

풀 기반 데이터 방송 시스템에서의 데이터 인기도를 고려한 캐싱 전략

(A Caching Strategy Considering Data Popularity in Pull-Based Data Broadcast Systems)

신 동 천 [†]

(Dong Cheon Shin)

요 약 캐싱은 데이터 방송 시스템에서 방송 요청의 경쟁을 줄임으로써 좁은 대역폭으로 인한 시스템 성능의 저하를 완화할 수 있는 유용한 방법이다. 본 논문에서는, 풀 기반 방송 시스템에서 클라이언트들간의 데이터 인기도를 반영하는 캐싱 전략을 제안한다. 아울러, 데이터 방송 버전을 이용하여 데이터 접근의 최근성을 반영할 수 있도록 하고 제안한 전략의 성능을 시뮬레이션을 통하여 평가한다. 성능 평가에 따르면, 히트율과 미스 비용을 함께 고려한 전략이 전통적 전략인 LRU 보다 성능 우위를 보이고 있다. 클라이언트들의 데이터 인기도를 고려한 전략은 일부 경우에 있어 성능 우위를 보여 주고 있다.

키워드 : 데이터 방송, 캐싱, 데이터 인기도

Abstract A caching is a useful technique to alleviate performance degradation due to the inherent narrow bandwidth by reducing contention of broadcast requests. In this paper, we propose a caching strategy for pull-based data broadcast system which considers data popularity among clients. In addition, the proposed strategy also reflects recentness of data access based on data broadcast version. Then, we evaluate the performance of proposed strategy through a simulation approach. According to the results, the strategy considering both hit ratio and miss cost shows better performance than the traditional LRU. In addition, the strategy considering data popularity among clients shows better performance in some cases.

Key words : data broadcast, caching, data popularity

1. 서 론

최근 들어, 무선통신 기술의 발달로 인하여 이동 컴퓨팅이라는 새로운 컴퓨팅 패러다임에 대한 관심이 고조되고 있다. 특히 이동하면서 무선으로 데이터를 전송하는 기술의 발달은 새롭고 폭 넓은 응용들을 출현시키고 있다. 이동 컴퓨팅 환경의 특징은 크게 무선과 이동성으로 나누어 볼 수 있다[1]. 기존의 유선망과 비교하여 무선망은 상대적으로 낮은 대역폭(bandwidth), 잦은 단절이라는 환경적인 제약요인이 있다. 이동성은 HPC(Hand held PC), PDA(Personal Digital Assistant), 휴대폰, 노트북, 팜톱 같은 휴대가 가능한 단말장치들로 보통은 가능하게 된다. 따라서 이동성으로 인하여 제한된 컴퓨

팅 능력, 낮은 배터리 수명, 제한된 인터페이스 화면이라는 기능적인 제약요인이 있다. 이러한 이동 컴퓨팅 환경의 특징 때문에 기존의 컴퓨팅 환경에서 제안된 여러 가지 기법이나 알고리즘들은 적용될 수 없거나 낮은 성능을 보이게 된다. 예를 들어, 캐싱전략, 질의처리, 트랜잭션 처리 문제에 무선과 이동성이라는 요인을 고려한 기법들이 효율적이게 된다[2].

제한된 대역폭에서 다수의 클라이언트에게 데이터를 전송하는 방법으로 데이터 방송(data broadcast) 기법이 관심을 끌어 왔다. 데이터 방송 기법이 갖는 가장 큰 장점중의 하나는 클라이언트의 수에 관계없이 데이터 서비스를 할 수 있다는 점이다. 방송되는 데이터 선결과 관련하여 데이터 방송 시스템은 크게 3가지 형태로 구분할 수 있다[3]. 서버가 클라이언트의 데이터에 대한 요청에 의존하지 않고 데이터 접근 확률이나 이력 같은 정적인 정보를 이용하여 자체 방송 알고리즘에 따라 필요한 데이터를 주기적으로 혹은 비 주기적으로 방송하

· 이 논문은 2005학년도 중앙대학교 학술연구비 지원에 의한 것임

[†] 중신회원 : 중앙대학교 정보시스템학과 교수

dcshin@cau.ac.kr

논문접수 : 2006년 2월 2일

심사완료 : 2006년 5월 16일

는 푸쉬(push) 기반의 방송 형태가 있다. 반면에 필요한 데이터를 클라이언트가 서버에 요청하여 서버가 데이터를 방송해 주는 풀(pull) 기반의 형태가 있다. 따라서, 푸쉬 기반 시스템은 클라이언트의 동적인 데이터 요구에 대한 패턴을 정확하게 파악할 수 없으므로 클라이언트의 요구 변화 패턴이 변하는 경우에 능동적으로 대처할 수 없는 근본적인 문제점을 갖게 된다. 한편 데이터 방송 프로그램의 일부는 클라이언트의 요구에 따라 구성하고 나머지는 접근 확률이나 이력에 기반하여 구성하는 혼합(hybrid) 형태가 있다.

제한된 대역폭으로 인하여 과거에는 대부분의 연구가 풀 기반의 응용보다는 푸쉬 기반의 응용에 중점을 두어 왔다고 할 수 있다. 그러나, 필요할 때 데이터를 요청하는 VOD(Video On Demand), NOD(News On Demand) 같은 많은 풀 기반의 응용이 무선 통신 기술의 발달과 함께 출현하고 있다[4,5]. 뿐만 아니라, 클라이언트가 필요로 하는 데이터도 매우 다양해짐에 따라 데이터 접근의 형태 변화도 많을 것으로 예상할 수 있다. 따라서, 풀 기반의 응용에 더욱 많은 관심을 기울일 필요가 있다.

