

이동 방송 환경에서 제어 정보를 줄이기 위한 데이터 그룹화 기법

한부형* 정성권* 조유근
서울대학교 컴퓨터공학과 † (주) 인포모아
bhhan@ssrnet.snu.ac.kr sung@dandelion.snu.ac.kr cho@ssrnet.snu.ac.kr

Data Grouping for Reducing Control Information in Mobile Broadcasting Environments

Boohyung Han* Sung Kwon Chung† Yookun Cho
Dept. of Computer Engineering, Seoul National University, † Infomore, Inc.

요 약

이동 방송 환경(mobile broadcasting environment)은 정보를 가지고 있는 서버가 다수의 이동 클라이언트에게 모든 데이터를 주기적으로 방송하면 클라이언트가 자신이 원하는 데이터가 방송 채널에 나타날 때 이를 검색하는 환경을 말한다. 이 때 클라이언트는 일관적이면서 가장 최근에 갱신된 데이터를 필요로 하기 때문에, 병행성 제어 기법을 사용하여 이러한 요구 조건을 충족시켜야 한다. 이동 방송 환경에서는 클라이언트가 여러 가지 제약점을 가지고 있기 때문에 서버가 클라이언트의 일관성 검사를 위한 제어 정보를 데이터와 함께 방송해야 한다. 그러나, 매 방송 주기마다 방송해야 하는 제어 정보의 양이 크기 때문에 일관성 검사에 영향을 주지 않는 범위에서 이를 줄일 필요가 있다. 본 논문에서는 제어 정보의 양을 줄이기 위하여 데이터를 그룹화(grouping)하는 기법을 제시하고, 시뮬레이션을 통해 데이터 그룹화가 클라이언트의 트랜잭션 중단(abort)에 미치는 영향을 알아본다. 서로 다른 데이터들을 하나의 그룹으로 묶으면 클라이언트가 일관성을 검사할 때 트랜잭션이 불필요하게 중단되는 경우가 더 많아질 수 있다. 그러나, 실험 결과 전체 데이터 수의 20% 정도 크기로 데이터를 그룹화하면 불필요하게 트랜잭션이 중단되는 경우가 거의 없으면서 제어 정보의 크기를 크게 줄일 수 있다는 사실을 알 수 있었다.

1. 서론

이동 방송 환경은 정보를 가지고 있는 서버가 다수의 이동 클라이언트에게 모든 데이터를 주기적으로 방송하면 클라이언트가 자신이 원하는 데이터가 방송 채널에 나타날 때 이를 검색하는 환경을 말한다[1]. 이 방법은 클라이언트가 서버에게 특별한 요청을 하지 않고도 정보를 얻을 수 있다는 장점을 가진다. 특히, 이동 클라이언트는 여러 가지 자원이 제한되어 있기 때문에 클라이언트가 서버에게 데이터를 직접 요청하는 방법보다 비용을 줄일 수 있다. 이 방법을 사용하는 응용은 경매, 주식 거래, 스포츠 관람 티켓, 전자 신문, 메일 검색, 교통 관리 시스템 등이 있다.

이동 방송 환경에서 클라이언트는 필요한 정보를 검색하기 위하여 읽기 전용 트랜잭션(read-only transaction)을 수행하고, 서버는 갱신 트랜잭션(update transaction)을 통해 자신의 데이터베이스의 내용을 계속적으로 갱신한다. 읽기 전용 트랜잭션은 자신이 읽는 데이터와 서버의 갱신 트랜잭션이 갱신하는 데이터가 일관적이어야 한다는 요구조건과 자신이 읽은 데이터가 갱신된 시점이 클라이언트의 제약 조건을 만족해야 한다는 요구조건을 가진다. 전자의 조건을 일관성(consistency)이라고 후자의 조건을 현행성(currency)이라 한다[2].

이동 방송 환경에서 이 두 가지 요구조건을 보장하기 위한 병행성 제어 기법(concurrency control mechanism)이 많이 연구되어 왔다. 그 중에서도 [3]과 [4]에서는 클라이언트가 일관적인 데이터를 읽을 수 있도록 하기 위한 제어 정보를 서버가 데이터와 함께 방송하는 방법을 사용하였다. 이 두 기법에서는 제어 정보를 서버가 계산하여 방송함으로써 자원이 제한되어

있는 이동 클라이언트가 부가적인 비용을 들이지 않고도 일관성을 검사할 수 있다는 장점을 가진다. 그러나, 제어 정보는 $n \times n$ 크기의 행렬(n 은 전체 데이터의 수)로 표현되어 매 방송 주기마다 전부 방송되어야 하기 때문에 네트워크 통신량이 매우 커질 수 있다. 따라서, 일관성 검사에 영향을 주지 않는 범위에서 제어 정보의 크기를 줄일 필요가 있다.

