

다중방송 채널에서 인덱스를 이용한 브로드캐스트 디스크 기반의 데이터 할당 기법

(A Data Allocation Method based on Broadcast Disks Using
Indices over Multiple Broadcast Channels)

이 원 택 [†] 정 성 원 ^{††}
(Wontaek Lee) (Sungwon Jung)

요 약 무선 환경에서 브로드캐스트방법을 이용하여 데이터를 전송할 때, 중요한 문제는 모바일 클라이언트가 요구하는 데이터에 빠르게 접근할 수 있도록 하는 것이다. 기존 연구는 선호도에 따라 정렬된 데이터들을, 제공된 채널의 수만큼 평균접근시간을 줄이는 최적의 분할지점을 찾아 채널에 할당하는 기법을 사용하였다. 이러한 방법의 문제는 동등 채널 내의 데이터 간 선호도 차이가 브로드캐스트 스케줄링에 반영되지 못한다는 것이다. 이 문제점을 해결하고자 기존 방법에 따라서 다중 채널에 데이터를 분할한 후, 브로드캐스트 디스크(broadcast disk)기법을 적용하여 동등 채널 내의 데이터들 사이에도 선호도 차이에 따라 차별적인 반복주기가 적용되도록 하는 방법이 제안 되었다. 그러나 기존의 데이터 분할은 선호도 차이에 따라 정렬된 데이터를 순서대로 채널에 할당하였기 때문에 각 채널 안에서의 데이터 간 선호도 차이가 크지 않았다. 결국 브로드캐스트 디스크 기법을 적용하더라도 각 데이터의 반복주기 차이는 선호도 차이에 따라 작게 나타나기 때문에 접근시간의 단축 효과는 크지 않다. 본 연구에서는 제공된 다중 채널에 선호도가 높은 데이터와 낮은 데이터가 고르게 분포되도록 하는 방법을 제안한다. 제안하는 방법은 정렬된 데이터들이 인접 데이터와 떨어져 각각 다른 채널에 포함되도록 할당한다. 기존의 방법은 인덱싱 기법을 사용하지 않더라도 상위채널부터 선호도 순서에 따른 데이터 접근이 가능했다. 하지만, 제안하는 방법은 선호도 순위에 따른 데이터 접근이 어렵기 때문에 접근시간을 줄이기 위해서는 인덱스사용이 필수적이다. 이에 대해서 제안하는 데이터 할당 방법에 접근시간 단축을 위한 효율적인 인덱스 기술도 제시한다. 인덱스 방법은 요구 데이터에 대한 빠른 채널 위치 검색을 위해 독립적인 인덱스 채널을 사용한다. 위의 과정을 통해서 제시하는 방법은 기존의 연구들보다 데이터의 선호도를 보다 정확하게 반영하여 전체 클라이언트의 선호 데이터에 대한 빠른 접근 성능을 보이는 것을 실험을 통해서 보여줄 것이다.

키워드 : 데이터 브로드캐스트, 다중 방송 채널, 이동 컴퓨팅, 데이터 할당

Abstract In this paper, we concentrate on data allocation methods for multiple broadcast channels. When the server broadcasts data, the important issue is to let mobile clients access requested data rapidly. Previous works first sorted data by their access probabilities and allocate the sorted data to the multiple channels by partitioning them into multiple channels. However, they do not reflect the difference of access probabilities among data allocated in the same channel. This paper proposes ZGMD allocation method. ZGMD allocates data item on multiple channels so that the difference of access probability in the same channel is maximized. ZGMD allocates sorted data to each channels and applies Broadcast Disk in each channel. ZGMD requires a proper indexing scheme for the performance improvement. This is because in ZGMD method each channel got allocated both hot and cold data. As a result, the sequential search heuristic does not allow the mobile client to access hot data items

[†] 학생회원 : 서강대학교 컴퓨터공학과
magicsniper@sogang.ac.kr
^{††} 종신회원 : 서강대학교 컴퓨터공학과 교수
jungsung@sogang.ac.kr

논문접수 : 2007년 10월 8일

심사완료 : 2008년 1월 23일

Copyright©2008 한국정보과학회 : 개인 목적이거나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지 : 데이터베이스 제35권 제3호(2008.6)

quickly. The proposed index scheme is based on using dedicated index channels in order to search the data channel where the requested data is. We show that our method achieve the near-optimal performance in terms of the average access time and significantly outperforms the existing methods.

Key words : Data Broadcast, Mobile Computing, Multiple Broadcast Channels, Data Allocation

1. 서론

무선 이동 통신 장치의 발달은 정보기기 사용에 대한 장소의 제약을 사라지게 했다. 이러한 무선 환경에서 무선기기를 사용하는 불특정 다수가 동시에 원하는 정보를 빠르게 얻기 위해서는 브로드캐스트(Broadcast) 방법이 가장 효율적이고 확장성이 있는 기법인 것으로 연구되어왔다.

브로드캐스트 방법을 통해서는 모든 클라이언트의 요구들을 동시에 만족시키는 어렵다. 브로드캐스트 방법은 서버가 전송해야할 데이터를 시간 순서에 따라 주기적으로 배열하여 전송한다. 때문에 클라이언트는 데이터 채널에 접속하여 원하는 데이터가 전송되기 까지 브로드캐스트 프로그램 스케줄에 따라 일정시간 기다려야 한다[1]. 이를 접근 시간(access time)이라고 정의하면 첫 번째 해결해야 할 문제는 전체 클라이언트에 대한 평균적인 접근 시간을 줄이는 브로드캐스트 프로그램을 스케줄링 하는 것이다. 즉, 전체 데이터에 대한 평균 접근 시간을 줄이는 브로드캐스팅을 위한 데이터 할당 문제이다[2].

기존에 다중채널에서 접근 시간을 줄이기 위한 많은 연구가 진행되어왔다[3-7]. 각 채널에 할당된 데이터의 개수와 접근확률의 합에 따라서 전체 데이터에 대한 평균 접근시간이 계산되고 이를 최소로 하는 데이터 할당이 목표였다. 하지만 기존 방법들의 문제는 각 채널에 할당된 데이터들은 각각의 접근확률이 다름에도 같은 전송주기를 갖게 된다는 것이다. 이러한 문제를 해결하기 위해 채널에 데이터의 할당 이후에 각 채널 내의 데이터 간 접근 확률의 차이를 반영하는 연구가 제시되었다[8].

브로드캐스트 디스크 구성으로 각각의 채널들에 대한 평균 접근시간을 단축하기 위해서는 데이터 간 접근 확률의 차이가 변수가 된다. S2AP 알고리즘의 데이터 할당방법[8]은 접근 확률순서대로 정렬된 데이터를 순서대로 묶어서 채널에 할당하기 때문에 데이터 간 접근확률의 차이는 가장 작은 상태가 된다. 이러한 할당은 접근 시간 단축의 효율을 떨어뜨리게 된다. 왜냐하면 접근확률 차이가 작은 데이터들에 대한 브로드캐스트 디스크 구성으로 각 데이터들의 반복주기 차이가 작게 나타나기 때문이다. 본 논문에서는 각각의 채널에서의 브로드캐스트 디스크 구성을 통해 다중 채널에 대한 평균 접근

시간을 효율적으로 단축시키는 새로운 데이터 할당방법을 제시하고자 한다.