한편, 데이터 방송 환경에서는 클라이언트에 캐시를 두어 필요한 데이터가 방송되기를 기다리지 않고 캐시를 액세스하여 시스템의 효율성을 높일 수 있다. 캐시에 있는 데이터는 서버에 있는 데이터의 사본으로 간주될 수 있으므로 일관성 유지 문제를 해결하고자 하는 여러 노력이 있어 왔다[6,7]. 캐시를 고려하는 경우 해결해야 될 또 다른 중요한 문제는 캐싱 전략이다.

전통적인 시스템 환경을 위한 여러 가지 대체 전략들이 제안되었다[8]. 그러나, 이러한 대체 전략들은 이동 컴퓨팅 환경의 특징을 고려하지 않은 전략들이기 때문에 좋은 성능을 보이기 어렵다. 따라서, 이동 컴퓨팅 시스템에서 데이터 방송 환경을 고려한 많은 전략들이 또한 제안되었다[3,9-12]. 이러한 전략들의 대부분은 데이터 방송 환경에서는 캐시의 히트율을 높이는 것뿐만 아니라 캐시 미스가 발생하는 경우 서버로부터 요구하는 데이터가 방송될 때까지 기다려야 하므로 미스에 따른 비용까지 고려해야 한다는 관점에서 제안되었다. 그러나, 대부분의 전략들은 데이터 접근 확률등과 같은 정적인 요소를 이용한 전략으로 푸쉬 기반의 시스템에 적합한 전략들이다.

서버가 데이터를 주기적이고 반복적으로 방송하는 푸쉬 기반의 방송 시스템과 달리, 풀 기반 시스템에서는 클라이언트의 요구에 따라 동적으로 데이터가 방송되어야 한다. 이는 데이터 방송이 전적으로 클라이언트의 요청에 의존하는 풀 기반 시스템에서는 데이터 방송의 효율성을 위해 푸쉬 기반 시스템에서 캐싱 전략을 위해 사용한 접근확률이나 이력 같은 정적인 요소를 도입하

기 어려움을 의미한다. 예를 들어, 풀 기반 시스템에서는 실질적인 데이터 접근 빈도수나 데이터 방송 요청 후 방송 때까지 경과된 시간등과 같은 동적인 요소를 고려할 수 있다.

뿐만 아니라, 대부분의 캐싱 전략이 자신에 관련된 데이터 인기도 즉, 접근 빈도수는 고려하지만 다른 클라이언트들의 데이터 접근 유형 즉, 다른 클라이언트들의 데이터 인기도는 전혀 고려하고 있지 않다. 풀 기반 시스템을 위한 캐싱 전략[10]에서도 예외는 아니다. 데이터에 대해 방송 요청을 한 클라이언트의 수는 데이터에 대한 인기도를 의미하므로, 이를 캐싱 전략에 고려함으로써 클라이언트는 자신이 필요한 데이터를 미리 캐싱하는 효과를 얻어 시스템의 성능 향상에 기여할 수 있을 것으로 기대할 수 있다.

본 논문에서는 풀 기반의 시스템에서 클라이언트 자신이 유지하는 정보만 고려 하는 기존의 연구와 달리 서버로부터 얻을 수 있는 정보인 클라이언트들의 데이터 인기도를 반영하는 캐싱 전략을 제안한다. 아울러, 전통적인 시스템에서 대표적인 캐싱 전략인 LRU(Least Recently Used) 전략의 특징인 데이터 접근의 최근성을 데이터 방송 시스템에서 효과적으로 반영하기 위하여 데이터 방송 버전을 고려한다. 그리고, 제안한 캐싱 전략에서 인기도의 고려가 성능에 미치는 영향을 시뮬레이션을 통하여 분석하고 평가한다.

본 논문의 구성은 다음과 같다. 2장에서는 주요한 기존 캐싱 전략들과 본 논문에서 성능 평가를 위해 사용하고 있는 방송 알고리즘을 간략히 소개한다. 3장에서는 캐싱 전략에서 고려하는 요소들을 소개하고 4장에서는 캐싱 전략을 제안한다. 5장에서는 성능 평가에 대해 기술한다. 끝으로, 6장에서는 결론을 맺는다.

2. 관련 연구

2.1 캐싱 전략

풀 기반 데이터 방송 시스템을 위한 캐싱 전략에서 클라이언트들의 데이터 인기도를 고려하는 전략의 차별성을 위해 기존의 주요 캐싱 전략을 간략히 소개한다.

- LRU(Least Recently Used): 전통적인 시스템에서 많이 쓰이는 대표적인 캐싱 전략으로 가장 최근에 접근되지 않은 데이터가 대체될 희생자로 선정되는 전략이다. 무선 환경, 특히, 데이터 방송 시스템 환경에서는 좋은 전략으로 받아들여지지 않고 있어 데이터 방송을 위한 캐싱 전략들이 제안되어 왔다.
- PIX(Probability of access and Frequency of broadcast): 각 페이지의 접근 확률과 방송 주기 내에 방송되는 페이지의 방송 빈도를 이용하여 대체할 데이터를 선정하는 전략이다. 각 페이지의 PIX 값은 접

근확률을 방송빈도수로 나눈 값으로(접근확률/방송 빈도수), PIX 값이 작은 데이터를 캐쉬에서 대체할 데이터로 선정한다. 따라서, 방송빈도가 상대적으로 매우 적거나 데이터의 접근 확률이 상대적으로 매우 큰 데이터가 캐쉬에 남게 되므로 히트율과 미스 비용을 함께 고려하는 전략이라 할 수 있다.

