

2.2 다중채널 환경 모델

멀티채널 환경에서의 브로드 캐스트 방법 중에 하나인 [7]의 방법을 예로 들어보자. 그림 1은 [7]의 방법을 이용한 데이터 채널에서의 브로드 캐스트 스케줄을 작성하기 위한 알고리즘이다. 알고리즘에서 각 데이터에 대한 온도(Temperature)는 그들에 대한 접근 빈도(Access Frequency)를 나타내는 값이다. 따라서 온도가 높다(Hot)는 것은 데이터에 대한 접근 빈도가 높다는 것을 의미하며, 차갑다(Cold)는 것은 상대적으로 접근 빈도가 낮음을 의미한다.

```

begin
/*INPUT :
data temperature, data, the number of channel(k)
Output :
k-channel Data Broadcast Cycle
*/
sort data according to the their temperature.

average_channel_temperature =  $\frac{\sum(\text{temperature of data in DB})}{\text{the number of data channels}}$ 
for(i=0; i<the number of data channels; i++){
do
allocate data in ith data_channel;
while(  $\sum(\text{temperature of data} \in \text{each channel}) < \text{average\_channel\_temperature}$  )
}
}
end
    
```

[그림 1] data broadcast cycle 생성 알고리즘

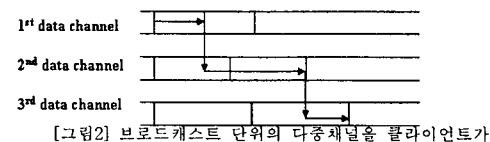
E, N, A, D, K, L, M, B, C, I, J, F, G, H와 같은 14개의 데이터와 4개의 채널이 주어질 경우 위의 그림1의 알고리즘을 적용시켜 보자. 데이터는 E, N, A, D, K, L, M, B, C, I, J, F, G, H의 순서로 온도에 따라 정렬되어 있고 평균 채널 온도가 70이라고 가정한다. 이때 데이터 E와 N의 온도가 70과 같거나 약간 초과한다고 가정한다면 데이터 E와 N은 채널1에 할당될 것이다. 이와 같은 방법 채널 2, 3, 4에 [A, D, K], [L, M, B, C], [I, J, F, G, H]와 같이 방송될 것이다.

다음 장에서는 위에서 설명한 예를 이용하여 다중 채널 환경에서 읽기 트랜잭션 처리를 위한 동시성 제어 기법에 대하여 설명하도록 하겠다.

3. 다중채널환경에서 읽기 트랜잭션 처리를 위한 동시성 제어 기법

3장에서 제안하는 방법을 설명하기 위해 필요한 가정들과 용어들을 살펴보도록 하겠다.

- 클라이언트는 순순간에 하나의 채널에만 접근이 가능하다.
 - 하나의 채널에서 다른 채널로 변환할 때 걸리는 시간은 무시할 수 있다고 가정한다.
 - 데이터 노드의 크기와 인덱스 노드의 크기는 동일하지 않다. 그러나 모든 데이터 노드들의 크기는 동일하며, 모든 인덱스 노드들의 크기는 동일하다고 가정한다.
 - 클라이언트는 자신의 읽기 트랜잭션에서 필요한 데이터를 이미 모두 알고 있다.
 - 각 채널에 할당된 데이터의 아이디는 브로드캐스트 사이클이 변하여도 변하지 않는다. 따라서 인덱스의 구조 또한 변경되지 않는다. 그러나 데이터의 값은 브로드캐스트 사이클이 변함에 따라 변경된다.
 - 일관성 주기(consistency cycle) : K개의 채널 중 브로드캐스트 사이클이 가장 큰 채널의 브로드캐스트 사이클과 동일한 사이클 길이로서 모든 채널들의 일관성을 유지해 주는 사이클이다.
 - 제어정보(control information)에서는 이전 일관성 주기에서 변경된 모든 채널의 데이터 아이디를 가지고 있다.
- 기존 단일채널 환경에서의 동시성 제어 기법은 브로드캐스트 주기 동안 일관성을 보장해줌으로서 가능했다. 그러나 단일채널에서의 브로드캐스트 주기는 다음과 같은 이유로 다중채널에서의 일관성의 단위가 되지 못한다.



[그림2] 브로드캐스트 단위의 다중채널을 클라이언트가

순차적으로 듣고 있는 경우

그림2는 단일 채널의 브로드캐스트 주기를 다중 채널에 적용하였을 경우 클라이언트가 채널을 듣고 있는 과정을 화살표를 통해 나타내고 있는 그림이다.

