

브로드캐스트 환경에서 에너지 효율과 데이터 접근빈도를 동시에 고려한 하이브리드 인덱스 배치 기법

박지현^o 박광진 강상원 김종완 임석진 황종선
고려대학교
(hyun^o, kjpark, swkang, wany, seokjin, hwang)@disys.korea.ac.kr

A Hybrid Index Allocation Scheme Considering both Energy Efficiency and Data Access Frequencies in Mobile Broadcast Environments

JieHyun Park^o KwangJin Park Sang-Won Kang Jongwan Kim SeokJin Im Chong-Sun Hwang
Department of Computer Science and Engineering, Korea University

요 약

이동 컴퓨팅 환경에서 다수의 이동 클라이언트들에게 데이터를 전달할 때는 클라이언트들의 동시 데이터 접근을 지원하는 브로드캐스트 기법을 사용하면 무선 채널 대역폭의 협소함과 클라이언트 측의 에너지 제약과 같은 단점을 해결할 수 있다. 기존 기법들은 클라이언트의 데이터에 대한 접근빈도(access frequencies)와 클라이언트의 에너지 효율(energy efficiency)을 동시에 고려하지 않았다. 따라서 원하는 데이터가 올 때까지 계속해서 채널을 들이야 함으로 인해 에너지 소비를 많이 하거나, 데이터를 얻을 때까지 추가한 많은 양의 정보에 따른 지연이 발생하는 단점이 있다. 본 논문에서는 클라이언트의 에너지 절약을 위한 tuning time을 최소화하고 실제 데이터를 얻을 때까지 소요되는 access time의 효율을 높이기 위해 데이터의 접근빈도와 에너지 효율을 동시에 고려하는 HIDAF: Hybrid Index considering Data Access Frequencies 기법을 제안한다. 제안하는 기법은 트리기반 기법과 해싱기반 기법을 함께 적용하여 구성된 인덱스 브로드캐스트 주기에 배치한다. HIDAF 기법은 데이터 접근빈도를 고려한 트리기반 인덱스를 배치함으로써 데이터를 얻기 위한 클라이언트들의 평균 access time을 줄일 수 있고, 이러한 인덱스에 해싱기반 기법을 추가함으로써 클라이언트의 에너지 효율을 최소화하는 새로운 브로드캐스팅 기법이다. HIDAF 기법은 브로드캐스트 주기에 데이터의 접근빈도를 고려한 인덱스를 교차로 추가하여 핫 데이터에 대한 클라이언트들의 access time을 줄임으로써 전체 사용자에 대한 평균 access time을 최소화하는 동시에 클라이언트들의 제한된 에너지 소비를 최소화하는데 목적이 있다. 제안기법에 대한 평가는 수학적 분석을 통해 HIDAF 기법과 기존의 브로드캐스트 기법의 성능을 비교 분석한다.

1. 서 론

이동 통신 네트워크의 발전으로 무선 채널을 통해 클라이언트(client)들은 언제 어디서든 원하는 정보를 얻을 수 있게 되었다. 클라이언트들의 자유로운 이동성은 이동 컴퓨팅 환경의 장점이지만 이로 인해 이동 클라이언트 단말기의 에너지 제한, 무선 통신 대역폭의 협소함, 높은 에러율, 잦은 접속 단절 등의 여러 가지 제약 조건이 발생한다[1]. 이러한 제약점을 가지는 이동 컴퓨팅 환경에서, 데이터 브로드캐스트 기법을 사용하면 다수의 이동 클라이언트의 질의처리를 개별적으로 하지 않으므로 무선 채널의 협소한 대역폭, 클라이언트 단말기의 제한된 에너지와 같은 제약점을 보완할 수 있다[8, 10].

클라이언트들의 데이터에 대한 접근빈도를 고려하지 않은 브로드캐스트 기법에서는 모든 데이터에 대한 access time이 동일하지만 일반적인 모든 데이터 접근빈도는 모든 데이터에 대해 동일하지 않으므로 접근빈도를 고려하여 access time을 차별화하는 브로드캐스트 기법이 필요하다.