제안하는 방법은 기존의 방법과는 다른 데이터 할당방법을 고려한다. 접근확률에 따라 정렬된 데이터들은 인접 데이터와 같은 채널에 할당되지 않도록 하여 데이터 간의 접근 확률 차이가 기존 방법의 데이터 할당보다 커지게 한다. 이런 방법으로 데이터 접근 확률의 분포를 수치화하여 이 수치를 최대화하는 데이터 할당을 하고자 한다. 이후에 채널에 할당된 데이터들의 브로드캐스트 디스크 구성을 통해 다중 채널에서의 전체 평균 데이터 접근시간을 단축시키고자 한다. 클라이언트가 원하는 데이터가 어떠한 채널에 있는지를 미리 안다면 전체 평균 데이터 접근시간 크게 줄일 수 있다. 기존의 데이터 할당방법은 접근 확률에 따른 순차적인 데이터 접근이 가능하며 인덱스 기술이 추가되지 않더라도 접근 시간에 대한 성능이 어느 정도 보장되었다. 그러나 제안하는 데이터 할당 방법에서는 접근확률 순서로 차례대로 채널에 할당되지 않기 때문에 이러한 성능보장이 어렵다. 제안하는 데이터 할당방법의 성능을 위하여 인덱스 방법의 연구가 필수적이기 때문에 적합한 인덱스 기술도 설명하고자 한다[9]. 활용하는 인덱스 방법은 독립적인 인덱스 채널이 필요하며, 인덱스 정보도 포인터 하는 데이터와 같은 주기의 반복할당이 이루어진다.

본 논문의 2장에서는 기존에 연구들과 그들의 문제점을 살펴본다. 3장에서는 제안하는 데이터 할당방법을 소개하고 4장에서는 제안하는 방법에 필요한 인덱스 방법을 설명한다. 5장에서는 제안하는 방법의 성능을 평가하며, 마지막 6장에서는 결론을 맺도록 하겠다.

2. 관련연구

이번 장에서는 기존의 연구들을 간략하게 살펴보도록 하겠다. 또한, 각 연구에서 제시한 데이터 할당 방법의 특징과 문제점을 분석해 보도록 하겠다.

기존에 연구들을 설명하기 위해 다음의 파라미터(parameter)를 정의한다. 전송하려는 데이터 아이템의 전체 개수는 N 이고 전체 채널의 개수는 K 가 된다. 각 데이터 아이템 d_i 는 접근 확률(access probability)을 가지게 되는 데, 이것을 p_i 로 표기한다. 모든 데이터 아이템의 접근 확률 p_i 를 합한 값은 $\sum_{i=1}^N p_i = 1$ 이다.

기존의 할당 방법들은 데이터를 각각의 접근확률에 따라 내림차순으로 정렬하고 그 데이터들을 적당한 방법으로 총 채널 개수에 맞게 차례로 할당한다. 이때 분할 방법에 따라 접근시간에 대한 성능차이가 크기 때문에 이를 단축하는 분할 방법을 찾는 것이 기존 연구들의 목표가 되었다. 기존의 연구들은 다음 식으로 평균 접근시간을 구한다. 데이터 아이템 d_i 의 채널을 통해 받기 위하여 기다려야 하는 시간 기대치를 w_i 라고 하자. 그러면 각각 p_i 의 접근 확률을 갖는 데이터 아이템들은 w_i 만큼 지연되므로, 평균 기대 지연값(average expected delay)은 p_i 와 w_i 의 곱을 모든 i 에 대하여 합한 값이 된다[4].

단일 채널 평균 기대 지연값 SCAED(single channel average expected delay) \bar{w}

기존의 연구들은 각 채널의 SCAED를 모두 더하여 다중 채널에 대한 평균 접근시간을 구하고자 하였다. 따라서 [4]의 방식의 경우, 전체 K 개의 다중 채널에 대한 평균 기대 지연값은 다음과 같다. 여기서 C_j 는 j 번째 채널을 나타내며 N_j 는 채널 C_j 에 할당된 데이터 아이템의 개수를 의미한다.

다중채널 평균기대 지연값 MCAED(Multi-channel average expected delay)

기존의 할당 방법들을 간략하게 비교한 것이 다음의 표 1이다. 앞으로 [4]의 할당방법은 편의를 위해 Y. Greedy 알고리즘으로 표기하겠다. FLAT[3] 방법은 다음의 표 1과 같이, 접근 확률을 전혀 반영하지 않고, 데이터 아이템의 개수에 따라 채널에 데이터를 할당하는 방식이고, Bin-packing[5]은 데이터의 접근 확률을 반영하지는 않지만, 데이터 개수에 따른 지연을 고려하지 않고 단순한 접근 확률의 합에 따른 구분에 의해 데이터를 할당한다. 따라서 각 두 데이터 할당방법의 알고리즘 수행시간은 빠르지만, 평균 데이터 접근 시간에 대한 성능에서는 효과적이지 못하다. 물론 접근확률을 데이터 할당에 반영하는 Bin-packing이 FLAT보다 우수한 성

능을 발휘한다. VF^k [6]와 Y_greedy[4]와 비교해서 설명하는 DP(Dynamic Programming)[4]방법은 접근 확률에 따라 정렬된 전체 데이터에 대하여 전체 멀티채널에 대한 평균 예상 지연 시간을 최소로 하는 데이터 할당을 동적 프로그래밍 알고리즘 기법을 이용하여 찾아낸다. 즉, MCAED를 최소로 하는 분할지점을 DP방법에 의해 찾아내는 것이다. DP방법은 다중 채널에서의 최소의 평균 예상 지연 시간을 갖는 최적의 할당을 찾아낼 수 있다는 장점이 있다. 기존의 FLAT[3], BP[5], VF^k [6] 등의 알고리즘과 비교하면 접근 시간 성능에 있어서 가장 우수한 성능을 보장한다. 다만, 알고리즘의 계산 시간과 수행에 $O(KN^2)$ 과 $O(KN)$ 의 저장 공간을 필요로 하기 때문에, 다른 알고리즘에 비해 계산 시간과 저장 공간에 대하여 현저히 떨어지는 성능을 보인다.

VF^k [6] 알고리즘은 트리 구조를 이용하여 평균 데이터 접근 시간을 줄이려는 알고리즘으로, DP와 유사하게 높은 평균 기대 지연값에 대한 성능을 보이지만, DP와 마찬가지로 알고리즘 수행시간이 오래 걸리는 복잡한 알고리즘으로, 이후 제안하는 알고리즘과 평균 기대 지연값에 대한 성능 비교를 위한 비교 방법이 될 것이다.

