

이동 데이터베이스 시스템에서 캐쉬 일관성 유지를 위한 이진 트리 기반의 무효화 보고*

임상민, 강현철
중앙대학교 컴퓨터공학과
e-mail:{smlim, hckang}@rose.cse.cau.ac.kr

Binary Tree-based Invalidation Reporting for Cache Consistency Maintenance in a Mobile Database System

Sangmin Lim, Hyunchul Kang
Dept. of Computer Science and Engineering, Chung-Ang University

요 약

이동 통신 기술의 발전으로, 이동컴퓨팅 환경에서의 데이터 서비스에 대한 수요가 증가하고 있다. 이동 호스트 내에 캐쉬가 존재할 경우, 대역폭의 절약 및 빠른 응답시간을 제공할 수 있지만, 캐쉬 일관성을 유지해야하는 부담이 생긴다. 한 셀 내에 존재하는 수 많은 이동 호스트들의 캐쉬 일관성 유지를 위해서 서버(MSS)가 캐쉬 무효화 보고를 일정시간마다 주기적으로 방송하는 방법은 효과적일 수 있다. 기존에 제시된 무효화 보고에서는 갱신된 데이터의 식별자가 사용되지만, 좀 더 적은 양의 대역폭을 소모하는 효율적인 무효화 보고가 되기 위해서는 데이터들을 그룹핑할 수 있는 방안이 연구되어야 한다. 본 논문에서는 서버의 데이터베이스를 데이터의 식별자 값을 기준으로 여러 구역으로 나누었을 때, 데이터의 갱신이 데이터베이스 내 인근 구역에서 많이 발생할 경우, 이진 트리를 기반으로 하여 효율적인 무효화 보고를 생성하는 방법을 제안하고 그 성능을 평가한다.

1. 서 론

이동 통신 기술의 급속한 발전으로, 이동 통신 환경에서 데이터 서비스가 제공되고 있다. 최근 휴대전화를 이용한 증권, 뉴스, 기상, 스포츠 등의 데이터 서비스 실시는, 앞으로 계속 확대 제공될 이동 데이터 서비스의 예이다.

이동컴퓨팅 환경의 가장 큰 특성은, 기존의 유선 네트워크에 연결된 호스트 컴퓨터로서 무선 통신 능력을 가지고 있는 MSS(Mobile Support Station)와 이동 호스트 간에 네트워크 연결상태가 불안정하다는 것과 의도적 혹은 비의도적인 접속단절이 빈번하다는 것이다. 또한, 유선환경보다 낮은 대역폭은 데이터 서비스를 하는 데 있어서 단점이 될 수 있다. 이런 문제를 해결하기 위한 대표적인 방법으로 방송과 캐쉬기법이 있다[1][2][3].

이동 호스트 내에서 캐쉬의 이용은 질의 응답시간의 단축, 대역폭의 절약, 그로 인한 배터리의 절약을 가져올 수 있지만, 캐쉬의 일관성을 유지해야 하는 부담이 생긴다. 따라서, 캐쉬 일관성 유지를 위한 효율적인 기법이 필요

하다.

MSS가 이동 호스트들의 무선 접속을 관장하는 지역을 셀이라하며, MSS가 한 셀 내에 주기적인 무효화 보고(invalidation report, 이하 IR)를 방송하여 캐쉬 일관성을 유지하는 방법은 비의도적인 짧은 단절이 자주 발생하는 이동 호스트가 많을 경우 효율적일 수 있다[2]. 하지만, 오랜 시간동안 MSS와의 단절로 인하여, IR만으로 캐쉬의 유효성을 확인하지 못할 경우, 이동 호스트 각각이 MSS 측으로 유효성 여부 확인을 요청하여 캐쉬를 유효화할 수도 있다[5].

본 논문에서는 IR을 방송하여 캐쉬 일관성을 유지하는 경우, IR의 양을 줄여서 대역폭을 절약할 수 있는 방안을 제시한다.

본 논문의 구성은 다음과 같다. 2절에서는 이진 트리 기반의 캐쉬 IR을 제안하고, 3절에서는 성능평가를 수행한 결과를 기술한다. 그리고 4절에서 결론을 내린다.

2. 이진 트리 기반의 캐쉬 무효화 보고

* 본 논문은 정통부의 정보통신우수시범학교 지원 사업에 의한 것임.

2.1 개요

이동컴퓨팅 환경에서 캐쉬 일관성 유지를 위한 방법으로 정기적인 IR의 방송을 이용할 수 있다[2][4]. IR의 내용 구성은 다음 2가지 방법이 제안되어 있다.