- PT(Probability and Time to broadcast): PIX가 미스비용 고려를 위해 사용하는 정적인 값인 방송 빈도수 대신에 다음 방송까지 걸리는 실질적인 시간을 사용하는 전략이다. 각 데이터의 PT값은 접근확률*(다음 방송 시간-현재 시간)으로, PT 값이 가장 작은 페이지가 대체할 데이터로 선정된다.
- CF(Closest First): 데이터 페이지 미스가 발생할 경우 미리 정해진 방송 스케줄에서 가장 먼저 방송될 데이터를 대체할 페이지로 선정하는 알고리즘이다. 따라서, CF는 LRU와 반대 개념의 알고리즘으로, LRU가 히트율을 높이기 위한 전략이라고 하면 CF는 미스비용을 줄이기 위한 전략이다.
- Gray: 히트율과 미스비용 모두를 고려하기 위해 CF와 LRU 전략을 병행하여 대체할 데이터를 선정하는 전략이다. 모든 데이터는 {black, gray, white} 세 가지 상태중 하나를 부여 받는다. black은 현재 단계에서 요청된 데이터를 의미하고, gray는 이전 단계에서 요청된 데이터를, white는 요청되지 않은 데이터를 의미한다. 현재 단계에서 데이터가 요청되면 black 표시를 하고, 데이터 페이지 미스가 발생할 경우 white로 표시된 데이터 중에서 CF를 이용하여 대체할 데이터를 선정한다.
- PWT(Priority and Waiting Time): 각 데이터에 클라이언트가 요청하는 횟수인 인기도(P)와 데이터 페이지 미스로 인한 데이터 요청 후 방송 때까지 걸리는 시간인 대기시간(waiting time)을 이용하는 전략이다. 이 값이 가장 작은 데이터가 대체할 페이지로 선정된다. LRU는 전통적인 시스템을 위한 전략인 반면에 나머지 전략들은 모두 데이터 방송 시스템을 위한 전략들이다. 그러나, PWT는 자신의 데이터 접근 빈도수(인기도)와 대기시간 같은 동적인 요소를 고려함으로써 풀 기반 시스템을 위한 전략이 가능하게 되었다. 반면에 다른 전략들은 접근 확률이나 정해진 방송 스케줄 같은 정적인 요소를 고려함으로써 푸시 기반 시스템에 적합한 전략들이라 할 수 있다. 표 1에서 보는 바와 같이 데이터 방송 시스템을 위한 전략들은 LRU와 달리 캐싱에 필요한 정보로 미스 비용을 고려하고 있으나 클라이언트들 사이의 데이터 인기도를 고려하지 않고 있다.

2.2 RxW 방송 알고리즘 (13)

RxW(Request x Waiting) 방송 알고리즘은 클라이

표 1 캐시 대체 전략

이름 \ 고려요소	히트율	미스비용	인기도
LRU[8]	O	X	X
PIX[3]	O	O	X
PT[3]	O	O	X
CF[9]	X	O	X
Gray[11]	O	O	X
PWT[10]	O	O	X

언트의 요청 빈도수와 요청시간을 근간으로 데이터를 선택하여 방송하는 스케줄링 기법이다. 스케줄링을 위하여 사용하는 기준 값은 클라이언트의 데이터 방송 요청 수 R과 요청 접수 후 방송되는 현재 시간까지의 대기시간(wait time) W의 곱을 사용한다. 방송이 요청된 데이터에 대해 아래와 같은 정보가 큐에 유지될 수 있다.

<Pid, Arrival_first_time, R>

Pid는 방송이 요청된 데이터의 식별자이며, Arrival_first_time은 여러 클라이언트들이 요청할 수 있으므로 처음으로 방송 요청된 시간을 나타내며, R은 요청된 횟수를 나타낸다. 서버는 각 데이터의 Arrival_first_time과 방송이 되는 현재 시간의 차로 대기 시간인 W를 계산하여 R과 곱한 값을 (RxW) 계산한다. 방송 스케줄링에서 계산된 RxW값을 비교하여 그 값에 따라 데이터를 정렬하고 값이 큰 순서에 따라 해당 데이터를 방송하게 된다.

RxW 기법은 간편한 연산으로 클라이언트의 요청율과 방송 대기시간을 함께 스케줄링에 반영할 수 있다. 방송 대기 시간을 함께 고려함으로써 요청율이 낮은 일부 데이터가 무한정 방송이 늦어지는 것을 방지하게 된다. 이는 데이터 방송 시스템에서 클라이언트에 대한 평균 응답 시간을 개선시킬 수 있게 된다. 뿐 만 아니라, 이 기법은 클라이언트의 요구에 따라 방송 스케줄링이 이루어짐으로써 클라이언트의 데이터 요청 변화를 방송 스케줄링에 동적으로 반영할 수 있게 된다. 따라서, 풀 기반의 데이터 방송 시스템에 적합한 방송 알고리즘으로 생각할 수 있다.

3. 캐싱 전략을 위한 요소들

3.1 클라이언트로부터의 고려 요소

캐싱 전략을 위해 클라이언트로부터 생성되는 요소로는 데이터 접근 빈도수와 대기시간을 고려한다. 데이터 접근 빈도수는 데이터에 대한 접근을 요청한 횟수를 의미한다. 대기시간은 접근 요청한 데이터가 캐쉬에 존재하지 않는 경우 데이터 방송 요청 후에 요청한 데이터가 방송되기까지 경과한 시간을 의미한다. 이러한 요소에 관한 정보는 각 데이터에 대해 클라이언트가 유지할

수 있다. 데이터 접근 빈도수는 궁극적으로 히트율을 높이기 위해 고려한 요소이며 대기시간은 미스비용을 줄이기 위해 고려한 요소이다.

데이터 접근 빈도수는 클라이언트의 데이터에 대한 관심 정도를 나타낸다. 데이터에 대한 클라이언트의 관심도는 시간이 흐름에 따라 변할 수 있다. 예를 들어, NOD나 VOD 같은 무선 서비스에서 어떤 데이터에 대한 액세스 수는 일정 기간 동안 급격히 증가하고 나서 점진적으로 감소할 수 있다. 그러므로, 증가하는 데이터 접근 빈도수와 감소하는 데이터 빈도수를 갖는 데이터를 구별할 필요가 있다. 즉, 데이터 접근의 최근성을 고려할 필요가 있다. 접근의 최근성을 고려함으로써 최근에 증가하는 데이터 접근 빈도수를 갖는 데이터가 최근에 감소하는 데이터 접근 빈도수를 갖는 데이터보다 캐쉬에 존재할 확률을 높일 수 있다.