동등 채널에 할당된 데이터 간에도 접근확률의 차이를 브로드캐스트 디스크 구성을 통하여 반영하고자 S2AP[8]라는 기존 연구가 있다. S2AP 방법을 설명하기 위하여 브로드캐스트 디스크 구성에 대한 접근시간 계산식을 다음에서 먼저 설명한다. 각 채널에서 데이터의 상대적인 접근 확률에 따라 반복을 고려한다면 이에 대한 평균 예상 지연시간을 계산하는 것이 필요하다. 먼저 데이터의 접근확률 분포가 지프분포에 가깝게 구성된 경우의 단일 채널에 대한 평균 예상지연 시간을 계산하기 위해서 다음의 파라미터를 정의한다. d_i 를 i 번째 데이터 아이템 i 를 제외한 나머지 브로드 캐스트 길이, t_i 를 아이템 d_i 의 반복횟수라 정의한다. S2AP 알고리즘은 동등 채널의 데이터들을 반복주기에 따라 브로드캐스트 디스크로 구성한다. 이에 따라서 [8]에서 제시한 단일 채널에 대한 평균 예상 지연 시간은 다음과 같다.

표 1 다중방송 채널 환경의 데이터 할당 알고리즘들의 비교

	Flat	Bin-packing	VF^k	Y_greedy
데이터 할당방법	$\frac{N}{K}$ (N : 데이터의 갯수 K : 채널의 개수)	$bin = \frac{1}{K}$ 각 채널의 데이터 확률의 합이 bin이 되도록 할당	데이터 접근 확률을 바탕으로 할당 트리를 구성	평균 기대 지연값을 최소로 하는 지점을 분할 지점으로 선정
복잡도	$O(K)$	$O(N \log N)$	$O(KN^2 \log K)$	$O((N+K) \log K)$
비교	클라이언트의 접근 확률 분포가 편향될수록 접근 시간이 커진다.	접근 확률이 반영되어 있으나, 알고리즘이 단순하다.	DP와 비슷한 성능을 보이나 시간복잡도가 크다.	DP와 VF^k 보다 수행시간이 빠르면서 비슷한 성능을 보여준다.
공통	접근확률의 순서로 정렬된 데이터들은, 전송할 채널 개수에 따라서 집합으로 구성되어 순서대로 채널에 할당된다. 즉, 전체 데이터를 분할하는 지점이 어디냐에 따라서 접근 시간의 성능이 좌우된다.			

$$S2AP-SCAED = \sum_{i=1}^N w_i p_i = \sum_{i=1}^N \frac{w_i}{t_i} p_i$$

S2AP 알고리즘 또한 각 채널의 S2AP-SCAED를 모두 더함으로써 전체 채널에 대한 MCAED를 구하고자 하였다. [8]의 방법에 따라 모든 채널에 대한 브로드캐스트 디스크 구성이 반영된 평균 접근 시간은 다음과 같다.

전체 채널이 멀티 디스크로 구성된 경우의 다중 채널 평균 기대 지연값

$$S2AP-MCAED = \sum_{i=1}^K \sum_{d \in C_i} \frac{w_i}{t_i} p_i$$

S2AP 알고리즘의 기본 토대는 먼저 Y_greedy 알고리즘으로 MCAED를 단축시키는 채널에의 데이터 분할을 실시한다. 이후 브로드캐스트 디스크 구성을 통하여 동등 채널에 대한 데이터 접근확률 차이를 반영한 브로드캐스트 프로그램의 스케줄링을 통하여 Y_greedy 방법보다 더 접근시간을 단축시키고자 하는 것이다. 다음 장에서는 제안하는 데이터할당 알고리즘에 대해 설명하고자 한다.

3. ZGMD 데이터할당 알고리즘

이 장에서는 제안하는 데이터 할당 방법에 대해 설명하고자 한다. 먼저 기존의 방법과는 다른 데이터간의 접근 확률의 분포를 최대화하는 할당방법을 설명한다. 다음에는 채널에 할당한 데이터들을 브로드캐스트 디스크로 재구성하는 방법과 접근시간 계산에 대해서 설명하고, 본 논문에서 제시하는 방법과 알고리즘에 대해 자세히 살펴보도록 한다.

3.1 환경 변수의 정의

본 논문에서는 다중 방송 채널에서 데이터를 브로드캐스트 환경에서의 데이터 할당방법을 설명하기 위해 다음의 환경을 가정한다.

표 2 제안하는 시스템 환경에서의 파라미터

N :	전체 데이터 아이템의 개수 ($1 \leq i \leq N$)
K :	데이터를 전송하기 위한 채널의 개수
p_i :	데이터 d_i 에 대한 접근확률 ($\sum_{i=1}^N p_i = 1$)
C_i :	i 번째 위치한 채널
P_j :	C_j 에 할당된 데이터 아이템의 접근확률의 총합 ($P_j = \sum_{d \in C_j} p_i$)
t_i :	데이터 아이템 d_i 의 반복횟수
N_i :	채널 C_i 에 할당된 데이터 아이템의 개수
N'_i :	채널 C_i 에서 반복횟수가 적용된 한 사이클에 포함된 데이터 아이템의 개수 ($N'_i = \sum_{d_j \in C_i} t_j$)

3.2 인덱스 기반의 데이터 할당방법

인덱스의 사용을 가정하고 각 다중채널에서 브로드캐스트 디스크 구성을 한다고 한다면 데이터 할당에 대한 중요한 변수는 각 데이터 아이템의 접근확률의 차이가 된다. 데이터간 접근확률의 차이가 클수록 브로드캐스트 디스크 구성의 결과 접근시간의 단축은 효과적으로 일어나기 때문이다. 각 데이터 아이템의 접근확률 차이의 수치적 표시를 위해 지프의 분포이론을 사용하고자 한다. 지프분포의 정의를 본 환경에 적용하면 다음과 같다. 데이터 d_i 에 대한 클라이언트의 접근 빈도를 f_i 라 하고, 전체 데이터를 클라이언트의 접근빈도에 따라서 내림차순 정렬시켰다고 할 때 접근 빈도에 따른 순위를 r_i , 전체 데이터의 개수를 N 이라 나타낸다. 여기에, 지프 분포 (zipf distribution)의 상태를 나타내는 지프 팩터 θ 를 적용한다면 데이터에 대한 접근빈도 f_i 는 지프의 법칙에 의한 공식을 적용하여 다음의 식으로 나타낸다.

$$f_i = C * 1/r_i^\theta$$

이때 상수 C 는 지프 분포에서 접근빈도 계산을 위한 상수이며 $C \cong N/10$ 로 나타낼 수 있다. 접근빈도의 차이는 접근 확률로 나타낼 수 있다. 데이터 d_i 의 접근 확률 즉 선호도 p_i 는 다음의 식으로 나타낸다.

$$p_i = \frac{f_i}{F_N} \text{ where } F_N = \sum_{i=1}^N f_i$$

이때, 정렬된 데이터들 사이의 접근확률 차이 $p_i - p_{i+1}$ 은 데이터 접근 빈도의 분포 정도를 나타내는 θ ($0 \leq \theta \leq 1$)값이 증가할수록 커진다. 제안하는 데이터 할당방법의 기본 아이디어는 지프분포의 정의를 역으로 이용하여 각 채널 C_j 에 할당된 데이터들의 접근확률이 주어졌을 때, 각 C_j 채널별로 할당된 데이터들의 지프 분포의 상태를 나타내는 지프 팩터 θ_j 를 수치화하는 것이다.