브로드캐스트 환경에서 클라이언트는 매순간 하나의 채널만을 들을 수 있다. 그러므로 그림 2처럼 채널1에서 첫 번째 브로드캐스트 사이클의 내용을 들은 뒤 두 번째 채널에서 첫 번째 브로드캐스트 사이클의 중간에서 듣기를 시작했으므로 클라이언트는 대기 모드로 대기하다가 다음 브로드캐스트 사이클의 시작부터 들어야 할 것이다. 이러한 구조는 채널1의 첫 번째 브로드캐스트 사이클의 데이터와 채널2의 두 번째 브로드캐스트 사이클의 데이터가 하나의 트랜잭션의 일관성을 요구할 경우 문제가 된다. 클라이언트는 채널1의 첫 번째 사이클을 들은 후 이미 읽은 데이터와 일관성이 있는 채널2의 첫 번째 사이클이 아니라 일관성이 다른 두 번째 브로드캐스트 사이클을 들을 것이기 때문이다.

본 논문에서 제시하는 동시성 제어를 설명하기 위하여 서버와 클라이언트의 작업을 설명하도록 한다. 우선 서버에서는 전력소모와 응답시간을 줄이기 위하여 인덱스와 데이터를 방송하게 된다. 3.1에서는 제안하는 다중 채널 구조를 위한 인덱스 채널구조에 대해서 간략하게 설명하고 3.2에서는 서버에서 일관성 있게 데이터를 방송하는 방법에 대하여, 3.3에서는 클라이언트가 다중채널 상에서 모바일 트랜잭션의 접근방법에 대하여 설명하도록 하겠다.

3.1 인덱스 채널구조 환경

다중 채널 또한 2장에서 설명한 단일 채널과 마찬가지로 배터리의 소모를 줄이기 위하여 인덱스를 제공해야 한다. 3장에서 사용하는 다중 채널 인덱스는 [8]에서 제안한 인덱스 채널 구조로서 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 인덱스 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 정보과 다음 제어정보의 시간 offset이 제공된다.

3.2 다중 무선 채널에서 데이터 채널 방송

2장에서도 언급한 것처럼 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.

```

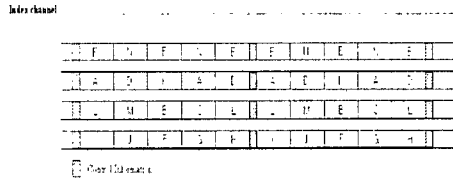
begin
/* INPUT:
k=the number of channel;
Dset_channel(i)=the set generated by figure 1 in channel i;
Bsize_channel(i)=the data size of broadcast cycle in channel i;
LBC=channel number of the longest broadcast cycle;
Consistency_cycle=Bsize_chaenn(LBC);
CI=the set of data ID updated during previous Consistency cycle over all of channels;
OUTPUT:
data allocation program over k data channel ;
*/
Ci=the data program in channel i;
for(i=1;i<=ki++){
for(j=1;j<=Consistency_cycle;j++){
a=i mod Bsize_channel(j);
if(a==0)(a=Bsize_channel(j));
Ci=Dset_channel(a);
}Ci=Ci+CI;
}
}
end
    
```

[그림3] 일관성을 유지하며 데이터를 방송하는 방법

그림 3은 다중무선 채널에서 데이터를 일관성 있게 방송하는 방법이다. 그림3을 통해 알 수 있듯이 데이터는 데이터 채널 중 브로드캐스트 사이클이 가장 긴 채널의 브로드캐스트 길이만큼 나머지 채널들을 반복된다. 즉 채널 중 브로드캐스트 길이가 가장 긴 채널의 브로드캐스트 사이클이 모든 채널의 일관성 주기가 된다. 이 가장 긴 채널의 브로드캐스트 사이클의 길이를 일관성 주기(Consistency Cycle)이라고 하자. 브로드캐스트 사이클이 가장 긴

채널이외의 모든 채널은 일관성 주기만큼 반복되는데 브로드캐스트 사이클의 모든 데이터가 반복될 필요는 없다. 즉, 공간이 남아 있는 만큼 데이터를 반복한다.

그림4는 2.2장의 예제를 바탕으로 그림3의 알고리즘을 적용한 그림이다. 채널1의 브로드캐스트 사이클 길이는 2, 채널 2의 브로드캐스트 사이클 길이는 3, 채널3의 브로드캐스트 사이클 길이는 4, 채널4의 브로드캐스트 사이클 길이는 5이므로 브로드캐스트 사이클 길이가 가장 긴 채널 4의 브로드캐스트 사이클 길이 5가 일관성 주기가 된다. 나머지 채널들은 그림6과 같이 일관성 주기인 5만큼 데이터를 반복하게 된다.



