f \qquad (1)$$

식 (1)에서 요구 회수가 많을수록, 최근에 요구할수록, 그리고 방송 횟수가 적을수록 희생자로 선택될 가능성이 줄어들게 됨을 쉽게 알 수 있다.

클라이언트가 요청하여 방송된 데이터 페이지는 비록 다른 클라이언트들도 요구하여 점차로 방송될 가능성이 높아질 수 도 있지만 필요성이 자신 혹은 소수 클라이언트에 국한될 가능성이 있어 다시 방송될 가능성이 줄어들 수 있으므로 무조건 캐쉬에 넣는 전략을 취한다.

VV 값을 구하기 위해 클라이언트는 (PID, R_n, R_v, B_f)로 구성되는 테이블을 유지한다. 여기서, PID는 데이터 페이지의 식별자를 의미한다. 페이지 수가 많아져 테이블의 크기가 매우 커지게 되면 검색의

효율성을 위해 VV를 포함한 특정값이 일정값 이하인 페이지들을 삭제하거나 PID를 키로 하는 해싱 등의 방법을 고려할 수 있다. 한편, 캐쉬 내에 있는 페이지들에 대한 VV 값중에서 가장 낮은 값을 갖는 페이지가 희생자로 선택되므로 이들 값들은 별도의 최소 히프(min heap) 구조나 정렬 리스트로 유지하여 효율성을 향상시킬 수 있다.

4. 결론

본 논문에서는 클라이언트의 데이터 요구 사항을 반영하기 위해 비트 벡터를 이용하여 사용자의 요구 패턴을 반영하는 데이터 방송 시스템에서 방송 알고리즘의 기본적인 특성을 고려한 캐쉬 대체 전략을 제안하였다. 제안한 대체 전략은 클라이언트의 데이터 액세스 빈도수와 최근성의 척도로 방송 버전번호를 사용했으며 방송 알고리즘이 캐쉬에 있는 데이터의 방송 가능성이나 방송 빈도수를 줄이기 때문에 방송 횟수를 함께 고려하였다.

제안한 전략은 캐쉬의 히트율을 향상시키고 아울러 미스 비용을 줄이기 위해 여러 가지 요소를 복합적으로 고려하였으나 이러한 복합적인 고려가 대체 전략에 얼마나 효율적인가를 검증하기 위해 제안한 전략에 대한 성능 분석이 수반되어야 할 것이다.

참고문헌

[1] D. Aksoy, M. Altinel, R. Bose, U. Cetintemel, M. Franklin, J. Wang, S. Zdonik, "Research in Data Broadcast and Dissemination", AMCP, pp. 194 - 207, 1998.

[2] S. Acharya, M. Franklin, S. Zdonik, "Balancing Push and Pull for Data Broadcast", Proc. of ACM SIGMOD, Tuscon, Arizona, pp. 183-194, May 1997.

[3] S. Acharya, R. Alonso, M. Franklin, S. Zdonik, "Broadcast Disks : Data Management for Asymmetric Communication Environments", Proc. of ACM SIGMOD, pp. 199 - 210, 1995.

[4] D. Barbara, "Mobile Computing and Databases - A Survey", IEEE Transactions on Knowledge Engineering, Vol. 11, No. 1, pp. 108-117, January/February 1999.

[5] D. Barbaba, "Certification Reports: Supporting Transactions in Wireless Systems," Proc. 17th Int. Conf. Distributed Computing Systems, Baltimore, May 1997.

[6] D. barbaba and T. Imielinski, "Sleepers and Workaholics: Caching Strategies in Mobile Environments," ACM SIGMOD, pp. 1-12, May 1994.

[7] G. Forman and J. Zahorjan, "The Challeges of Mobile Computing," IEEE Computer, 27(6), April 1994.

[8] Q. Hu, D. L. lee, and W. C. Lee,"Dynamic Data Delivery in Wireless Communication Environment," white Paper, GTE Lab. Incorporated. 1999.

[9] E. O'Neil, P. O'Neil, G. Weikum, "The LRU-K Page Replacement Algorithm for Database Disk Buffering", Proc. of ACM SIGMOD, pp. 297-306, May 1993.

[10] K. L. Wu, P. S. Yu, and M. S. Chen, "Energy-Efficient Caching for Wireless Mobile Computing," 12th Int. Conf. data Eng., pp.336-345, Feb. 1996.