본 논문에서는 제어 정보를 생성할 때 접근 확률이 낮은 데이터들을 하나의 그룹으로 묶어 동일한 데이터인 것처럼 인식하는 방법을 제시하고, 이 방법을 사용하여 제어 정보를 줄이면 읽기 전용 트랜잭션이 불필요하게 중단되는 경우가 얼마나 발생하는지 알아본다. 본 논문은 다음과 같이 구성된다. 먼저 2절에서는 이전에 수행되었던 관련 연구들을 살펴보고, 3절에서는 제어 정보를 줄이기 위하여 데이터들을 그룹화하는 방법을 설명한다. 4절에서는 데이터 그룹화가 트랜잭션의 중단에 미치는 영향을 알아보고, 마지막으로 5절에서 결론을 맺는다.

2. 이전 연구들

그림 1은 네 개의 데이터 a, b, c, d 를 가지고 있는 간단한 이동 방송 환경의 예를 보인다. UT_i, UT_m, UT_n 은 서버에서 수행되는 갱신 트랜잭션을 나타내고, RT_i 와 RT_n 은 클라이언트에서 수행되는 읽기 전용 트랜잭션을 나타낸다. 또한 $R(a)$ 는 데이터 a 에 대한 읽기를, $W(a)$ 는 데이터 a 에 대한 쓰기를, $B(a)$ 는 데이터 a 에 대한 방송을 각각 나타낸다. 이 예에서 서버는 세 개의 갱신 트랜잭션을 수행하며 주기적으로 네 개의 데이터를 방송하고, 클라이언트는 자신이 원하는 데이터가 방송될 때 읽기 전용 트랜잭션을 수행하여 이를 읽는다.

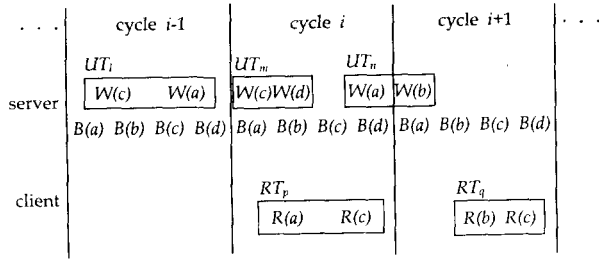


그림 1. 이동 방송 환경

이 때 갱신 트랜잭션과 읽기 전용 트랜잭션간에는 일관성 요구 조건이 만족되어야 하는데, 이동 방송 환경에서는 가장 완화된 일관성인 갱신 일관성(update consistency)이 적합하다고 알려져 있다[3]. 갱신 일관성을 만족하려면 각 읽기 전용 트랜잭션은 자신이 직접적으로나 간접적으로 읽은 데이터를 갱신한 트랜잭션에 대해서만 일관적이어야 한다. 또한 응용에 따라 다양한 형태의 현행성을 요구하는데, 경제나 주식 거래와 같은 응용에서는 정확한 결과를 얻기 위해서 읽기 전용 트랜잭션이 항상 가장 최근에 갱신된 값을 읽을 것을 요구한다. 이를 최신 현행성(latest-bound currency)이라 한다[2].

Shanmugasundaram 등은 갱신 일관성을 보장할 수 있는 병행성 제어 기법을 제안하고 평가하였다[3]. 이 기법에서 서버는 각 방송 주기마다 k 주기가 시작하기 전에 마지막으로 갱신된 값과 함께 제어 정보 행렬을 방송한다. 이 기법을 사용하면 갱신 일관성을 보장할 수 있지만, 각 방송 주기가 시작한 이후에 갱신된 값을 현재의 방송 주기에서 방송할 수 없다는 단점을 가진다. 이는 일관성을 검사하는 것이 방송 주기 단위로 이루어져서 한 방송 주기가 시작하고 난 후에는 새로 갱신된 값을 방송할 수 없기 때문이다. 그림 1로 설명하면, 서버가 방송 주기 i 에서 UT_m 이 갱신된 데이터 c 를 방송하지 못하고 UT_i 이 갱신된 값을 방송해야 한다. 따라서 RT_p 가 읽은 데이터 c 는 가장 최근에 갱신된 값이 아니며, 이러한 읽기를 실패된 읽기(stale read)라 한다.