데이터의 접근빈도만을 반영하여 전달할 경우, 클라이언트의 데이터에 대한 평균 access time은 단축되지만 원하는 데이터를 얻을 때까지 오랜 시간 동안 무선 채널을 들고 있어야 하므로 많은 에너지를 소비하게 된다. 또한, 접근빈도를 고려하지 않은 인덱스를 추가하는 브로드캐스트의 경우에는 에너지 소비 면에서는 효율적이지만 추가한 많은 양의 인덱스에 따른 지연이 발생한다.

본 논문에서는 데이터 접근빈도와 에너지 효율을 모두 고려하여 데이터를 브로드캐스트 하는 기법인 HIDAF 기법을 제안한다. HIDAF 기법은 일부 전체 데이터에 대한 인덱스 대신, 데이터의 접근빈도를 고려한 인덱스를 추가하여 브로드캐스트 주기를 줄임으로써 다수의 사용자에 대한 평균 access time을 줄이는 것과 트리기반 인덱스에 해싱 버킷을 추가하여 트리기반 인덱스를 파티션함으로써 트리높이를 줄여 tuning time을 줄이는 두 가지 목적이 있다.

본 논문의 구성은 다음과 같다. 2장에서는 기존의 연구들에 대해 알아보고 3장에서 기존 연구의 문제점을 지적하고 새로운 데이터 브로드캐스트 기법을 제안한다. 4장에서는 수식을 통한 분석으로 제안된 기법의 성능을 분석하고 기존의 기법들과 비교하고, 5장에서 결론을 맺는다.

2. 관련 연구

브로드캐스트의 가장 큰 장점은 확장성(scalability)으로 이동 클라이언트의 수가 증가하여도 그에 따른 추가 통신비용이 거의 발생하지 않는다는 점이다[5]. 또한 무선 통신 네트워크 환경에서 데이터를 다운로드 하는 것은 요청을 업링크 하는 것보다 더 적은 에너지를 필요로 하기 때문에 브로드캐스트를 사용할 경우 이동 클라이언트 단말기의 에너지 사용도 줄일 수 있다[9]. 데이터 브로드캐스트 구성 기법에서 기존의 연구들은 크게 두 가지로 분류할 수 있다. 하나는 클라이언트들의 데이터 접근빈도를 고려해서 데이터들의 전송 횟수를 달리하여 평균 access time을 단축시킨 브로드캐스트 기법[2]이고 또 다른 연구 방향

은 클라이언트의 제한된 에너지 소비를 줄이기 위해 서버가 데이터와 데이터에 대한 인덱스를 함께 브로드캐스트 하고 클라이언트는 이러한 인덱스를 이용하여 원하는 데이터가 브로드캐스트 되는 시간동안에만 채널을 들는 선택적 데이터 수신(selective tuning)을 가능하게 하는 것이다[3, 6].

2.1 접근빈도를 고려한 브로드캐스트

데이터의 접근빈도 차이를 고려하여 데이터들의 전송 빈도를 달리함으로써, 클라이언트들이 데이터를 얻기 위해 기다려야 하는 시간을 차별화하는 브로드캐스트 기법에서 데이터는 클라이언트의 상대적인 접근 빈도에 따라 핫 데이터(hot data)와 콜드 데이터(cold data)로 분류된다[2]. 클라이언트들이 자주 참조하는 핫 데이터는 다른 데이터들보다 전송 횟수를 많이 하고, 그렇지 않은 콜드 데이터는 전송 횟수를 적게 하여 그림 1의 형태로 데이터를 브로드캐스트 할 수 있다. 핫 데이터를 먼저 전송한 후에 콜드 데이터를 전송하는 방식은 핫 데이터가 콜드 데이터보다 자주 전송됨으로써 접근빈도는 만족시키지만, 핫 데이터를 얻기 위해 기다려야 하는 시간이 일정하지 않기 때문에 그림 1과 같이



그림 1. 데이터의 접근빈도를 고려한 브로드캐스트

전송빈도가 다른 데이터들을 서로 교차시켜 전송하면 클라이언트가 핫 데이터를 얻기 위해 기다려야 하는 access time을 일정하게 조절할 수 있다.