정리 1. 한 개의 채널에 데이터 선호도에 따라서 내림차순으로 정렬된 데이터가 1부터 N 까지의 할당된 상황을 가정한다. 데이터 d_i 에 대한 접근 빈도를 f_i , 접근 빈도의 내림차순에 따른 순위를 r_i 로 나타낸다. 각 데이터 d_i 의 접근확률이 p_i ($1 \leq i \leq N$)와 같이 주어졌다면 인접하는 두 데이터 d_j 와 d_{j+1} 의 지프 분포 상태는 지프 팩터 θ_j ($1 \leq j \leq N-1$)로 나타낼 수 있다.

증명. 데이터 d_i 의 접근확률 p_i 에 대해서는 지프의 법칙에 따라 $p_i = C/(r_i^\theta * f_N)$ 같이 나타낼 수 있다. $r_1 = 1$ 이고 상수 C 값을 알기 때문에 $p_1 = \frac{C}{f_N}$ 이고 따라서 $f_N = \frac{C}{p_1}$ 으로 구할 수 있다. $p_2 = \frac{C}{2^{\theta_1} * f_N}$ 이고 C 값과

f_N 을 구하였기 때문에, d_1 와 d_2 의 지프 분포 상태를 나타내는 지프 팩터 θ_1 를 구할 수 있다. d_2 와 d_3 의 지프 분포 상태를 나타내는 지프 팩터 θ_2 는 $p_3 = \frac{C}{3^{\theta_2} * f_N}$ 으로부터 구할 수 있다. 이러한 방법으로 d_j 와 d_{j+1} 의 지프 분포 상태를 나타내는 지프 팩터 $\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7, \dots, \theta_{N-1}$ 값을 모두 구할 수 있다.

정리 1에 의하여 n개의 데이터가 한 채널에 할당된 경우, 각각의 θ_j 를 모두 구하고 이에 대한 평균값으로 단일 채널의 평균지프분포 정도를 구할 수 있다. 즉, 이 채널에 대한 평균 지프 분포 정도는 다음과 같다.

Single channel Average Zipf distribution

$$\text{degree(SCAZD)} = \left(\sum_{j=1}^{n-1} \theta_j \right) / (n-1)$$

$\theta_j = \log\left(\frac{C}{p_{j+1} * f_N}\right) / \log(r_{j+1})$ 이므로 SCAZD는 아래와 같이 구해진다.

$$\text{SCAZD} = \left(\sum_{j=1}^{n-1} \log\left(\frac{C}{p_{j+1} * f_N}\right) / \log(r_{j+1}) \right) / (n-1)$$

K개의 채널에 대해 각 채널 C_i 의 모든 SCAZD를 구하면 이 합을 평균을 구함으로써 전체 다중 채널에 할당된 평균 지프 분포 정도를 구할 수 있다. K개의 채널이 제공되고, 각 채널에 n_i 개씩의 데이터가 할당되었을 때, 다중 채널에 대한 평균 지프 분포 정도는 다음과 같다.

Multi channel Average Zipf distribution

$$\begin{aligned} \text{degree(MCAZD)} &= \left(\sum_{i=1}^K \left(\sum_{j=1}^{n_i-1} \theta_j \right) / (n_i - 1) \right) / K \\ &= \left(\sum_{i=1}^K \left(\sum_{j=1}^{n_i-1} \log\left(\frac{C}{p_{j+1} * f_{n_i}}\right) / \log(r_{j+1}) \right) / (n_i - 1) \right) / K \end{aligned}$$

전체 평균 지프분포 정도를 최대화 하는 데이터 할당 방법의 첫 번째로 Dynamic Programming 기법을 이용한 방법이 있을 수 있다. 즉 K개 채널에 대한 N개의 데이터를 전체 평균 지프 분포 정도를 가장 크게 하는, DP 방법의 최적의 해를 $optzipf_{1,K}$ 라 한다면, 다음의 식으로 나타낼 수 있다.

$$optzipf_{i,K} = \max_{l \in \{i, i+1, \dots, N-1\}} \left(\left(\sum_{i=1}^{n_K} \theta_i \right) / n + optzipf_{l+1, K-1} \right)$$

위의 DP 방법의 계산 수행시간은 $O(NK^K)$ 이다. 이 방법은 데이터 개수와 채널수가 급격히 증가하는 상황에서는 실제적이지 않다. 다음으로 Y_greedy 방법을 통해 알고리즘 수행시간을 효과적으로 단축시키면서 MCAZD의 성능을 DP방법에 근사시키는 알고리즘을 제안하고자 한다.

3.3 제안하는 다중채널에서의 데이터 스케줄링 알고리즘

제안하는 방법은 각 채널에서의 MCAZD를 최대화 하는 것에 근접하도록 greedy 방법으로 데이터를 할당 하며 이후 [8]에서 제안한 각 채널에서의 할당 횟수 결정 알고리즘에 따라 각 채널에 할당된 데이터에 대한 할당횟수를 구한 후 Acharaya method[1]에서 제안한 방식에 따라 브로드캐스트 디스크로 구성한다. 제안하는 스케줄링 알고리즘은 그림 1과 같다.

지프의 법칙에 따른 데이터 간의 선호도 분포 정도를 나타내는 θ 는 2개 이상의 데이터들 사이에서 측정 가능하다. 따라서 제안하는 greedy 할당을 위한 초기 상태는 선호도 순으로 정렬된 데이터들이 제공된 채널의 수 만큼 순서대로 하나씩 할당된 상태이다. 다음으로 내림 차순 정렬된 순서에서 할당된 다음의 데이터부터 MCAZD를 최대화 하는 greedy 할당을 시작한다. 데이터 아이템 $d_i (i \geq K)$ 는 K개의 채널 중 계산된 MCAZD를 최대화 하는 채널에 할당된다. 데이터 아이템 d_i 가 할당된 이후 계속해서 d_{i+1} 은 다시 계산된 MCAZD가 최대가 되는 채널에 할당된다. 이과정은 정렬된 순서의 마지막 데이터가 할당될 때까지 계속된다. 데이터들이 제공된 각 채널에 할당된 이후, 브로드캐스트 디스크 구성은 각 채널에서 독립적으로 이루어진다. 브로드캐스트 디스크 구성으로 평균 지연시간의 최대한의 단축을 위해 먼저 [8]에서 제안한 단일채널에서 상대적 데이터 접근확률에 따라 최적의 반복횟수를 찾는 계산 방법을 사용할 것이다. 이후 각 채널별로 기존의 S. Acharaya에

Algorithm: ZGMD

input: a set of N unit sized items(d_i) sorted by the decreasing order of their access probabilities
 K = the number of data channels($C_j, 1 \leq j \leq K$)
 p_i = access probability of data d_i

Output: broadcast program of data items allocated.