(1) 리스트 기반의 IR

리스트 기반의 IR은 최근에 갱신된 데이터의 식별자와 갱신된 시점의 타임스탬프 쌍의 리스트로 구성된다[2]. 이때, ‘최근’이란, MSS에서 시스템 파라미터로 설정된 시간 동안을 의미하며, 보통 $w(\geq 1)$ 개의 IR 구간으로 정한다. 이동 호스트는 갱신된 데이터의 식별자와 타임스탬프로 자신의 캐쉬 내에 데이터들의 타임스탬프를 비교하여, 캐쉬를 유효화한다.

(2) 이진 비트열(bit sequences) 기반의 IR

이진 비트열 기반의 IR은 이진 비트열과 타임스탬프의 집합으로 구성된다[4]. 각 비트는 특정 데이터의 갱신 여부를 알려 주게 되므로, 이동 호스트는 각 비트 정보를 이용하여 캐쉬를 유효화한다.

리스트 기반의 IR은 직접 데이터의 식별자를 방송하기 때문에, 갱신된 데이터의 개수가 적은 경우에는 효율적일 수 있지만, 갱신된 데이터의 개수가 많은 경우에는 많은 양의 대역폭 소모를 가져올 수 있다. 이진 비트열 기반의 IR은 항상 같은 양의 IR을 방송하지만, 갱신이 적게 발생할 경우 불필요한 데이터들의 정보(불필요한 비트 정보)들을 방송하게 되는 단점을 가진다.

이에 반하여, 리스트 기반의 IR에서처럼 데이터의 식별자를 방송하되, 갱신된 데이터 중 그 식별자들이 서로 근접한 값을 갖는 데이터들에 대해서는 그룹을 형성하여 방송하는 기법은 효율적일 수 있다.

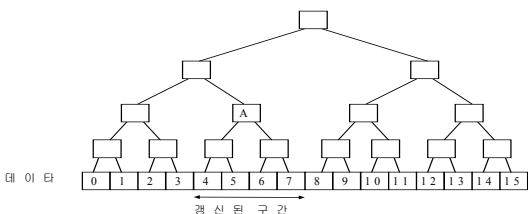


그림 1 이진 트리를 이용한 그룹핑을 통한 무효화 보고

본 절에서는 이진 트리에 기반한 새로운 형태의 효율적인 IR 구성 방법을 제안한다. 이진 트리 기반의 IR은 이진 트리를 이용하여 갱신된 데이터들을 그룹화한 뒤, 이

그룹화된 정보로 IR을 구성한다. 그림 1에서 보는 것처럼, 데이터베이스 내의 갱신된 데이터의 식별자가 4, 5, 6, 7일 경우, 이미 제안된 상기 두 기법에서는 4, 5, 6, 7번 데이터 모두 IR의 내용으로 포함되었다. 하지만, 이진 트리 기반의 IR은 A라고 표시된 노드만이 IR의 내용으로 포함되어 IR의 크기가 줄어들 뿐만 아니라, 이전의 IR과 동일한 정보를 전달하게 된다.

2.2 자료구조

데이터 갱신이 IR 직전에 발생한다는 가정 하에 IR의 형태는 표 1과 같이 정의된다.

종 류	형 태
리스트 기반의 IR	< Flag, {TS ₁ , data_ID, data_ID, ...}, ..., {TS _w , data_ID, data_ID, ...} >
이진 트리 기반의 IR	< Flag, {TS ₁ , {level, node_ID}, ... }, ..., {TS _w , {level, node_ID}, ... } >

표 1 무효화 보고의 형태

리스트 기반의 IR에서 Flag는 현재 방송하는 정보가 어떤 용도로 사용될 지를 알려주는 역할을 한다. 예를 들면, 캐쉬 일관성 유지가 아닌, 다른 데이터 방송 서비스도 방송될 수 있으므로, 이동 호스트가 현재 방송된 데이터가 어떤 용도인지를 판별하기 위한 것이다. TS_i($i=1, 2, \dots, w$, 단 $w \geq 1$)는 데이터들이 현재 시점에서 i 번째 이전 구간에 갱신되었음을 알려주는 타임스탬프 값을 나타내며, data_ID는 특정 데이터를 구별할 수 있는 식별자이다.

이진 트리 기반의 IR에서 level 값은 루트 노드에서는 0이며, 레벨이 증가하면서 1씩 증가하는 값이다. 그리고, node_ID는 그 노드를 루트로 하는 서브 트리의 리프 노드에 해당되는 데이터들이 모두 갱신되었음을 나타내기 위한 것이다. 예를 들어, 그림 1과 같이 데이터가 16개인 상황에서 리스트 기반의 IR 형태가 <0, {100, {4,5,6,7}}> 이었다면, 이진 트리 기반의 IR은 <0, {100, {{2, A}}}>의 형태를 갖는다.