[그림4] 서버 측에서 방송되는 인덱스와 데이터구조

3.3 다중채널 상에서 모바일 트랜잭션의 접근 방법

다중채널 상에서 모바일 트랜잭션의 접근방법은 그림5와 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;
CC(i≤i≤k)= the length of broadcast cycle of channel i;
CI_size=k=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_positon/DSize ;
position =Size_of_Data_Channel* period_number+OFFSET;
if (current_position < CI_position) {
while(current_index_positon>position) {
period_number++;
position=size_of_Data_Channel *period_number+OFFSET;
}
}
if (current_position ≥ real_CI) { CCN++; bp_num=0; }
position=(CCN * (CCk+CI_size)+1)+CI*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)+CI*BP_num +OFFSET;
bp_num++;}
}
}
end
    
```

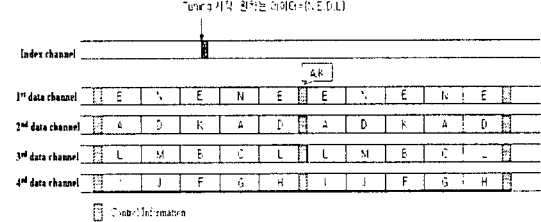
[그림 5] 다중채널 상에서 모바일 트랜잭션의 접근방법

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[5]. 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.

클라이언트가 원하는 데이터를 N,E,D,L이라고 가정해보자. 이때 인덱스를 통하여 채널정보와 데이터의 offset, 다음 제어정보의 offset을 알아내고 데이터 채널의 N과 L를 읽은 다음 제어 정보를

읽게 된다. 이때 제어 정보는 데이터 아이디 A, K 이므로 클라이언트가 더 읽어야 하는 데이터 D, E와 일치 하지 않으므로 일관성이 유지되어 그림 6과 같이 트랜잭션은 계속 진행된다.

구체적으로 그림5의 알고리즘을 통하여 원하는 데이터의 시간 오프셋을 알아보도록 하자. (current_position = 25, Dsize=10, Bcast(1)=2, CI_size=1) N=20*1+10=30, L,E,D는 모두40으로 position이 동일하다. 이때 알고리즘에 의해 가장 긴 브로드캐스트 사이클을 갖고 있는 3번째 채널의 L을 time offset 40에 먼저 읽고 CI validation 검사 후 E=51+20*0+0=52, D=51+30*0+10=62시간에 데이터를 읽게 된다.



[그림6] validation을 통과한 트랜잭션의 실행

4. 결론

본 논문에서는 기존 연구에서는 시도되지 않았던 다중채널 환경 하에서 읽기 트랜잭션을 처리하기 위한 서버와 클라이언트 측의 동시성 제어기법을 제안하였다. 지금까지 제안된 방법들은 브로드캐스트환경에서 트랜잭션 처리는 모두 단일 채널환경에서 브로드캐스트 사이클을 트랜잭션 일관성의 단위로 간주하고 연구되었다. 그러나 다중채널을 사용해야 하는 환경에서 기존의 브로드캐스트 단위로 트랜잭션의 일관성을 보장할 수 없다. 다중채널에서는 클라이언트가 매순간 하나의 채널만을 들을 수 있기 때문에 각 채널마다 트랜잭션의 일관성이 있는 브로드캐스트 사이클을 들는 것은 불가능하다.

따라서 서버 측에서 일관성 주기만큼 다른 채널들의 일관성 있는 데이터를 반복하는 것과 이전 일관성 주기 안에 변경된 데이터에 대한 제어정보를 방송함으로써 일관성 있는 데이터를 제공하고 클라이언트 측에서는 제어정보에 따라 일관성을 검사하면서 읽기 트랜잭션을 수행할 수 있었다.

5. 참고문헌

- [1] S. Acharya, M. Franklin, S. Zdonik, and R. Alonso, "Broadcast Disks: Data Management for Asymmetric Communication Environments," Proc. ACM SIGMOD International Conference on Management of Data, pp. 199-210, 1995.
- [2] S. Acharya, M. Franklin, and S. Zdonik, "Balancing Push and Pull for Data Broadcast," Proc. ACM SIGMOD Int. Conf. on Management of Data, pp. 183-194, 1997.
- [3] P. A. Bernstein, V. Hadzilacos, N. Goodman, "Concurrency Control and Recovery in Database Systems," Addison Wesley, Massachusetts, 1987.
- [4] H. Cho, "Concurrency Control for Read-Only Client Transactions in Broadcast Disks," IEICE Trans. Commun., vol.E86-B, no.10, 2003.
- [5] V. Lee, K-W. Lam, T-W Kuo, "Efficient validation of mobile transactions in wireless environments," The Journal of Systems and Software, pp.183-193, 2004.
- [6] K. Prabhakara, K.Hua, and J. Oh, "Multi-Level Multi-Channel Air Cache Design for Broadcasting in a Mobile Environment", in Proceedings. 16th International Conference on Data Engineering, 2000.
- [7] K. Prabhakara, K. Hua, and J. Oh, Multi-Level Multi-Channel Air Cache Design for Broadcasting in a Mobile Environment, in Proceedings. 16th International Conference on Data Engineering, 2000.
- [8] Sungwon Jung, Byungkyu Lee and Sakti Pramanik, "A Tree-Structure Index Allocation Method with Replication over Multiple Broadcast Channels in Wireless Environments", IEEE transactions on knowledge and data engineering, VOL 17, No.3, March 2005