MCAZD_j = computed value when d_i allocated C_j by expression (8)

N_j = the number of data in channel j

Phase 1. <Allocation phase at K channels>

for(i=1; i<=K; i++)

 Allocate the data d_i to the data channel C_i ;

max = 1;

for(i=K; i<=N; i++) {

 for(j=1; j <= K; j++){

 Compute MCAZD_j ;

 if(j>1 && (MCAZD_j > MCAZD_{j-1})) max = j;

 }

 Allocate the data d_i to the data channel C_{max} ;

 }

end

Phase 2. < Construction phase of broadcast disk in each channel>

for (i=1; 1<=K; i++) {

 for (j=1; j<=N_i; j++)

 Compute t_j for data item d_i by the allocation number

 determination algorithm proposed in [7];

 Construct Broadcast Disk Program for channel C_i

 by Acharaya method in [1];

 }

end

그림 1 제안하는 ZGMD 데이터 스케줄링 알고리즘

의해 제시된 방법[1]을 토대로 브로드캐스트 디스크 프로그램 구성한다.

그림 2는 표 3의 데이터에 대한 ZGMD 할당의 초기 수행 모습을 보여주며, 그림 3은 초기 할당 이후의 계속 해서 진행되는 ZGMD 알고리즘의 데이터 할당의 모습을 보여준다. 접근확률에 의해 정렬된 16개의 데이터 아이템 중 상위 E, P, A는 3개의 채널에 순서대로 각각 하나씩 할당되는 것이 ZGMD 알고리즘의 초기 수행 모습이다. 데이터 E, P, A가 할당되어 초기화된 상황에서 데이터 D는 3개의 채널중 MCAZD의 값이 최대인 채널 1에 할당된다. 데이터 D가 채널 1에 할당될 경우의 MCAZD는 0.19, 채널 2에 할당될 경우의 MCAZD는 0.15, 채널 3에 할당될 경우의 MCAZD는 0.07이 된다. 즉, 데이터 아이템 D가 채널 1에 할당되는 것이 MCAZD 0.19로 가장 높은 값을 만족한다.

그림 4는 다시 데이터 E, P, A, D가 할당된 상황에서 데이터 N을 할당하는 모습을 보인다. 3개의 채널에서의 MCAZD를 각각 계산하여 역시 이것이 최대가 되는 채널 2에 할당되는 것이다.

그림 5는 데이터 아이템의 모든 할당이 이루어진 후 브로드캐스트 디스크 구성이 이루어진 최종 결과를 나타낸다. greedy 알고리즘에 의해 채널에의 데이터 할당이 끝나게 되면, 동등 채널에 포함된 데이터 아이템들의 접근확률 차이에 따라 반복횟수 t_i 를 구하게 된다. 채널