[4]에서는 이를 개선하기 위하여 "가상 방송 주기"라는 개념을 도입해 제어 정보 행렬을 계산함으로써 실패된 읽기가 발생하지 않도록 하였다. 가상 방송 주기란 각 데이터 i 가 방송되는 주기를 말한다. 따라서 데이터 i 에 대한 가상 방송 주기 번호는 i 가 방송될 때마다 1씩 증가한다. 이 기법에서 제어 정보 행렬의 임의의 항 $C(i,j)$ 는 데이터 j 의 최신 값에 영향을 주면서 동시에 데이터 i 를 마지막으로 갱신한 트랜잭션이 종료된 가상 방송 주기 번호로 설정된다. 그러면 클라이언트가 수행되는 읽기 전용 트랜잭션은 데이터 j 를 읽기 전에 자신이 이전에 데이터 i 를 읽었던 가상 방송 주기 번호와 $C(i,j)$ 값을 비교하여 일관성 검사를 한다.

3. 데이터 그룹화

본 논문에서는 서버와 클라이언트가 데이터를 접근할 확률이 Zipf 분포[5][6]를 따른다고 가정한다. Zipf 분포는 보통 비규칙적인 접근 패턴을 모델할 때 사용된다. 모든 데이터에 1부터 n 까지 번호를 부여한다고 할 때, Zipf 분포에서 데이터 i 를 접근할 확률은 $(1/i)^\theta$ 에 비례하며, $\theta (0 \leq \theta < 1)$ 가 증가함에 따라 접근 패턴도 비대칭적이 된다. 예를 들어, $\theta = 0.95$ 이면 접근 패턴이 매우 비대칭적인 경우이고, $\theta = 0$ 이면 모든 데이터가 동일한 확률을 가지고 접근되는 경우를 의미한다.

서버와 클라이언트가 데이터를 비대칭적으로 접근하는 상황을 가정한다면, 일부 데이터들의 접근 확률은 매우 높지만 반대로 나머지 데이터들의 접근 확률은 매우 낮게 된다. 따라서,

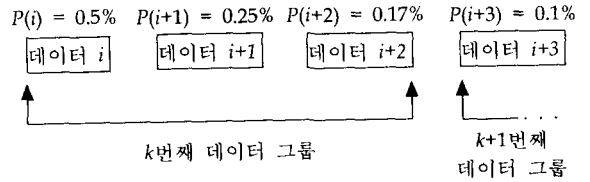


그림 2. 데이터 그룹화의 예 (기준 확률 = 1.0%)

제어 정보를 생성할 때 상대적으로 접근 확률이 낮은 데이터들을 그룹화한다면 제어 정보 행렬의 크기를 줄일 수 있다. 이를 위해 본 논문에서는 그룹화의 기준이 되는 기준 확률을 정의하고, 이를 기준으로 데이터를 그룹화한다. 만약 임의의 데이터 i 의 접근 확률이 기준 확률보다 크다면, i 는 혼자서 하나의 그룹을 형성한다. 반대로 데이터 i 의 접근 확률이 기준 확률보다 작다면, 접근 확률이 기준 확률보다 작은 다른 데이터들과 함께 하나의 그룹으로 그룹화된다.

그림 2는 데이터 그룹화의 예를 보여준다. 기준 확률이 1.0%이고, 데이터 i 의 접근 확률이 0.5%라면 기준 확률보다 커지지 않을 때까지 다음 데이터들의 접근 확률을 더한다. 따라서 데이터 $i, i+1, i+2$ 의 접근 확률을 모두 더하면 0.92%이고, 이는 기준 확률보다 작기 때문에 하나의 데이터 그룹이 된다. 만약 데이터 $i+3$ 도 이 그룹에 포함하면, 접근 확률의 합이 기준 확률을 넘어가게 되기 때문에 허용되지 않는다.