본 논문에서는, 데이터 접근의 최근성을 반영하기 위해 서버로부터 데이터가 방송될 때 데이터와 함께 받는 방송 버전번호를 이용한다. 방송 버전번호는 매 방송 주기마다 1씩 증가하는 값으로 정의한다. 따라서, 최근에 방송될수록 큰 값의 방송 버전번호를 갖게 된다. 즉, 임의의 방송 주기 i 동안의 접근 수(A_i)를 그대로 사용하지 않고 방송 버전 번호(V_i)를 곱한 값($A_i * V_i$)을 사용함으로써 같은 접근 수를 갖더라도 방송 버전번호가 큰 데이터 즉, 최근에 방송된 데이터가 큰 값을 갖게 함으로써 최근성을 반영한다. 현 방송주기 $i+1$ 에서 데이터 접근 수는 직전 방송주기 i 까지 누적된 접근 수가 되므로 데이터 접근 빈도수(F_a)는 식 (1)과 같이 표현할 수 있다.

$$F_a = \sum (A_i * V_i) \quad (1)$$

식 (1)로부터 동일한 접근 수를 갖는다 하여도 최근에 접근 빈도수가 많은 데이터가 더 큰 접근 빈도수를 갖게 됨을 알 수 있다. 특히, 방송 버전번호를 이용함으로써 각 데이터 접근의 최근성 반영에 점진적인 가중치를 적용하고 있는 효과를 볼 수 있음을 알 수 있다.

대기시간은 데이터 방송 환경에 의존하므로 가장 최근의 데이터 방송 환경을 반영하기 위하여 식 (2)와 같이 최근의 대기시간에 더 많은 가중치를 두는 값으로 대기시간(WT_{val})을 설정한다.

$$WT_{val} = Val(WT_i) \quad (2)$$

식 (2)에서 $Val(WT_i)$ 는 식 (3)과 같이 반복적으로 정의된다.

$$Val(WT_i) = (Val(WT_{i-1}) + WT_i) / 2 \quad (3)$$

식 (3)에서 $Val(WT_{i-1})$ 는 i 번째 방송 요청 전까지 가중치가 적용되어 누적된 대기시간을 의미하며 WT_i 는 i 번째 방송 요청에 대한 대기시간을 의미한다.

3.2 서버로부터의 고려요소

미스비용을 줄이기 위해서는 방송 빈도수가 낮은 데이터가 방송 빈도수가 높은 데이터보다 캐쉬에 존재하는 것이 바람직한 당연하다. 또한, 대부분의 방송 알고리즘들에서 데이터의 방송 빈도수는 일반적으로 클라이언트가 요청한 방송 요청 수에 비례한다고 말할 수 있다. 따라서, 클라이언트 사이에 데이터 인기도를 반영하기 위해서 그 데이터에 방송 요청을 한 클라이언트들의 수를 고려하는 것이 바람직하다고 할 수 있다.

데이터 인기도는 시간의 경과에 따른 클라이언트의 데이터에 대한 관심의 변화를 반영한다. 그림 1은 데이터 인기도와 관련하여 한 유형의 특성을 보여주고 있다 [4]. 그림 1에서 y 축은 인기도를, x 축은 시간을 의미하며 특정 데이터에 대한 접근 확률이 일정기간 동안 급격히 증가하다가 시간이 흐름에 따라 점진적으로 감소함을 보여주고 있다. 이는 새로운 관심을 모으는 데이터가 발생하는 경우 클라이언트들의 접근 변화 형태를 보여 준다고 할 수 있다. 데이터 인기도와 관련한 이러한 유형은 데이터 방송 시스템에서 향후 더 많은 응용 서비스에서 나타날 수 있다[4,5]. 따라서, 캐싱 전략에서 이러한 유형의 데이터 인기도를 반영함으로써 필요로 하는 데이터를 미리 캐싱하는 효과를 얻을 수 있다면 성능 향상에 기여할 수 있다.

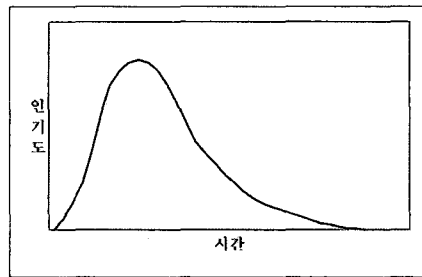


그림 1 데이터 인기도 유형

데이터 인기도를 반영하기 위하여 데이터에 대해 방송 요청을 한 클라이언트의 수를 사용한다. 이러한 정보는 서버가 데이터를 방송할 때 해당 데이터와 함께 보낸다. 데이터에 대한 인기가 그림 1에서처럼 동적으로 변할 수 있으므로 다른 요소와 다르게 가장 최근의 데이터 인기도를 반영하여 데이터 인기도(P)를 식 (4)처럼 표현한다.

$$P = C_n \quad (4)$$

식 (4)에서 C_n 은 현재 방송주기 동안에 방송 요청한 클라이언트의 수이며 1과 최대 클라이언트 수인 n 사이의 값을 갖는다.

4. 캐싱 전략

데이터 방송 시스템에서 효율적인 캐싱 전략은 히트를 향상뿐만 아니라 미스에 따른 비용도 줄여야 한다고 지적하였다. 또한, 클라이언트들의 데이터에 대한 인기도를 반영함으로써 가까운 미래에 필요로 할 수 있는 데이터를 미리 캐싱 하는 일이 필요함도 지적하였다. 이러한 원칙을 반영하는 캐싱 전략을 제시하기 전에 히트율과 미스 비용에 관련한 몇 가지 직관을 제시한다.

