

다중 무선 방송채널에서의 효과적인 모바일 트랜잭션 처리 기법

정호련 박성욱[○] 정성원

서강대학교 컴퓨터학과

queenking@sogang.ac.kr psw0405@mclab.sogang.ac.kr jungsung@sogang.ac.kr

An Efficient mobile transaction processing scheme over multiple wireless broadcast channels

Ho-Ryun Jeong Sung-wook Park[○] Sungwon Jung

Department of Computer Science, Sogang University

1. 서론

최근 휴대폰이나 PDA와 같은 컴퓨터 하드웨어와 블루투스와 같은 무선 네트워크 기술의 급격한 발달은 모바일 컴퓨팅을 점차 가능케 하고 있다. 그러나 현존하는 기술들은 물리적 제약 조건으로 인해 좁은 통신 대역폭, 잦은 접속 단절, 배터리 부족 등의 문제를 가진다. 이러한 문제들을 해결하기 위해 데이터를 일정한 주기 동안 반복적으로 전송하는 무선 데이터 방송 기법에 대한 많은 연구가 있었다. 이러한 기법은 클라이언트가 명시적으로 데이터를 요구하지 않아도 원하는 데이터를 수신할 수 있으므로 상향 통신 대역폭을 낭비하지 않고, 서버는 클라이언트의 수에 대한 확장성을 가지는 장점이 있다.

그러나 이러한 환경에서는 많은 사용자들이 동시에 하나의 아이템을 읽거나 갱신하는 상황이 발생하고, 따라서 데이터를 일관성 있게 관리하기 위해서는 트랜잭션 단위의 동시성 제어 기법이 필요하다. 하지만 무선 데이터 방송환경의 특성상 단일채널에서의 방송 사이클을 주기로 하여 일관성을 보장하는 전통적인 트랜잭션 처리 기법을 그대로 적용할 수 없다. 따라서 여러 가지 이유로 멀티채널(Multi-channel) 환경에서의 데이터 방송 기법이 연구 되어 왔다.

그러나 다중채널 방송을 기반으로 트랜잭션을 실시 할 경우 하나의 클라이언트는 하나의 채널만을 들을 수 있는 속성 때문에 기존의 단일채널 트랜잭션 환경에서 사용되어오던 방송 단위의 일관성은 적용되기 어렵다.

본 논문에서는 다중채널 환경에서 트랜잭션 처리를 위한 서버와 클라이언트 측의 동시성 제어기법을 제시해 보도록 하겠다.

2. 본론

2.1 인덱스 채널구조 환경

다중 채널 또한 단일 채널과 마찬가지로 배터리의 소모를 줄이기 위하여 인덱스를 제공해야 한다. 이번 장에서 사용하는 다중 채널 인덱스는 alphabetic Huffman 트리 구조를 이용하여 인덱스만을 위한 전용 채널을 사용한다. 인덱스 전용채널 제공하는 것은 기존의 데이터와 인덱스의 크기가 같아야한다는 가정이나 사용되는 인덱스 트리의 깊이와 같은 수의 채널이 주어지지 않을 경우 본래의 성능을 보이지 못하는 단점을 보완한 다중채널 인덱싱 기법이다. 그러나 우리가 제안하는 방법에서는 기존 논문과는 다르게 인덱스 정보를 하나의 채널에만 인덱스를 할당하도록 한다. 또한 인덱스 노드에서는 데이터의 채널번호, 자식 인덱스 또는 데이터 노드에 대한 정보와 다음 제어정보의 시간 offset이 제공된다.

2.2 다중 무선 채널에서 데이터 채널 방송

서버는 데이터베이스 내의 데이터의 일관성을 유지할 책임을 가진다. 본 논문에서는 모든 채널에서 일관성이 있는 데이터를 일정 시간만큼 반복하는 것과 제어 정보(control information)를 통하여 읽기 트랜잭션의 일관성을 보장하도록 하였다.

다중무선 채널에서 데이터를 일관성 있게 방송하는 방법은 다음과 같다. 데이터는 데이터 채널 중 방송 사이클이 가장 긴 채널의 방송 길이만큼 나머지 채널들을 반복된다. 즉 채널 중 방송 길이가 가장 긴 채널의 방송 사이클이 모든 채널의 일관성 주기가 된다. 이 가장 긴 채널의 방송 사이클의 길이를 일관성 주기(Consistency Cycle)이라고 하자. 방송 사이클이 가장 긴 채널이외의 모든 채널은 일관성 주기만큼 반복되는데 방송 사이클의 모든 데이터가 반복될 필요는 없다. 즉, 공간이 남아 있는 만큼 데이터를 반복한다.