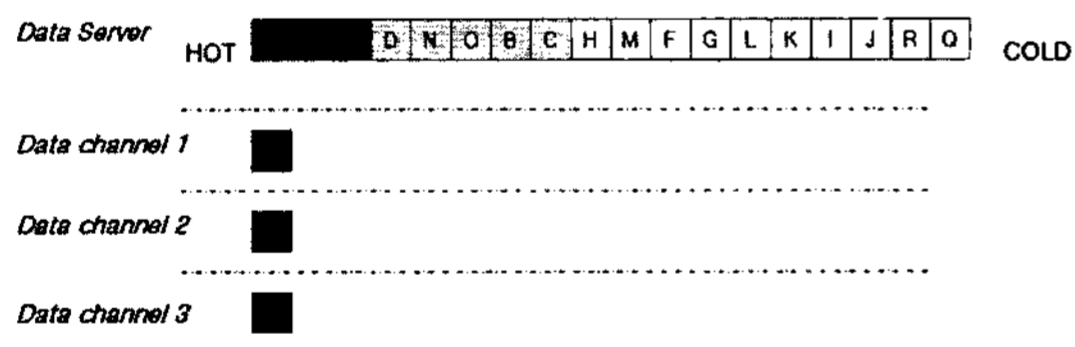


그림 2 ZGMD 할당을 위한 첫 번째 할당 단계

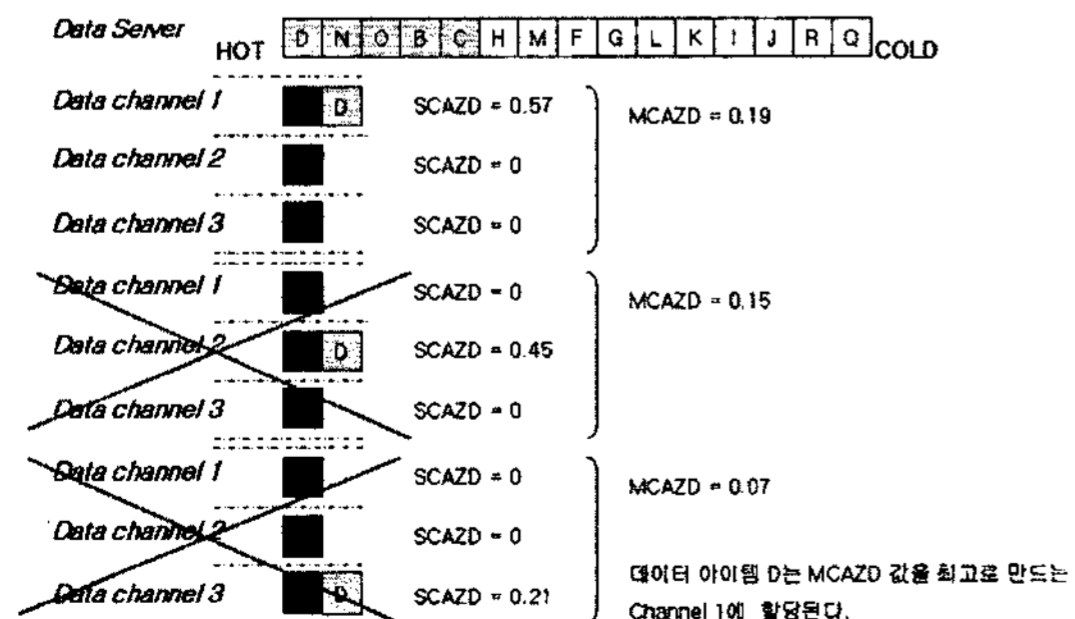


그림 3 ZGMD 할당 과정의 예제-1

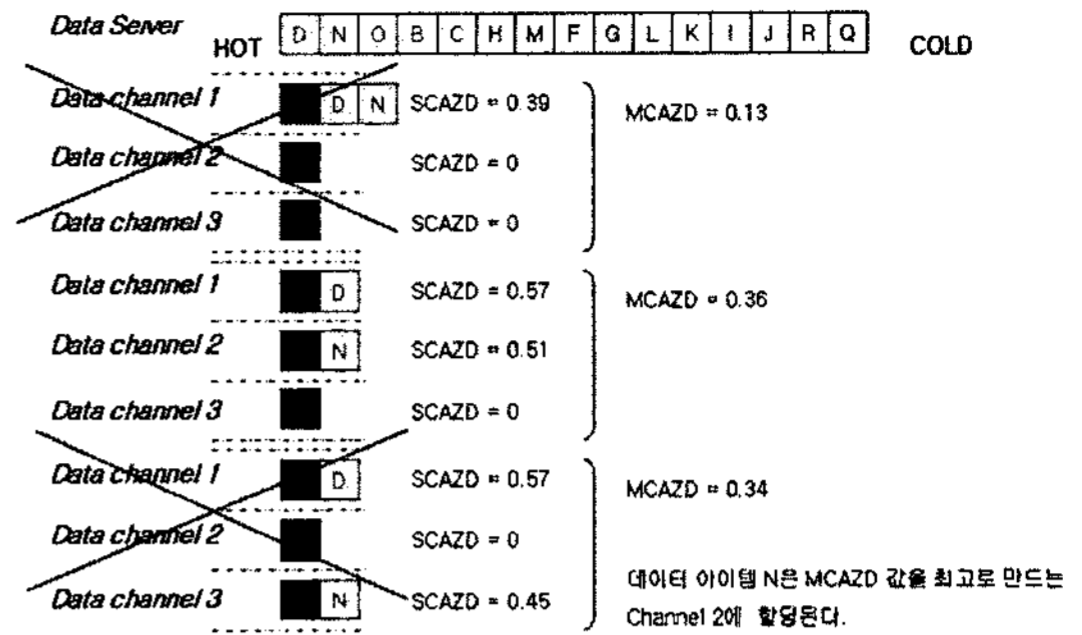


그림 4 ZGMD 할당 과정의 예제-2

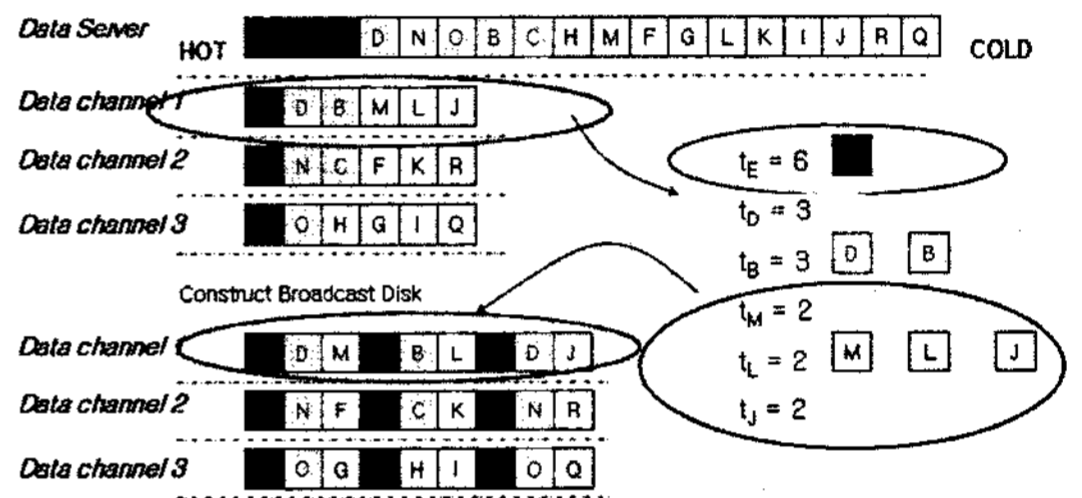


그림 5 ZGMD 할당 과정의 최종결과

1에 포함된 데이터 아이템 E, D, B, M, L, J의 경우 각 t_i 는 S2AP의 반복도를 구하는 식에 따라서 각각 6, 3, 3, 2, 2, 2로 계산되었다. 데이터 아이템들은 반복주기에 따라 $s_1=\{E\}$, $s_2=\{D, B\}$, $s_3=\{M, L, J\}$ 의 집합으로 구성된다. 각 집합의 원소들은 반복도에 따라서 Chunks로 구성된다. 따라서 가장 Hot 데이터의 집합인 s_1 은 $\{E\}$, s_2 는 $\{D\}$ 와 $\{B\}$, s_3 는 $\{M\}$ 과 $\{L, J\}$ 의 Chunks를 가지게 되고 s_1, s_2, s_3 의 각 Chunks는 각 마이너 주기에 한 번씩 할당되어 $\{E, D, M\}$ 의 1 마이너주기, $\{E, B, L\}$ 의 2 마이너주기 등의 과정으로 각각의 마이너 주기를 만들어간다. 이러한 과정을 거쳐 한 채널에 대해서 최종 메이저주기가 만들어지고 이것이 모든 채널에서 이루어진 결과가 그림 5이다.

ZGMD 알고리즘의 수행시간은 $O(N \log K)$ 이다. 이것은 DP방법을 이용한 MCAZD를 찾는 알고리즘보다 우수하게 빠른 시간이며, 기존의 VF^K 보다도 빠른 수행시간의 성능을 보인다. ZGMD 데이터 할당 방법에서는 [8]의 방식으로 MCAED를 계산할 수 없다. 왜냐하면 각 정렬된 데이터들은 순서대로 채널에 할당되지 않았기 때문이다. 즉, 인덱스가 없는 접근확률에 따른 순차적인 검색이 어렵다. 따라서 별도의 인덱스 기술이 적

표 3 18개의 정렬된 데이터와 접근확률

E	P	A	D	N	O	B	C	H	M	F	G	L	K	I	J	R	Q
0.25	0.18	0.14	0.1	0.09	0.06	0.06	0.06	0.05	0.04	0.04	0.03	0.02	0.02	0.01	0.01	0.01	0.01

용되지 않았을 때, 제안하는 데이터 할당 방법의 다중 채널 평균 기대 지연값은 다음 정리와 같이 계산된다.

$$\sum_{i=1}^{N_1} \frac{N_1' - t_i}{t_i} p_i + N_1' * P_2 + \sum_{i=1}^{N_2} \frac{N_2' - t_i}{t_i} p_i + (N_1' + N_2') * P_3 + \sum_{i=1}^{N_3} \frac{N_3' - t_i}{t_i} p_i + \dots + \sum_{j=1}^{K-1} N_j' * P_k + \sum_{i=1}^{N_k} \frac{N_k' - t_i}{t_i} p_i$$

제안하는 방법은 평균 접근시간이 기존의 FLAT, Y_greedy나 VF^K방법과는 다른 식에 의해 계산된다는 것을 보였다. 기존의 데이터 할당방법은 선호도가 높은 데이터는 상위채널에, 선호도가 낮은 데이터는 하위 채널에 할당함을 보장하였다. 그러나 제안하는 데이터 할당방법은 선호도가 높은 데이터와 낮은 데이터를 전체 채널에서와 유사한 비율로 각 채널에 골고루 할당한다. 위에서 소개한 접근시간의 계산은 선호도가 높은 데이터에 대해서 검색을 위해 마지막 채널까지 모두 검색해야 하는 부담을 가지고 있다. 결국 접근시간 성능을 위해서는 데이터가 어떤 채널에 있는지를 검색하기 위한 인덱스 기술이 필요하다. 다음 4장에서는 제안하는 데이터 할당방법의 다중채널 평균 기대지연의 성능을 향상시키기 위한 인덱스 기술을 제시할 것이며 인덱스 환경에서 제안하는 데이터 할당방법이 동작하는 모습을 설명할 것이다.