• H1: 데이터 접근 빈도수가 높은 데이터는 캐시에 존재할 필요가 있다.

클라이언트가 필요로 하는 데이터는 캐시에 존재하거나, 방송 중, 혹은 서버에게 요청을 하게 된다. 방송 중인 경우는 현재 방송 주기 동안에 해당 데이터가 방송 되기를 기다려야 하며 서버에게 요청을 하는 경우는 다음 방송 주기에 방송 되기를 기다려야 한다. 캐시에 존재하는 경우는 바로 액세스할 수 있으므로 데이터 접근 빈도수가 높은 데이터는 캐시에 존재하게 하는 것이 히트율을 높이는 것이 된다.

• H2: 대기 시간이 큰 데이터는 캐시에 존재할 필요가 있다.

대기시간은 미스비용과 밀접한 관련이 있다. 즉, 대기 시간이 길수록 미스 비용도 크게 된다. 따라서, 대기 시간이 긴 데이터일수록 캐시에 존재하여야 미스 비용을 줄일 수 있다.

• H3: 높은 데이터 인기도를 갖는 데이터는 캐시에 존재할 필요가 있다.

앞서 지적한 바와 같이 인기도가 증가하고 있는 데이터는 클라이언트들 사이에서 당분간 인기도를 유지할 수 있다. 이는 히트율을 높이고 미스비용을 줄이기 위해 인기도가 높아지는 데이터는 캐시에 존재할 필요가 있음을 의미한다.

직관 H1을 반영하기 위해서는 접근 빈도수가 큰 데이터가 캐시에 존재하도록 하여야 하며 직관 H2를 반영하기 위해서는 역시 대기시간이 큰 데이터를 캐시에 존재하도록 해야 함은 자명하다. 따라서, 직관 H1과 H2만을 반영하는 캐싱 전략 AWT는 접근 빈도수와 대기시간의 곱으로 식 (5)와 같이 설정할 수 있다. 전략 AWT에서는 2 요소간의 곱의 값이 제일 작은 데이터가 대체될 희생자로 선정되므로 접근 빈도수가 클수록 그리고 대기시간이 길수록 캐시에 존재할 가능성이 높게 되어 직관 H1과 H2를 반영하고 있음을 알 수 있다.

$$AWT : Fa * WT_{val} \quad (5)$$

[10]에서도 AWT와 마찬가지로 데이터 접근 빈도수와 대기시간을 고려하고 있으나 최근성 반영을 위해 임의로 정해진 가중치를 사용하고 있다. 이러한 가중치 설정 문제는 최적화 문제이므로 또 다른 문제가 된다. 이러한 이유로 본 논문에서는 데이터 접근의 최근성과 대

기 시간의 최근성을 반영하기 위해 방송 버전 번호 같은 다른 방안을 사용한다. AWT 전략의 제시는 데이터 인기도를 고려한 전략이 그렇지 않은 전략에 비해 성능에 어떠한 영향을 미치는지를 분석하기 위해 제시되었으며 평가 대상으로 선정하였다.

한편, 직관 H3를 반영하기 위해서는 인기도가 높은 데이터를 캐시에 존재하도록 해야 함은 역시 자명하다. 따라서, 직관 H1 및 H2와 함께 직관 H3를 반영하는 캐싱 전략 AWTP는 식 (6)과 같이 3 요소의 곱으로 설정할 수 있다. 전략 AWTP에서도 역시 3 요소간의 곱의 값이 제일 작은 데이터가 대체될 희생자로 선정되므로 접근빈도수 및 대기시간이 크고 인기도가 높을수록 캐시에 존재할 가능성이 높게 되어 위의 3가지 직관 모두를 반영하고 있음을 알 수 있다.

$$AWTP : Fa * WT_{val} * P \quad (6)$$

2가지 전략 모두에서 고려한 각 요소들의 곱을 사용하는 것은 이들 값의 단위가 동일하지 않기 때문에 예를 들어 곱을 사용할 수 없기 때문이다. 이들간의 가중치를 고려해 보는 것은 좋은 시도라 생각되지만 곱을 사용하는 경우 이들 요소간의 가중치 고려는 의미가 없게 되어 본 연구에서는 고려할 수 없게 된다.

5. 성능 평가

5.1 성능 평가 환경

제한한 캐싱 전략들을 기존의 LRU 전략의 성능을 시뮬레이션을 통하여 평가하였다. 시뮬레이션 모델은 그림 2와 같으며 CSIM[14]을 이용하여 구현하였다.

각 클라이언트는 페이지 수신기, 질의 생성기, 캐시 관리자, Pull 관리자의 4개의 프로세스로 구성된다. 먼저 질의 생성기에 의해 하나의 읽기 연산이 제기되면 캐시 관리자를 통해 해당 데이터가 캐시에 존재하는지 확인하게 된다. 만일 캐시에서 해당 데이터를 찾을 수 없을 경우에는 서버에 요청(pull)을 제기하게 된다. 서버는 RxW 방송 기법에 따라 요청된 데이터 페이지를 방송한다. 뿐만 아니라, 클라이언트 캐싱을 위해서 서버측에

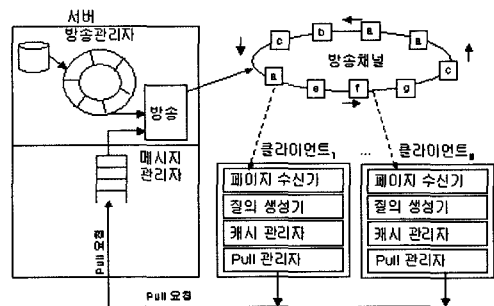


그림 2 시뮬레이션 모델

서 수집된 각 페이지의 인기도와 큐에서의 대기 시간을 필요로 하기 때문에 이러한 정보가 추가적으로 클라이언트에 전달된다.