2.3 다중채널 상에서 모바일 트랜잭션의 접근 방법

다중채널 상에서 모바일 트랜잭션의 접근방법은 그림1와 같다.

```

begin
/* Input:
x=data node that needs to be accessed;
Dsize = the size of a data node;
Cnum = data channel number over which data node x is
being broadcast;
OFFSET = the time offset for retrieving the data node x
over data channel Cnum;
BC = the number of data nodes that must be broadcast
over the data channel Cnum before the data node x is
broadcast again;
current_position = the current position;
CCi(1≤i≤k) = the length of broadcast cycle of channel i;
CI_size = the size of Control Information;
CI_num = the number of Control Information listening
client
BP_num = the number of broadcast cycle during
Consistency cycle;
CI_position = time offset listened in index node
CCN = the number of Consistency Cycle after listening CI;
real_CI = time offset of next Control Information;
Output:
position = the current time position of data node x
*/
size_of_data_Channel = BC * Dsize;
period_number = current_index_position/DSize ;
position = Size_of_Data_Channel * period_number + OFFSET;
    
```

```

if (current_position < CI_position) {
    while(current_index_position > position) {
        period_number++;
        position = size_of_Data_Channel * period_number + OFFSET;
    }
} else {
    if (current_position ≥ real_CI) { CCN++; bp_num = 0; }
    position = {CCN * (CCk+CI_size)+1}+CCi*BP_num+OFFSET;
    if(position is duplicated){
        give priority to channel having the longest broadcast cycle;
        while(position is not duplicated){
            position = {CCN * (CCk+CI_size) + 1} + CCi * BP_num
                +OFFSET;
            bp_num++;
        }
    }
}
end
    
```

그림 1. 다중채널 상에서 모바일 트랜잭션의 접근방법

클라이언트 측에서 트랜잭션 수행 중에 제어정보(control information)를 듣게 되었다면 validation을 수행하게 된다. validation 방법은 현재 수행하고 있는 읽기 트랜잭션 중 아직 읽지 못한 데이터의 아이디가 제어 정보 안에 있는 데이터의 아이디와 같은지를 확인한 뒤 만약 제어정보에 필요한 데이터의 아이디와 일치하지 않는다면 트랜잭션을 계속실행하고 만약 일치한다면 트랜잭션을 중지시키고 재실행한다. 제어정보에 있는 데이터는 이전의 일관성 주기 동안에 변경되었다는 의미이므로 이전 일관성 주기와 진행될 일관성 주기 동안 데이터의 일관성이 같지 않다는 것을 의미한다. 따라서 수행은 중지되고 재실행 되어야 한다.

2.4 성능 평가

2.4절에서는 다양한 실험을 통해 본 논문에서 제시하는 다중 채널에서의 읽기 트랜잭션 처리 기법이 효과적임을 보인다. 단일채널에서의 Lee가 제안한 검증을 이용한 낙관적 동시성 제어 기법과 비교를 통해서 본 논문에서 제안하는 방법의 성능을 분석한다. 앞으로는 편의를 위해서 이 논문에서 제안하는 기법을 TPMC(Transaction Processing in Multi Channel), Lee가 제안한 동시성 제어 기법을 FBOCC로 쓰도록 하겠다. TPMC가 선호도를 반영한 다중채널 환경에서의 트랜잭션 처리 기법으로 FBOCC 또한 단일채널에서 선호도를 반영한 Acharya가 제안한 multi-disk 방법으로 방송을 구성하였다. 본 논문에서는 트랜잭션의 응답 시간을 성능 지표로 삼았다.