2.2 에너지 효율을 고려한 브로드캐스트

이동 클라이언트의 단말기는 에너지가 제한되어 있기 때문에 에너지의 소비를 줄이기 위해 브로드캐스트 주기에 인덱스를 배치시키는 기법이다. 클라이언트는 인덱스를 통해 원하는 데이터가 브로드캐스트 되는 시간정보를 파악한 후, 그 데이터가 브로드캐스트 되는 시간에만 깨어나 선택적으로 데이터 수신이 가능하다 [3, 6, 7].

2.2.1 트리기반 기법

트리기반 기법을 사용한 인덱스를 이용한 브로드캐스트 기법의 구성 요소는 브로드캐스트 채널을 통해서 전송되는 가장 작은 논리적 단위를 버킷(bucket)이라고 하고 버킷들의 집합을 세그먼트(segment)라고 한다. 인덱스 세그먼트는 데이터들의 키 값들을 가지고 있고, 모든 데이터 버킷은 다음 인덱스에 대한 포인터(pointer)와 다음 브로드캐스트 주기의 오프셋(offset)을 가지고 있다. 그림 2는 브로드캐스트 되는 데이터들 사이에 트리기반 인덱스를 반복해서 전송하는 기법 중 하나인

(1,m)인덱싱을 나타낸다.



그림 2. (1,m) 인덱싱

2.2.2 해싱기반 기법

해싱기반 인덱스를 이용한 브로드캐스트 기법은 최소 tuning time을 보장하지만 데이터 버킷마다 해싱 매개변수를 추가하기 때문에 access time이 길어진다 [6].

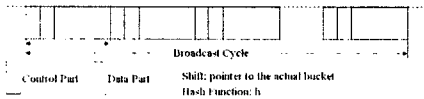


그림 3. 해싱 기반 브로드캐스트 구조

그림 4는 기본적인 해싱 기반 브로드캐스트 구조이다. 각 데이터 버킷은 조절파트(control part)와 데이터파트(data part)로 구성된다. 데이터파트는 실제 데이터를 포함하고 있고 조절파트는 데이터에 대한 정보를 저장한다. 클라이언트는 조절파트를 듣고 원하는 데이터일 경우에만 데이터 버킷을 수신한다.

3. HIDAF 기법

이 장에서는 이동 컴퓨팅 환경에서 데이터 브로드캐스트 주기에 데이터 접근빈도를 고려한 트리기반 인덱스를 교차하여 배치할 뿐만 아니라 이러한 트리기반 인덱스에 해싱기반 기법을 접목한 HIDAF: Hybrid Indexing Scheme considering Data Access Frequencies 기법을 제안한다. [2]에서 제안하는 기법(그림 1)은 데이터의 접근빈도 차이를 고려하여 데이터를 전송하는 횟수를 구분함으로써 핫 데이터에 대한 access time은 상대적으로 줄이지만 클라이언트가 원하는 데이터를 얻을 때까지 오랜 시간 동안 무선 채널을 들고 있어야 하므로 많은 에너지를 소비한다. [3]에서 제안하는 (1,m)인덱싱(그림 2)에서는 접근빈도를 고려하지 않았지만 인덱스를 추가함으로써 에너지 소비는 줄어드는 반면 추가된 필요 이상의 인덱스의 비효율적인 중복에 따른 부가적인 지연이 발생한다.