자신이 필요로 하는 데이터를 획득한 클라이언트는 또 새로운 데이터 읽기 연산을 제기할 준비를 하게 된다. 각 클라이언트는 하나의 페이지를 요청하여 받을 때까지 새로운 페이지에 대한 연산을 제기하지 않기 때문에 시뮬레이션 중 작업의 부하는 클라이언트의 수로 일정하게 유지될 수 있으므로 폐쇄된 큐잉 모형(closed queuing model)을 채택한 것으로 볼 수 있다. 표 2는 시뮬레이션 모델과 관련된 주요 시스템 및 응용 매개변수로 다른 과거 연구에서 고려하는 주요 변수들을 보여 주고 있다.

성능평가 시간은 200,000(시뮬레이션 시간)으로 하였으며 성능 왜곡을 최소화 하기 위해 전반부의 결과는 버리고 후반부의 결과만을 반영하였다. 데이터베이스의 데이터 페이지의 수를 실제 이동 응용에서 이용되는 크기에 비해 적은 5000으로 설정하였는데 이는 클라이언트 간에 적정 수준의 방송 히트율을 확보하기 위한 것이다. 클라이언트는 자신이 필요로 하는 데이터가 캐시에 포함되어 있지 않을 경우 서버에 해당 데이터를 직접 요청하여 얻게 되는데 이러한 요청은 서버의 큐에 보관되어 처리된다. 클라이언트가 제기한 데이터의 요청에 대해 해당 데이터를 얻은 후에 일정한 시간이 경과하는 동안 새로운 연산을 제기하지 않도록 하기 위해서 think_time을 3으로 설정하였다.

서버가 요청된 데이터를 클라이언트로 방송하면 각 클라이언트는 적어도 down_time 만큼의 시간, 즉 0.1이 지나야 해당 데이터를 읽을 수 있으며 또한 데이터 요청을 서버로 전송하기 위해서도 upload_time 만큼의 시간, 즉 0.1이 필요하다. 캐시로부터 데이터를 읽는데 소요되는 시간도 응답 시간에 영향을 줄 수 있으므로 캐시로부터 하나의 데이터 페이지를 액세스하는 시간, 즉 cache_acc_time을 0.001로 설정하였다.

클라이언트마다 선호하는 데이터를 다르게 만들기 위해서 각 클라이언트가 일정기간 동안 액세스하는 데이터의 범위(acc_range)를 1000으로 설정하여 전체 5000개의 데이터 페이지 중 1000개에 해당되는 부분만 액세스하도록 하였다. 이 1000개의 데이터 페이지 중에서 집중적으로 액세스하는 부분(hotarea_size)을 100으로 정하여 hotacc_rate의 값을 조정함으로써 각 클라이언트가 선호하는 데이터 페이지를 원하는 값으로 조절할 수 있도록 하였다. hotacc_rate는 클라이언트의 핫영역 접근 비율로 0.95 는 한 순간 각 클라이언트는 전체 5000개의 데이터 페이지 중 1000개의 페이지만 액세스하게 되고 그 중 95%의 요청이 100개의 부분에 집중됨을 의미한다. frequency와 offset은 각각 클라이언트의 접근 유형 변경 주기와 변경 크기를 나타내며 1000(시뮬레이션 시간)마다 클라이언트의 접근 영역이 100씩 변경됨을 의미한다.

5.2 성능 평가 결과

시뮬레이션 결과의 정확성을 제고시키기 위해서 각 시뮬레이션의 초기 단계에 수집된 자료는 결과에 반영하지 않았으며, 또한 각 시뮬레이션에 대해 난수 생성기의 시드 값을 변경시키면서 여러 차례 반복 실험한 결과에 대한 평균을 이용하여 최종 결과를 산출하였다. 데이터 방송 시스템 환경에서는 전통적 시스템 환경과는 달리 히트율(%)이 아니라 캐시 미스에 따른 대기 시간의 고려가 성능에 커다란 영향을 주므로 성능 척도로는 평균 응답 시간(시뮬레이션 시간)을 사용하였다. 캐싱 전략에서 캐쉬 크기에 대한 각 전략들의 성능을 비교하기 위하여 우선 캐쉬 크기를 변화시키면서 실험을 하였다.

그림 3은 캐쉬 크기 변화에 따른 전략들의 성능을 보여 주고 있다. 캐쉬 크기가 증가함에 따라 각 전략들의 응답 시간은 전반적으로 감소 추세를 보여 주고 있으며 일정 크기에 이르며 전략간 성능 차가 없음을 알 수 있다. 이는 전통적인 사실에 입각하여 볼 때 캐쉬 크기가 커짐에 따라 히트율이 자연스럽게 증가되면서 전반적

표 2 시스템 및 응용 매개변수

매개변수	설명	단위	설정값
db_size	데이터 페이지의 수	개(항목)	5000
num_clients	클라이언트의 수	개	300
cache_size	캐시 크기	개(항목)	50~300
think_time	각 요청 간 대기시간	시뮬레이션시간	3
down_time	데이터 방송 소요 시간	시뮬레이션시간	0.1
upload_time	방송 요청 소요 시간	시뮬레이션시간	0.1
cache_acc_time	캐시 접근 소요 시간	시뮬레이션시간	0.001
acc_range	일정주기동안 액세스 범위	개(항목)	1000
hot_size	acc_range내 핫영역 크기	개(항목)	100
hot_rate	핫영역 접근비율		0.95~0.05
frequency	접근 유형 변경 주기	시뮬레이션시간	1000
offset	접근 유형 변경크기	개(항목)	100

로 미스 비용이 감소함에 기인한다고 말할 수 있다. 그러나, 일부 구간(특히, 캐쉬 크기 150)에서의 예외적인 결과는 데이터 방송 시스템에서 응답 시간에 영향을 주는 요인이 캐쉬 크기뿐만 아니라 외적 요인이라 볼 수 있는 방송 알고리즘과 같은 서버 측의 상황에 영향을 많이 받을 수 있음을 보여 주고 있다고 판단된다. 즉, 히트율은 증가되지만 미스가 발생하는 경우 서버에 방송 요청을 한 일부 데이터에 대한 방송이 방송 알고리즘의 스케줄링 전략으로 인하여 매우 늦어지는 경우에 대기 시간이 증가되어 전체적으로 응답 시간의 증가를 초래한 것으로 판단된다. 따라서, 성능 평가 대상이 되는 전략의 절대적인 값의 변화보다는 동일한 환경에서 전략들간의 상대적인 성능 비교 결과가 궁극적으로 의미를 갖는다고 할 수 있다.