4. 실험 및 결과

본 논문에서는 제어 정보를 줄이기 위해 데이터를 그룹화하는 것이 읽기 전용 트랜잭션이 중단되는 수에 미치는 영향을 알아보기 위하여 시뮬레이션을 수행한다. 먼저 시뮬레이션한 병행성 제어 기법은 Shanmugasundaram의 기법[3]과 [4]에서 제시한 기법이다. 이 절에서는 전자를 t -인원 현행성 기법(t -vintage), 후자를 최신 현행성 기법(latest-bound)으로 각각 부른다. 본 논문에서 구현한 시뮬레이터는 하나의 서버와 클라이언트를 가진다. 이동 방송 환경에서는 클라이언트의 수가 전체 성능에 전혀 영향을 주지 않기 때문에 하나의 클라이언트만 실험한다. 표 1은 실험에 사용된 인자와 내정(default) 값이다. 모든 트랜잭션과 데이터 접근은 지수 분포(exponential distribution)를 따라 도착한다고 가정한다. 오프셋 인자는 서버와 클라이언트의 접근 패턴간의 차이를 나타낸다. 오프셋이 0이면 서버와 클라이언트의 접근 분포가 완전히 일치하는 경우이고, 오프셋이 k 이면 클라이언트가 집중적으로 접근하는 데이터를 k 만큼 이동시켜 서버가 이를 덜 접근하는 경우이다. 본

표 1. 시뮬레이션 인자

인자	내정 값
총 데이터의 수	1000
총 갱신 트랜잭션의 수	10000
갱신 트랜잭션의 평균 도착간 시간	100
갱신 트랜잭션 당 데이터 접근 수	10
갱신 트랜잭션 당 데이터 갱신 수	2
읽기 전용 트랜잭션의 평균 도착간 시간	200
읽기 전용 트랜잭션 당 데이터 읽기 수	4
서버 읽기의 평균 도착간 시간	10
서버 읽기의 범위	1000
서버 갱신의 평균 도착간 시간	10
서버 갱신의 범위	500
클라이언트 읽기의 평균 도착간 시간	100
클라이언트 읽기의 범위	250
Zipf 분포에서의 θ	0.95
서버와 클라이언트 접근간의 오프셋	0

실험에서는 비용 척도로서 읽기 전용 트랜잭션이 갱신 일관성을 만족하지 못하여 중단되는 수를 측정한다.

그림 3 (a)는 두 기법에서 $\theta = 0.95$ 이고 오프셋이 0일 때 중단된 읽기 전용 트랜잭션의 수를 보여준다. 그래프에서 x축은 데이터 그룹의 수, y축은 중단된 트랜잭션의 수를 나타낸다. 데이터 그룹의 수가 1000개이면 데이터 그룹화를 전혀 하지 않은 경우를 의미하고, 데이터 그룹의 수가 작아질수록 데이터 그룹화의 기준 확률을 높인 경우를 의미한다. 두 기법에서 중단된 트랜잭션의 수는 전체적으로 최신 현행성 기법이 t-이전 현행성 기법보다 약간 작다. 이는 최신 현행성 기법에서 클라이언트가 가장 최근에 갱신된 데이터를 읽음으로써 직렬화 그래프(serialization graph)의 순환(cycle)을 제거할 수 있기 때문이다[4]. 한편, 데이터를 그룹화하면 그렇지 않은 경우보다 더 많은 트랜잭션이 중단된다. 특히 데이터 그룹의 수가 100개보다 작아지면 중단된 트랜잭션의 수가 기하급수적으로 늘어나게 되고, 데이터 그룹의 수가 10개 정도가 되면 최신 현행성 기법에서 중단된 트랜잭션의 수가 t-이전 현행성 기법보다 오히려 많아진다. 그러나, 데이터 그룹의 수가 500개에서 1000개 정도일 때까지는 중단된 트랜잭션의 수가 동일하고, 200개 정도일 때도 중단된 트랜잭션의 수가 거의 늘어나지 않는다.