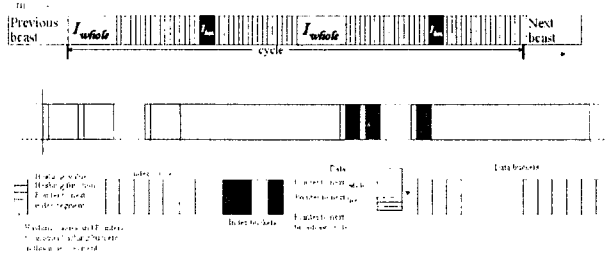


그림 4. 제안기법인 HIDAF 기법

본 논문에서는 데이터의 브로드캐스트 주기에 전체 데이터의 인덱스를 m번 모두 추가하지 않고 데이터의 접근빈도를 고려한 일부 데이터에 대한 트리기반 인덱스를 교차로 추가하고, 이러한 트리기반 인덱스에 해싱기반 기법을 접목시켜 구성된 인덱스를 배치하는 브로드캐스트 기법을 새로이 제안한다. 그림 4와 같이 전체 데이터에 대한 인덱스의 전송횟수를 줄이는 대신, 접근빈도를 고려하여 핫 데이터에 대한 인덱스를 추가하는 HIDAF 기법의 경우, 브로드캐스트 주기가 기존 기법인 (1,m) 인덱싱에 비해 짧아진다. 따라서 많은 사용자들이 자주 원하는 데이터인 핫 데이터에 대한 인덱스를 얻기 위한 시간이 줄어들기 때문에 전체 데이터들에 대한 평균 access time이 줄어든다. 또한 HIDAF 기법은 해싱 값, 해싱 함수, 다음 인덱스 세그먼트를 가리키는 포인터 및 현재 인덱스 세그먼트에 포함된 모든 해싱 버킷들의 포인터로 구성된 해싱 매개변수를 포함하는 버킷을 트리기반 인덱스에 추가함으로써 하나의 트리기반 인덱스를 해싱함수에 따라 여러 개의 작은 인덱스들로 나눈다. 클라이언트는 해싱 버킷을 통해 얻은 해싱 함수를 이용해서 원하는 데이터의 키 값과 해싱 버킷의 해싱 값을 비교하여 일치할 때에만 일치하는 해싱 값을 가진 데이터들에 대한 정보를 가지고 있는 트리기반 인덱스만 읽으면 된다. 해싱 기법을 통해 나누어진 인덱스 트리의 높이는 나누기 전의 트리 높이보다 낮아지기 때문에 트리 전체를 읽어야 하는 트리기반의 인덱스만 사용했을 경우보다 에너지 소비가 낮아진다. 자세한 알고리즘은 그림 5, 6과 같다.

표 1. 매개변수

m	the number of replications of Indices
BS_I	Broadcast Schedule with Index
D	$\{d_1, d_2, d_3, \dots, d_n\}, n = D_{hot} + D_{cold} $
D_{hot}	$\{D_{hot} \in D d_{h_1}, d_{h_2}, d_{h_3}, \dots, d_{h_{n(s+1)}}\}$
D_{cold}	$\{D_{cold} \in D d_{c_1}, d_{c_2}, d_{c_3}, \dots, d_{c_{n-(s+1)}}\}$
I_{whole}	Index of D
I_{hot}	Index of D_{hot}
$D_{hot} : D_{cold}$	1 : s (ratio of the number of data)
k_{pw}	the number of levels in the partition of I_{whole} tree
k_{ph}	the number of levels in the partition of I_{hot} tree
H	hashing bucket
H_f	hashing function
w, h	the number of hashing bucket for I_{whole}, I_{hot}

Index construction.
 D sorted in the descending order according to access frequency;
 D_{hot} is the number of $n/(s+1)$ data which hold a high rank;
 The remaining data is called $D_{cold} = D - D_{hot}$;
 I_{whole_i} is defined as the order of partition of I_{whole} ;
 I_{hot_j} is defined as the order partition of I_{hot} ;
 The hashing function is modulo a, b.

```

Algorithm BroadcastScheduling
//indices are allocated in broadcast cycle.
input: D, data set;
output: BS_I, broadcast schedule with indices;

procedure:
count = 1
for (s=1; s <= m; s++) do
if count is odd then
for (i=1; i <= a; i++) do
add  $H_i$  to  $BS_I$  //해시 버킷 추가
add  $I_{whole_i}$  to  $BS_I$ ; //홀수 번째  $I = I_{whole}$ 
end for
else
for (j=1; j <= a; j++) do
add  $H_j$  to  $BS_I$ 
add  $I_{hot_j}$  to  $BS_I$ ; //짝수 번째  $I = I_{hot}$ 
end for
end if
add  $d_{(i+s)/m-n/m+1}$  to  $d_{(i+n)/m}$  to  $BS_I$ ; //  $D$ 를 인덱스의 뒤에 추가
count++;
end for
return  $BS_I$ ;
    
```