그림 3에서 보는 바와 같이 미스 비용을 고려한 AWT가 LRU 보다 상대적으로 성능 우위를 보여주고 있다.

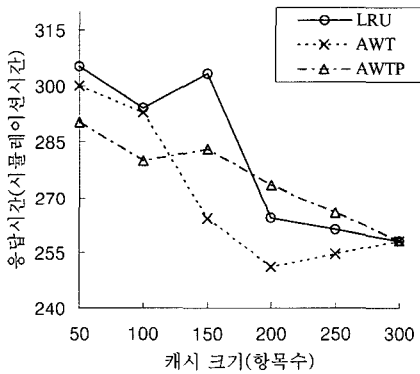
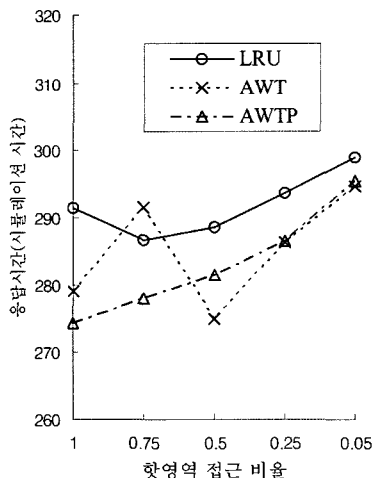


그림 3 캐쉬 크기 변화에 따른 성능

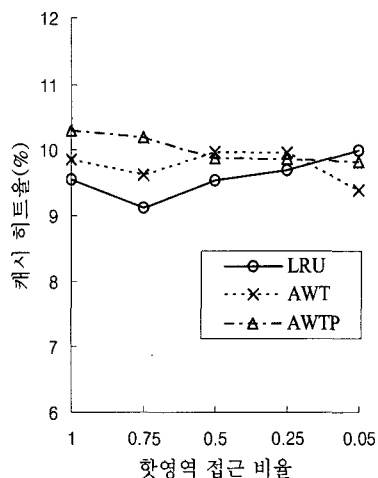


(a) 응답시간

이는 데이터 방송 시스템에서는 히트율 뿐만 아니라 미스에 따른 비용을 함께 고려하는 것이 바람직함을 입증해 주고 있음을 알 수 있다. 한편, 클라이언트들의 데이터 인기도를 고려한 AWT는 상대적으로 캐쉬 크기가 작은 경우 AWT 보다 성능 우위를 보이지만 캐쉬 크기가 커지는 경우는 오히려 그렇지 못함을 보여 주고 있다. 이는 캐쉬 크기가 작을수록 미리 캐싱하는 효과를 기대할 수 있지만 캐쉬 크기가 클수록 상대적으로 다른 전략에 비해 그 효과는 감소하기 때문이라 생각된다. 결국, 미리 캐싱하는 효과를 보지 못함에도 불구하고 이를 위한 요소인 데이터 인기도를 고려하게 됨에 따라 캐싱 전략이 비효율적으로 운영되기 때문으로 판단된다. 이 결과는 데이터 방송 시스템을 위한 캐싱 전략 수립에서 캐싱 전략에 영향을 줄 수 있는 여러 요소들을 복합적으로 고려하는 것이 반드시 좋은 전략이 될 수 없음을 입증하고 있다.

한편, 3가지 전략들의 성능 차이가 크게 나타나지 않은 것은 인기도와 대기 시간을 함께 고려하는 방송 알고리즘인 RxW 의 근본적인 특성에 따라 전략들의 약점을 보완해 준 결과로 볼 수 있다. 뿐만 아니라, 캐시 미스로 제기된 데이터 방송요청으로 증가하는 대기 시간이 캐싱 전략의 차이로 인해 얻어지는 시간에 비해 상대적으로 커진 것에 기인하고 있다. 결국, 캐싱 전략보다는 방송 알고리즘이 전체 시스템 성능에 결정적인 요인으로 작용하고 있음을 알 수 있다

클라이언트들의 데이터 인기도를 고려한 전략이 미리 캐싱 하는 효과를 갖고 있음을 알아보기 위해 클라이언트들의 데이터 접근 핫영역 비율을 변화시키며 실험을 한 결과를 그림 4는 보여 주고 있다.



(b) 히트율

그림 4 핫영역 비율에 따른 성능

그림 4(a)에서 보는 바와 같이 핫영역 비율이 작을수록 즉, 데이터 접근이 분산될수록 AWTP가 AWT에 비해 성능 우위를 보이고 있지 못함을 알 수 있다. 핫영역 비율이 작아짐은 클라이언트의 관심 데이터 영역이 커짐을 의미하므로 미리 캐싱하는 효과는 감소하는 것이 당연하다고 할 수 있다. 한편, 미스 비용을 고려한 AWT와 AWTP가 역시 LRU 보다는 전반적으로 성능 우위에 있음을 알 수 있다.