4. 접근시간 단축을 위한 인덱스 기술

이번 장에서는 제안하는 ZGMD 데이터 할당방법의 접근시간 성능 향상을 위한 인덱스 기술에 대하여 설명한다. 제안하는 데이터 할당방법의 접근시간 단축을 위해 기존의 SIRAH[9] 인덱스 할당 방법을 적용하고자 한다. SIRAH 알고리즘은 인덱스 정보만을 전송하는 독립적인 채널을 사용하고 접근 시간을 효율적으로 단축하는 인덱스 채널과 데이터 채널의 비율을 구한다. 따라서 제안하는 방법에서 필요한 빠른 위치 검색을 가능하게 해준다. 또한 인덱스의 전송 스케줄을 가리키는 데이터가 전송되는 주기와 같은 비율로 구성할 수 있도록 한다. 이것은 접근 확률에 따라서 선호도가 높은 데이터의 접근 지연을 보다 단축시키며 인덱스 트리를 효율적으로 사용하도록 한다. 그러나 ZGMD 할당방법은 SIRAH가 가정한 기존의 데이터 할당환경과 달리 동등 채널에서도 접근확률에 따라 데이터들의 반복주기가 다르게 반영되도록 하였다. 따라서 제안하는 데이터 할당 방법에 대해서 SIRAH의 인덱스 할당 방법을 적용하기 위해서는 새로운 구성방법이 필요하다. 이번 장에서는 SIRAH 인덱스 할당방법에 대한 설명과 제안하는 ZGMD

데이터 할당 방법에 SIRAH 인덱스 할당 방법을 적용하기 위한 방안을 제시하고자 한다.

4.1 인덱스기술 적용을 위한 가정과 시스템 환경

세 개의 채널과 접근확률 순으로 정렬된 데이터가 있다고 가정하자. Y_greedy 할당에서는 접근확률에 따라 순서대로 1채널부터 3채널까지 데이터를 할당한다. 원하는 데이터를 검색하기 위해서는 채널 1부터 3까지 차례대로 검색을 하게 된다. 사전에 원하는 데이터가 어느 채널에 있는지 안다면 모든 채널을 차례로 검색할 필요 없이 원하는 채널만 검색하여 접근시간을 단축할 수 있다.

그림 6은 제안하는 방식의 스케줄링 예제이다. 접근확률이 높은 데이터와 낮은 데이터들은 3개의 채널에 고르게 할당된다. Y_greedy 같은 순차적인 검색은 제안하는 환경에서는 어렵다. 예를 들어 접근확률 순위 3의 데이터 A에 접근하길 원한다고 가정하자. 기존의 Y_greedy 할당 환경에서는 A보다 앞선 데이터 E와 P, 두 개의 데이터를 검색하는 비용으로 데이터 A에 접근할 수 있다. 그러나 제안하는 할당 환경에서는 채널 1과 채널 2의 데이터를 모두 검색하는 시간 지연에 대한 비용을 치르고서야 데이터 A가 있는 채널 3에 접근 할 수 있게 된다. 그러나 데이터 A가 채널 3에 있다는 정보를 클라이언트가 사전에 알고 있다면 지연 없이 바로 채널 3에 통해 데이터 A에 접근할 수 있다.

SIRAH의 인덱스 할당방법을 설명하기 위하여 다음의 용어를 설명한다. initial probe는 브로드캐스트를 듣기 시작한 시점을 나타낸다. 접근 시간(access time)은 initial probe로부터 원하는 데이터를 받을 때까지 걸리는 평균 시간이다. 원하는 데이터를 받기 위해서는 인덱스에 먼저 접근하는 것이 필요하다. 인덱스 접근 시간(index access time)은 이때 initial probe로부터 원하는 데이터를 가리키는 인덱스를 받을 때까지 걸리는 평균 시간을 말한다. 따라서 접근 시간에는 인덱스 접근 시간이 포함된다.

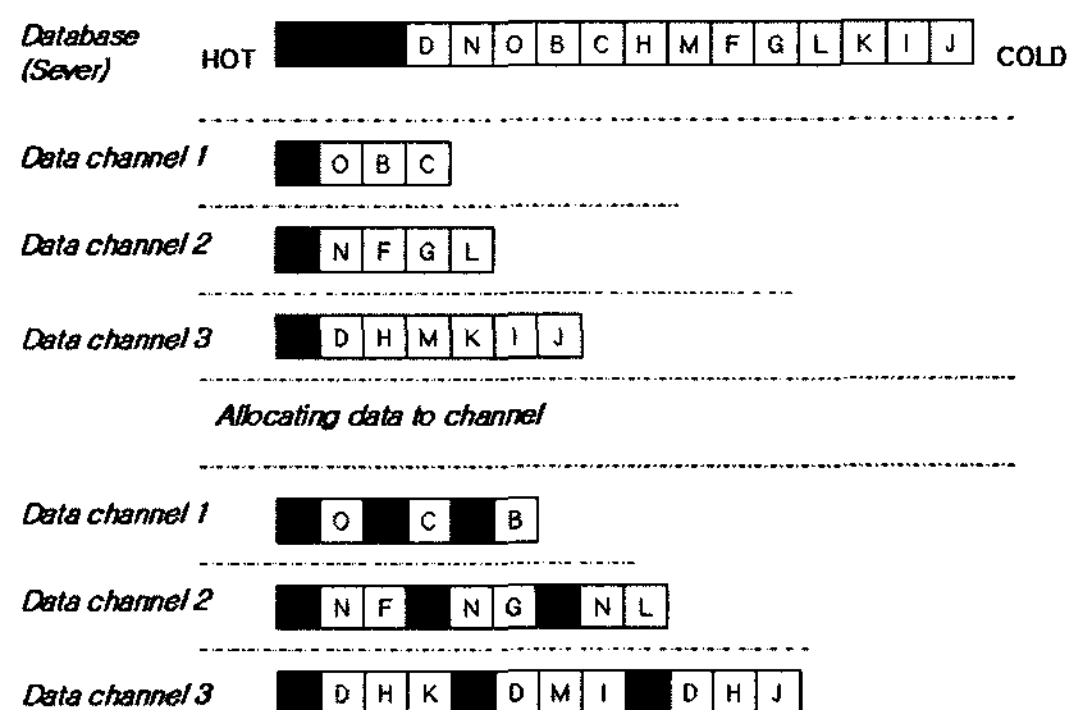


그림 6 ZGMD 방법으로서의 데이터 할당 예제

4.2 SIRAH 인덱스 할당방법

멀티채널 환경에서의 브로드캐스트를 위한 인덱스 방법 중 트리를 사용한 인덱스 방법은 서버에서의 인덱스 유지비용은 있지만, 클라이언트에게 적은 튜닝시간을 제공하며 데이터 접근확률에 따라 인덱스 접시시간도 유사한 비율로 적용할 수 있다는 장점을 가진다[10,11].