그림 5. HIDAF 기법의 주기를 구성하는 알고리즘

```

Mobile client requests for data with key Q
Tune into the broadcast channel
Read the bucket
Retrieve Pointers and Offset
Go to doze mode
(1) if data with key Q is  $D_{hot}$  then
Tune in again index segment ( $I_{whole}$  or  $I_{hot}$ )
Read the hashing bucket
if hashing value  $s$  or  $j = H_{f_i}$  or  $j(Q)s$  or  $j < H_{f_i}$  or  $j(Q)$ 
Read next H
else
Read next index segment
end if
else
Tune in again index segment ( $I_{whole}$ )
Read the hashing bucket
if hashing value  $s = H_{f_j}(Q)$ 
Read the partition of index
else if hashing value  $s < H_{f_j}(Q)$ 
Read next H
else
Read next index segment
end if
end if
Go to doze mode
Tune in again when data with key Q are broadcasted
Download data with key Q
(2) if Q = key broadcasted already then
Go to doze mode
Repeat from (1) in next broadcast
end if
    
```

그림 6. 클라이언트의 다운로드 알고리즘

표 1은 그림 5, 6과 4장의 수식에서 사용할 매개변수이다. 그림 5는 앞서 설명한 브로드캐스트 주기를 구성하는 기법을 설명한 알고리즘이다. 본 논문의 인덱스를 구성하기 위한 각 데이터에 대한 접근빈도와 핫 데이터, 콜드 데이터의 비율은 지프분포(Zipf distribution)를 따른다 [4].

그림 6은 클라이언트가 원하는 데이터를 다운로드 하는 알고리즘이다. 모든 데이터 버킷은 전체 데이터에 대한 인덱스와 핫 데이터에 대한 인덱스의 포인터, 두 개를 포함하고 다음 브로드캐스트 주기에 대한 오프셋을 가진다. 클라이언트는 doze mode와 active mode로 동작한다. doze mode란 브로드캐스트 채널로 어떤 데이터가 들어오는지 확인하지 않고 최소의 전력만을 사용하여 작업하는 상태이고 active mode는 브로드캐스트 채널을 통해 원하는 데이터를 수신하기 위해 청취하는 상태로 에너지를 소비하는 상태를 말한다.

4. 수학적 성능 분석

이 장에서는 수학적 분석을 통해 HIDAF 기법을 평가하고 기존의 브로드캐스트 기법인 (1,m) 인덱싱과 HIDAF 기법의 성능을 비교 분석한다. 브로드캐스트 기법의 성능은 access time과 tuning time, 두 가지 기준에 의해 평가된다. 여기서 시간은 버킷의 수로 측정하며 핵심 버킷의 크기는 1로 가정한다 [11]. access time은 probe wait과 bcast wait을 합한 시간이며, probe wait은 클라이언트가 브로드캐스트 채널을 듣기 시작해서 원하는 데이터에 대한 인덱스 정보를 얻게 될 때까지의 평균 시간이고, bcast wait은 인덱스를 들은 시점으로부터 원하는 데이터를 얻을 때까지 소요되는 평균 시간을 나타낸다. tuning time은 클라이언트가 실제로 채널을 듣는 시간을 나타낸다. 즉, 클라이언트가 active mode로 데이터 수신을 위해 에너지를 사용하는 시간을 나타낸다.

HIDAF 기법의 access time을 수식으로 표현하면 수식 (1)~(5)와 같다. 수식 (1),(2)의 probe wait(D_{hot})과 probe wait(D_{cold})은 각각 D_{hot} 과 D_{cold} 에 대한 probe wait이고, 수식 (3)의 bcast wait은 전체 주기의 $\frac{1}{2}$ 이며, 수식 (4)의 access time(D_{hot})은 probe wait(D_{hot})과 bcast wait을 합한 시간이다. 수식 (5)의 access time(average)은 D_{hot} 과 D_{cold} , 두 데이터에 대한 평균 access time을 나타낸다.