그림 4(b)는 히트율에 대한 비교 결과를 보여 주고 있다. 핫영역 비율이 클수록 AWTP가 AWT 보다 다소 높은 히트율을 보이고 있음을 알 수 있다. 이는 AWTP가 미리 캐싱하는 효과를 핫영역 비율이 클수록 가질 수 있음을 보여준다고 할 수 있다. 그림에서 히트율과 관련하여 또 한가지 주목할 것은 응답 시간은 히트율에 반드시 비례하지는 않는다는 사실이다. 이는 응답 시간은 히트율 뿐만 아니라 미스에 따른 비용 즉, 대기 시간에 영향을 받음을 말해 주고 있다. 따라서, 데이터 방송 시스템에서의 캐싱 전략은 히트율 향상과 함께 미스 비용을 최소화 할 수 있는 요소를 고려해야 함을 입증하고 있다.

6. 결론

데이터 방송 시스템에서 캐싱 전략은 기존의 전략과는 달리 히트율 뿐만 아니라 미스에 따른 비용을 줄일 수 있는 전략이 필요함은 널리 알려진 사실이다. 본 논문에서는 미스 비용을 줄일 수 있는 전략이 필요함을 실험을 통해 입증하였다. 또한, 클라이언트들의 데이터 인기도를 캐싱 전략에 반영하여 미리 캐싱하는 효과를 얻을 수 있는 전략을 제시하고 그 효과를 성능 평가를 통해 알아보았다.

성능 평가에 따르면, 히트율 향상과 미스 비용을 함께 고려한 전략인 AWT가 전통적 전략인 LRU 보다 전반적으로 다소 좋은 성능을 보여 주고 있어 미스 비용 고려가 필요함을 입증하고 있다. 그러나, 데이터 인기도를 함께 고려한 전략인 AWTP는 관심 데이터가 집중될수록 그리고 캐쉬가 작을수록 AWT 보다 다소 좋은 성능을 보여 주어 미리 캐싱하는 효과가 어느 정도 있음을 보여 주고 있다. 그러나, 그렇지 못한 경우에는 오히려 AWT 보다 낮은 성능을 보여 주어 데이터 방송 시스템에서 캐싱 전략 수립을 위해 여러 가지 요소를 복합적으로 고려하는 것이 성능 우위를 항상 보여 주지는 않음을 알 수 있다. 또한, 데이터 방송 시스템의 성능은 캐싱 전략에도 영향을 받지만 데이터 방송 알고리즘과 같은 요인에 의해 크게 좌우될 수 있음을 알 수 있다. 따라서, 효율적인 캐싱 전략은 데이터 방송 알고리즘의 특성을 고려하여 반영하는 전략이어야 함을 입증한다고 할 수 있다.

참고 문헌

- [1] D. Barbara, "Mobile Computing and Databases-A Survey," IEEE Transactions on Knowledge Engineering, Vol. 11, No. 1, pp. 108-117, 1999.
- [2] S. K. Madria and B. K. Bhargava, "A Transaction Model to Improve Data Availability in Mobile Computing," Journal of Distributed and Parallel Databases, Vol. 10, No. 2, pp. 127-160, 2001.
- [3] S. Acharya, M. Franklin, and S. Zdonik, "Balancing Push and Pull for Data Broadcast," Proc. of ACM SIGMOD, pp. 183-194, 1997.
- [4] T. Choi, Y. Kim, K. Chung, "A Prefetching Scheme based on the Analysis of User Access Patterns in News-On-Demand System," Proc. of ACM Int. Conf. on Multimedia, pp. 145-148, 1999.
- [5] C. Griwodz, M. Bär, L. C. Wolf, "Long-term Movie Popularity Models in Video-On-Demand Systems or the Life of an on-Demand Movie," Proc. of ACM Int. Conf. on Multimedia, pp. 349-357, 1997.
- [6] K. Y. Lai, Z. Tari, and P. Bertok, "Cost Efficient Broadcast based Cache Invalidation for Mobile Environments," Proc. of ACM Symposium on Applied Computing, pp. 871-877, 2003.
- [7] X. Shao and Y. Lu, "Maintain Cache Consistency of Mobile Database Using Dynamical Periodical Broadcast Strategy," Proc of Int. Conf. on Machine Learning and Cybernetics, pp. 2389-2393, 2003.
- [8] S. Galvin and P. B. Galvin, Operation System Concepts, 4th Edition, Addison Wesley, 1994.
- [9] S. Khanna and V. Liberatore, "On Broadcast Disk Paging," Proc. of ACM Symposium on the Theory of Computing, pp. 634-643, 1998.
- [10] Y. J. Lee and D. C. Shin, "Performance of Caching Strategies for Pull-based Data Broadcast Systems in Mobile Computing Environments," Journal of Computer Information Systems, Vol. 15, No. 4, pp. 102-115, 2005.
- [11] V. Liberatore, "Caching and Scheduling for Broadcast Disk Systems," Technical Report 98-71, UMIACS, 1998.
- [12] J. Xu, Q. Hu, W.-C. Lee, and D. L. Lee. "Performance Evaluation of Optimal Cache Replacement Policy for Wireless Data Dissemination," IEEE Transactions on Knowledge and Data Engineering, Vol. 16, No. 1, pp. 125-139, 2004.
- [13] D. Aksoy and M. Franklin, "RxW: A Scheduling Approach for Large-Scale Data Broadcast," IEEE/ACM Transactions on Networking, Vol. 7, No.6, pp. 846-860, 1999.
- [14] H. Schwetman, CSIM User's Guide for Use with CSIM Revision 16, Microelectronics and Computer Technology Corporation, 1992.



신 동 천

1985년 2월 서울대학교 컴퓨터 공학과 졸업(학사). 1987년 2월 한국과학기술원 전산학과 졸업(석사). 1991년 2월 한국과학기술원 전산학과 졸업(박사). 1991년 1월~1993년 2월 한국전산원 선임연구원. 1993년 3월~현재 중앙대학교 정보시스템학과 교수. 2004년 1월~2005년 1월 Indiana University Bloomington, School of Informatics 방문 교수. 관심분야는 모바일 컴퓨팅, 이동통신 서비스 모델링, 데이터베이스