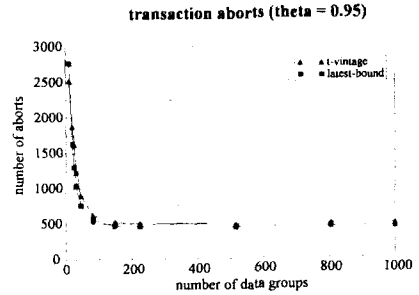
그림 3 (b)와 그림 3 (c)는 데이터의 접근 패턴을 변화시킬 때 데이터 그룹화의 영향을 알아보기 위하여 각각 θ 를 0.95에서 0.05까지 변화시킬 때와 오프셋을 0에서 100까지 변화시킬 때 최신 현행성 기법에서 중단된 트랜잭션의 수를 보여준다. θ 와 오프셋의 값에 상관없이 t-이전 현행성 기법은 최신 현행성 기법과 비슷한 형태로 나타나기 때문에 그래프를 간단하게 나타내기 위해 생략하였다. 그림 3 (b)에서 θ 가 작아질수록 데이터의 접근 패턴이 대칭적이기 때문에 중단되는 트랜잭션의 수는 작아지지만, 데이터 그룹화가 트랜잭션의 중단에 미치는 영향은 점점 커지기 때문에 중단된 트랜잭션의 수가 급격히 늘어나는 지점이 약간씩 커진다. 또한 데이터 그룹의 수가 100개 이하면 θ 가 작아질수록 오히려 중단된 트랜잭션의 수가 더 많아진다. 그림 3 (c)에서 오프셋이 작아질수록 서버와 클라이언트의 접근 분포가 많이 겹치기 때문에 중단되는 트랜잭션의 수가 커지며, 중단된 트랜잭션의 수는 그림 3 (b)와 비슷한 분포를 보인다.

5. 결론

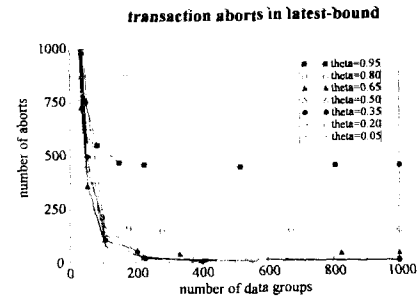
본 논문에서는 이동 방송 환경에서 네트워크 통신량을 줄이기 위하여, 제어 정보를 생성할 때 데이터의 접근 확률을 기반으로 데이터를 그룹화하는 기법을 제시하였다. 또한, 이 기법을 사용할 때 읽기 전용 트랜잭션이 중단되는 수가 얼마나 달라지는지를 살펴보았다. 실험 결과, 데이터의 접근 패턴에 따라 약간씩 차이를 보이기는 하지만 총 데이터 수의 20% 정도로 데이터를 그룹화하면 트랜잭션이 불필요하게 중단되는 경우를 거의 유발하지 않으면서 제어 정보를 전송하는데 드는 네트워크 통신량을 원래의 4%(= 20%×20%) 정도로 크게 줄일 수 있다는 사실을 알 수 있었다.

6. 참고 문헌

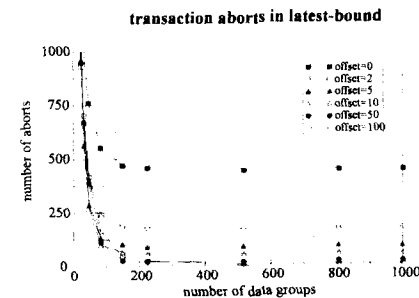
[1] T. Imielinski and B. R. Badrinath, "Mobile Wireless Computing: Challenges in Data Management," *Communications of the ACM*, Vol. 37, No. 10, pp. 18-28, October 1994.
 [2] H. Garcia-Molina and G. Wiederhold, "Read-Only Transactions in a Distributed Database," *ACM Transactions on Database Systems*, Vol. 7, No. 2, pp.209-234, June 1982.
 [3] J. Shanmugasundaram, A. Nithrakashyap, R. Sivasankaran, and K. Ramamritham, "Efficient



(a) $\theta = 0.95$ 이고, 오프셋이 0일 때



(b) θ 를 변경할 때



(c) 오프셋을 변경할 때

그림 3. 중단된 트랜잭션의 수

Concurrency Control for Broadcast Environments," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 85-96, June 1999.

[4] 한부형, 정성권, 조유근, "이동 방송 환경에서 최신 현행성을 보장하는 병행성 제어 기법", *정보과학회논문지* 제출·심사 중
 [5] J. Gray, P. Sundaresan, S. Englert, K. Baclawski, and P. J. Weinberger, "Quickly Generating Billion-Record Synthetic Databases," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 243-252, May 1994.
 [6] S. Acharya, R. Alonso, M. Franklin, and S. Zdonik, "Broadcast Disks: Data Management for Asymmetric Communication Environments," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 199-210, May 1995.