그림 7은 수식 (4),(5)를 토대로 그래프를 이용해서 HIDAF 기법의 access time을 기존 기법과 비교한 것이다. (a)는 제안 기법에 추가되는 인덱스들의 크기가 기존 기법의 것보다 작아지기 때문에 HIDAF 기법의 브로드캐스트 주기가 기존 기법의 주기보다 짧아지게 된다. 따라서 m 값의 변화에 따른 access time이 기존의 (1,m) 인덱싱에 비해 완만히 증가하고, 모든 m 값의 구간에서 기존의 기법인 (1,m) 인덱싱보다 HIDAF 기법의 성능이 더 좋았던 것을 알 수 있다. (b)는 데이터 수의 변화에도 제안 기법의 access time이 기존기법보다 줄어들음을 보여준다.

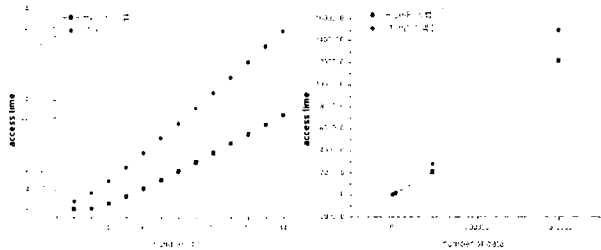
$$\text{probe wait}(D_{hot}) = \frac{1}{4}(I_{whole} + I_{hot} + \frac{2D}{m}) + \frac{1}{4}H(w+h) \quad (1)$$

$$\text{probe wait}(D_{cold}) = \frac{1}{2}(I_{whole} + I_{hot} + \frac{2D}{m}) + \frac{1}{2}H(w+h) \quad (2)$$

$$\text{bcast wait} = \frac{1}{2}(\frac{m \cdot I_{whole}}{2} + \frac{m \cdot I_{hot}}{2} + D) + C + \frac{m}{2}H(w+h) \quad (3)$$

$$\text{access time}(D_{hot}) = \frac{1}{4}[(m+1) \cdot I_{whole} + (m+1) \cdot I_{hot} + (\frac{2}{m} + 2) \cdot D] + C + [(H(w+h))(\frac{1}{4} + \frac{m}{2})] \quad (4)$$

$$\text{access time(average)} = \frac{m}{4} \cdot (I_{whole} + I_{hot}) + \frac{3}{10} \cdot (I_{whole} + I_{hot}) + \frac{3D}{5m} + \frac{D}{2} + C + [\frac{3+5m}{10}H(w+h)] \quad (5)$$



(a) 인덱스 수의 변화에 따른 access time (b) 데이터 수의 변화에 따른 access time

그림 7. 그래프를 통한 성능 비교 분석

HIDAF 기법의 tuning time을 수식으로 표현하면 수식 (6)~(8)과 같다. tuning time은 데이터 버킷 하나를 읽음으로써 포인터와 오프셋을 얻은 후, 다운로드하고자 하는 데이터의 key 값을 핵심 합수로 계산한 값과 핵심 버킷의 핵심 값이 일치할 가능성 = 1, 일치하지 않을 가능성 = 4, 일치하지 않으면 다음 핵심 버킷을 읽어야 할 가능성 = 4를 5로 나누고, 트리 기반 인덱스의 높이를 더한 것으로 계산할 수 있다. 수식 (6),(7)의 tuning time(D_{hot})과 tuning time(D_{cold})은 각각 D_{hot} 과 D_{cold} 에 대한 tuning time이고, 수식 (8)의 tuning time(average)은 D_{hot} 과 D_{cold} , 두 데이터에 대한 평균 tuning time을 나타낸다.

$$\text{tuning time}(D_{hot}) = 1 + \frac{(1 \times 4 + 1 \times 4) + (1 \times 1)}{5} + \frac{k_{pu} + k_{sh}}{2} \quad (6)$$

$$\text{tuning time}(D_{cold}) = 1 + \frac{(1 \times 4 + 1 \times 4) + (1 \times 1)}{5} + k_{pu} \quad (7)$$

$$\text{tuning time(average)} = 1 + \frac{(1 \times 4 + 1 \times 4) + (1 \times 1)}{5} + \frac{3k_{pu} + 2k_{sh}}{5} \quad (8)$$

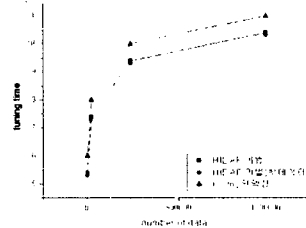


그림 8. tuning time 비교

그림 8을 통해 알 수 있듯이 핵심기반 기법을 통해 인덱스가 나눠지기 때문에 일어아 하는 트리의 높이가 낮아져서 tuning time도 기존의 기법에 비해 줄어든다.