2.3 서버(MSS)의 IR 생성과정

앞 절에서 기술한 자료구조를 토대로, 서버의 IR 생성 과정을 기술하면 다음과 같다.

- ① 데이터베이스 내의 데이터의 개수가 n 일 때,

리프 노드의 개수가 n 인 이진 트리를 생성

- ② 갱신이 발생한 데이터에 대응되는 리프 노드에 갱신이 되었음을 표시
- ③ 생성된 이진 트리의 가장 하위 레벨(리프노드들)부터 가장 상위 레벨(루트노드)순으로 스캔하면서 다음을 반복

현재 스캔하는 레벨에서, 부모 노드가 같은 인접한 노드가 모두 갱신이 일어났다면, 부모 노드에 갱신 표시를 하고, 자식 노드의 갱신 표시를 없앴

- ④ 가장 상위 레벨(루트 노드)부터 가장 하위 레벨(리프 노드들)순으로 스캔하면서, 갱신표시가 있는 것을 IR에 기입
- ⑤ IR을 방송

2.4 이동 호스트의 캐쉬 무효화 과정

이진 트리 기반의 IR을 수신한 이동 호스트는 아래와 같은 과정으로 캐쉬 유효화를 수행한다.

- ① IR을 수신
- ② 이진 트리 형태의 IR로부터 level정보와 node_ID를 이용하여 갱신된 데이터들의 식별자를 찾아냄
- ③ 갱신된 모든 데이터들을 캐쉬영역으로부터 삭제

파라미터	설명	기본값 (단위)
databaseSize	데이터베이스 내의 데이터의 개수	1000 (개)
tranArrRate	트랜잭션의 발생 확률	0.01 (개/초)
updatePerTrans	트랜잭션 당 갱신되는 데이터의 개수	5(개)
hotUpdateSet	전체 데이터베이스의 크기 대비 갱신이 자주 발생하는 구역의 크기 비율	0.01
hotUpdateProb	hotUpdateSet을 갱신할 확률	0.9
w	IR이 포함하는 구간의 개수	1
broadcastInterval	IR이 방송되는 주기	20(초)

표 2 성능 실험 파라미터

3. 성능 평가

본 절에서는 리스트 기반의 IR과 이진 트리 기반의 IR

의 성능을 비교 평가한다.

3.1 성능 평가 척도 및 파라미터

성능 평가 척도는 방송되는 IR의 양, 곧 대역폭 소모량으로 정한다. 성능실험 파라미터는 표 2와 같다.

3.2 실험결과

본 논문이 제안하는 이진 트리 기반의 IR은 데이터베이스에서 자주 갱신되는 구역의 범위와 갱신 회수의 정도에 따라서 성능에 영향을 미치게 된다. 따라서, hotUpdateSet의 크기 변화에 따른 실험과 tranArrRate변화에 따른 실험을 수행하였다. 이 실험을 위해서 데이터베이스의 크기는 1000개의 데이터로 하고, hotUpdateSet은 0.01로, tranArrRate는 0.01로, 또한 w는 1로 기본 설정하였다.

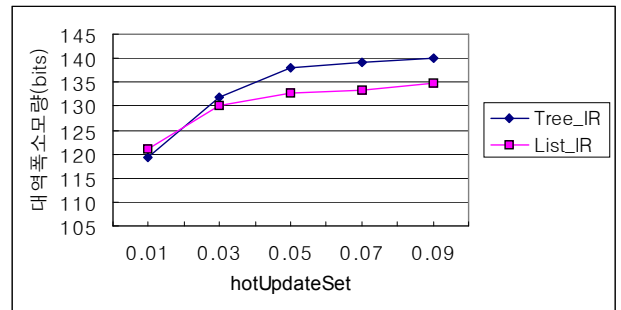


그림 2 자주 갱신되는 구역의 크기 변화에 따른 대역폭의 소모량 (databaseSize=1000, tranArrRate=0.01, w=1)