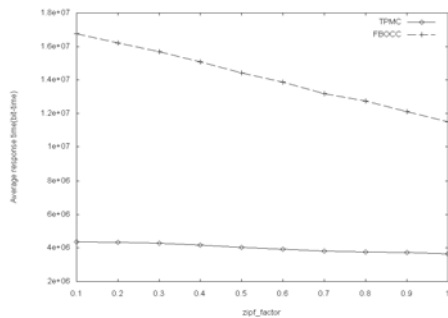


그림 2. 접근 편향성에 따른 평균 응답 시간

Fig.2는 접근 편향성을 나타내는 zipf factor에 따른 TPMC와 FBOCC의 응답시간을 나타낸 것이다. zipf factor가 증가함에 따라 응답시간이 감소하는 것을 확인할 수 있다. TPMC는 기본적으로 다중 채널을 기반으로 하는 것이기 때문에 접근 편향성에 대하여 zipf factor가 커짐에 따라 응답시간이 감소하는 것을 확인할 수 있다. Fig.3은 TPMC와 FBOCC의 서버 트랜잭션 갱신 비율에 따른 평균 응답시간을 비교한 것이다. 그림에서 확인할 수 있는 것과 같이 TPMC와 FBOCC 모두 서버 트랜잭션의 쓰기 비율이 증가함에 따라 평균 응답시간이 증가하는 것을 확인할 수 있다. Fig.4를 통해서 알 수 있듯이 트랜잭션의 데이터 개수가 증가할수록 TPMC의 평균 응답 시간이 FBOCC의 평균 응답 시간보다 좋은 것을 확인할 수

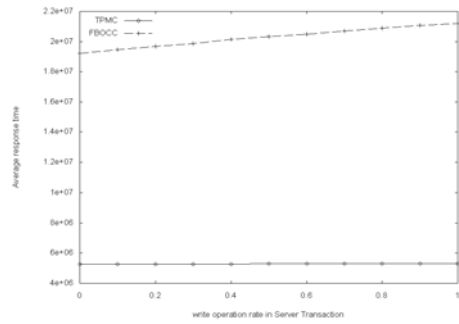


그림3. 서버 트랜잭션 갱신률에 따른 평균 응답 시간

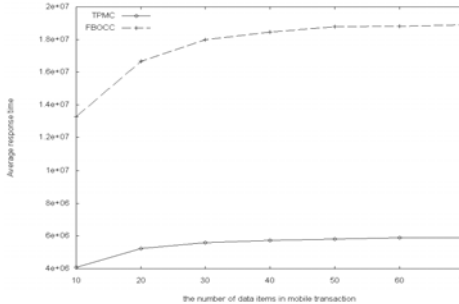


그림4. 모바일 트랜잭션의 데이터 개수에 따른 평균 응답시간

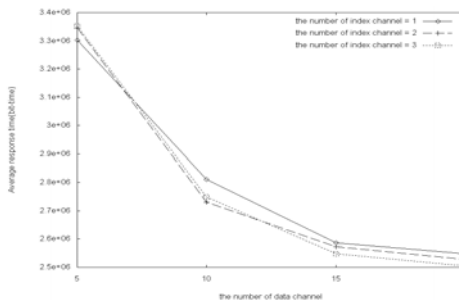


그림5. 인덱스 채널과 데이터 채널의 개수에 따른 평균 응답 시간

있다. 이는 짧은 방송 주기의 TPMC가 보다 긴 방송 주기로 재실행을 실행하는 FBOCC기법보다 재실행을 위한 대기 시간이 짧아지기 때문이다. Fig.5를 살펴보면 우선 인덱스 채널의 개수에 상관없이 데이터 채널의 개수가 증가함에 따라 응답시간이 감소하는 것을 확인할 수 있다. 이는 데이터 채널의 개수가 증가할수록 하나의 데이터 채널에 할당되는 데이터의 개수가 적어지기 때문에 가장 긴 채널의 방송 주기로 결정되는 LBCPC 또한 짧아지기 때문이다.

3. 결론

본 논문에서는 기존 연구에서는 시도되지 않았던 다중채널 환경 하에서 읽기 트랜잭션을 처리하기 위한 서버와 클라이언트 측의 동시성 제어기법을 제안하였다. 이 논문의 실험에서는 기본적으로 인덱스 전용 채널과 데이터 전용채널 구조를 이용하였다. 또한 LBCPC(Longest Broadcast Cycle Per Channel)라는 새로운 일관성의 단위를 제시했다. 데이터 전용채널에서는 이 LBCPC만큼 같은 BCPC(Broadcast Cycle Per Channel) 내의 데이터를 반복하여 방송한다. 또한 LBCPC마다 전체 데이터에 대한 제어 정보를 이용하여 자체적으로 트랜잭션의 검증을 실시한다. 이로 인해 트랜잭션의 일관성이 유지될 뿐만 아니라 단일채널의 긴 방송 주기보다 짧은 LBCPC로 인하여 재실행을 위한 대기 시간을 줄여줌으로써 평균 응답 시간을 줄여줄 수 있었다.