5. 결론

이동 컴퓨팅 환경에서 데이터 전송을 위해 사용되는 기존의 브로드캐스트 기법들은 에너지의 효율성은 고려하지 않고 데이터의 접근빈도만을 고려하거나, 접근빈도는 고려하지 않은 인덱스를 배치하고 중복하는 기법으로 연구되었다. 본 논문에서는 클라이언트가 원하는 데이터를 얻는 때까지의 access time과 tuning time을 동시에 줄이는 HIDAF 기법을 제안한다. 이 기법은 브로드캐스트 전송 주기에 전체데이터에 대한 인덱스를 중복하는 대신 핫 데이터에 대한 인덱스를 교차시켜 중복시키는 기법으로 브로드캐스트 주기를 줄임으로써 핫 데이터를 원하는 다수의 클라이언트들의 access time을 줄여 데이터에 대한 평균 access time을 단축시키고, 트리 기반 인덱스를 핵심기반 기법으로 나누어 클라이언트들의 tuning time 또한 줄일 수 있다.

6. 참고 문헌

- [1] G.H. Forman and J. Zahorjan, "The Challenges of Mobile Computing," IEEE Computer 27(4), 1994, pp. 38-47.
- [2] S. Acharya, R. Alonso, M. Franklin, and S. Zdonik, "Broadcast Disks: Data Management for Asymmetric Communications Environments," in Proceedings of the ACM SIGMOD Conference on Management of Data, San Jose, California, May 1995, pp. 199-210.
- [3] T. Imielinski, S. Viswanathan, and B.R. Badrinath, "Data on Air: Organization and Access," IEEE Transactions of Data and Knowledge Engineering, vol. 9, no. 3, 1997, pp. 353-372.
- [4] Chih-Hao Hsu, Guanling Lee, and Arbee L.P. Chen, "Index and Data Allocation on Multiple Broadcast Channels Considering Data Access Frequencies," in Proceedings of Third International Conference on Mobile Data Management, 2002, pp. 87-92.
- [5] Pin-Kwang Eng, "Disseminating Data in Unreliable Wireless Environment," in Proceedings of Third International Conference on Mobile Data Management, 2002, pp. 157-158.
- [6] X. Yang and A. Bouguettaya, "Broadcast-Based Data Access in Wireless Environments," proc. Int'l Conf. Extending Database Technology (EDBT '02), Mar. 2002.
- [7] Ming-Syan Chen, Kun-Lung Wu, and Philip S. Yu, "Optimizing Index Allocation for Sequential Data Broadcasting in Wireless Mobile Computing," Proceedings in IEEE Transactions on Knowledge and Data Engineering, 2003, pp. 161-173.
- [8] A. Celik, Ping Ding, and J.A. Holliday, "Data broadcasting with Data Item Locality and Client Mobility," in Proceedings of the IEEE International Conference on Mobile Data Management, 2004, pp. 166.
- [9] Andrew Y. Ho, and Dik Lun Lee, "Data Indexing for Heterogeneous Multiple Broadcast Channel," in Proceedings of the IEEE International Conference on Mobile Data Management, 2004, pp. 274-283.
- [10] B. Zheng, W.C. Lee, and D.L. Lee, "Search Continuous Nearest Neighbors On the Air," in Proceeding of the First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, 2004.
- [11] Xu Yang and Athman Bouguettaya, "Adaptive Data Access in Broadcast-Based Wireless Environments," IEEE Transactions of Data and Knowledge Engineering, vol. 17, no. 3, March 2005, pp. 326-338.