그림 2에서 Tree_IR은 이진 트리 기반의 IR을 나타내며, List_IR은 리스트 기반의 IR을 나타낸다. hotUpdateSet의 값이 작을 경우, 이진 트리 기반의 IR을 방송하는 것이 리스트 기반의 IR을 방송하는 것보다 더 적은 양의 대역폭을 소모하였다. 그러나 hotUpdateSet이 클수록, 곧 갱신이 자주 일어나는 구역의 범위가 넓어질수록, 리스트 기반의 IR을 방송할 경우 더 적은 양의 대역폭을 소모하였다. 이와 같은 결과는 이진 트리 기반의 IR이 리프 노드가 아닌 리프 노드의 조상(ancestor) 노드들을 보내야만 더 좋은 성능을 나타내게 되는데, 그림 2에서는 자주 갱신되는 구역의 범위가 커짐에 따라서, 조상 노드를 보낼 가능성이 점차 적어지게 된다. 이로 인해, 각 노드마다 노드들의 레벨을 나타내는 정보가 오버헤드로 작용하여 오히려 제안한 기법이 더욱 좋지 않은 성능을 나타내게 된다. 하지만, hotUpdateSet이 계속해서 증가하더라도 성능의 차이가 계속 증가하지 않게 되는데, 그 이유는, level 정보의 오버헤드만큼 이상으로는 성능 차이가 나지 않기 때문이다.

그림 3은, tranArrRate가 증가할수록 이진 트리 기반의

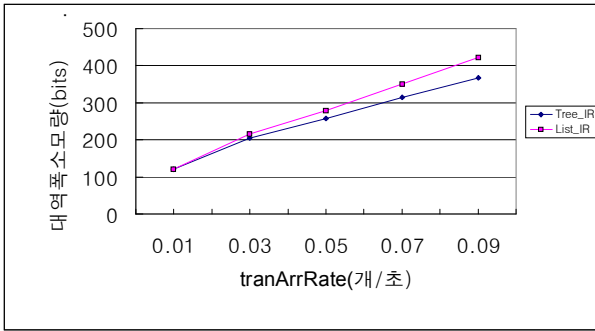


그림 3 트랜잭션 도착률 변화에 따른 대역폭의 소모량

(databaseSize=1000, hotUpdateSet=0.01, w=1)

IR을 방송할 경우가 더욱 좋은 성능을 나타냄을 보여준다. 이는 갱신이 자주 일어나게 되면 이웃하는 리프 노드들이 갱신되어 조상 노드로 대표될 가능성이 높아지게 되기 때문이다.

4. 결론

이동 컴퓨팅 환경에서 다양한 데이터 서비스를 지원하기 위해서는 무선 대역폭의 소모를 가져올 수 밖에 없다. 따라서, 이동 호스트에서 캐쉬의 이용은 무선 대역폭의 소모를 줄일 수 있는 효율적인 방법이 될 수 있다. 이동 호스트 내에 캐쉬가 존재하게 되면, 대역폭의 절약 및 빠른 응답시간을 제공할 수 있지만, 캐쉬 일관성을 유지하는 부담이 생긴다. 따라서, 캐쉬 일관성을 유지하기 위해서 한 셀 내에서 서버(MSS)가 무효화 보고(IR)를 일정시간마다 주기적으로 방송하는 기법은 효율적일 수 있다.

본 논문에서는 갱신된 데이터들을 이진 트리를 기반으로 그룹핑하여 생성되는 이진 트리 기반의 IR을 제안하였다. 데이터베이스에서 갱신이 자주 일어나는 구역의 범위가 넓어지게 되면, 이진 트리 기반의 IR을 방송하는 것이 리스트 기반의 IR을 방송하는 것보다 트리의 레벨 정보에 의한 오버헤드로 인하여, 좋지 못한 성능을 나타내었다. 하지만, 갱신이 자주 일어날 경우에는, 갱신이 인접한 구역에서 일어날 확률이 높아지므로, 제안한 IR이 좋은 성능을 나타내었다.

참고문헌

[1] S. Acharya et al., "Broadcast Disks : Data Management for Asymmetric Communication Environments," *Proc. ACM SIGMOD Int'l Conf. on Management of Data*, pp. 199-210, 1995.

[2] D. Barbara and T. Imielinski, "Sleepers and Workaholics : Caching Strategies in Mobile

Environments," *Proc. ACM SIGMOD Int'l Conf. on Management of Data*, pp. 1-12, 1994.

[3] T. Imielinski et al., "Energy Efficient Indexing On Air," *Proc. ACM SIGMOD Int'l Conf. on Management of Data*, pp. 25-36, 1994.

[4] J. Jing et al., "Bit-Sequences: An Adaptive Cache Invalidation Method in Mobile Client/Server Environments," *ACM/Baltzer Mobile Networks and Applications*, Vol. 2, No. II, 1997.

[5] K. Wu et al., "Energy-Efficient Caching for Wireless Mobile Computing," *Proc. IEEE Int'l Conf. on Data Engineering.*, pp. 336-343, 1996.