데이터 접근빈도에 따라서 인덱스 트리를 구성하여 데이터 검색에 대한 지연 시간을 단축하려는 연구들은 기존에 이미 많이 제시되었다. 그 중에 독립적인 인덱스 채널을 이용하는 기술은 Alphabetic 호프만 트리 기반의 접근 방법[11]이 있다. 그러나 이 방법은 트리 깊이만큼 개수의 인덱스 채널을 이용하기 때문에 채널 자원의 낭비가 크다. 인덱스 채널의 증가는 데이터 채널의 감소를 의미하며, 데이터 채널 감소로 인한 데이터의 접근 지연 시간의 증가는 큰 부담이 된다. 따라서 데이터의 기대 접근시간의 단축을 위한 최적의 인덱스 채널의 수를 찾고, 제공된 인덱스 채널에 데이터가 할당되는 비율과 같은 정도로 인덱스를 할당하려는 SIRAH 방법이 제시되었다[8].

하지만 SIRAH 알고리즘을 제안하는 할당 방법에 바로 적용시키지 못하는 이유는 Index Set의 구성과 반복주기 계산의 문제 때문이다. SIRAH는 동등 채널에 포함된 데이터들이 모두 같은 반복주기를 가질 것을 가정한다. 따라서 Index Set에 포함되는 원소는 모두 같은 채널에 속한 데이터들의 인덱스들이다. 그러나 제안하는 데이터 할당 방법은 동등 채널의 데이터들이라도 모두 접근 확률에 따라 서로 다른 반복 주기를 갖는다. SIRAH을 적용하기 위해서는 같은 채널의 데이터들이 아닌 반복주기가 같은 데이터들로 Index Set을 구성하는 것이 필요하다. 또한 Index Set의 구성이 달라짐으로 인해서 Index Set의 반복주기를 계산하는 것과 스케줄링의 방법에 있어서 보완이 필요하게 된다.

4.3 제안하는 데이터 할당방법에 대한 SIRAH의 적용

제안하는 데이터 방법에 따라서 데이터는 그림 6과 같이 채널에 할당된 후 브로드캐스트 디스크로 구성된다. 이에 대한 인덱스 구성 트리는 그림 7과 같다. 제안하는 방법은 데이터 채널에서 브로드캐스트 되는 데이터의 반복주기를 계산하여 같은 반복주기를 갖는 데이터를 집합으로 묶는다. 데이터 아이템 d_i 의 반복주기 r_i 는 다음과 같이 계산된다.

$$r_i = \frac{t_i}{N_j} \quad (d_i \in C_j, C_j \text{는 } j\text{-번째 채널}(1 \leq j \leq K),$$

$$N_j = \sum_{d \in C_j} t_i)$$

GR을 반복주기에 따라 묶은 데이터의 집합이라 정의한다. R_i 는 GR_i 에 속한 데이터들의 반복주기이다. 그림 6의 예제에 따르면 데이터 아이템 E는 총 6의 브로드캐

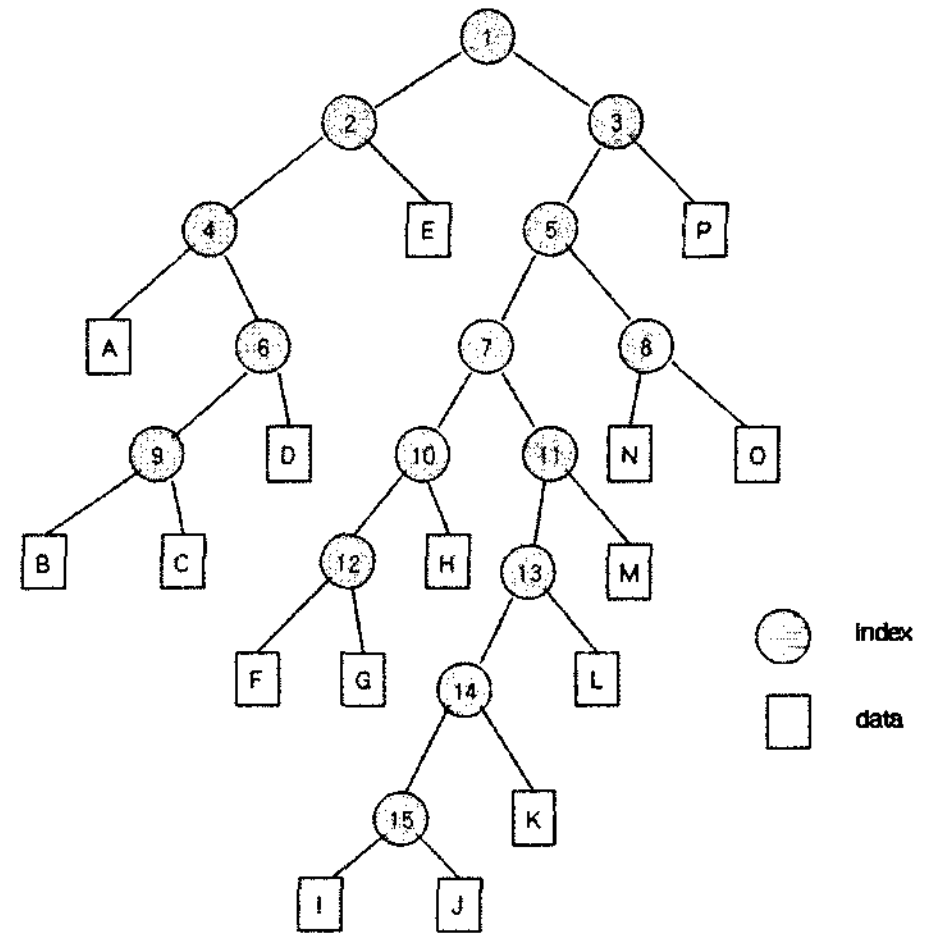


그림 7 Alphabetic Huffman tree를 이용한 인덱스 트리를 구성 예제

$GR_1 = \{E\}$	$R_1 = 1/2$
$GR_2 = \{P\}$	$R_2 = 1/3$
$GR_3 = \{A, D\}$	$R_3 = 1/4$
$GR_4 = \{N\}$	$R_4 = 1/5$
$GR_5 = \{O, B, C\}$	$R_5 = 1/6$
$GR_6 = \{H, M\}$	$R_6 = 1/8$
$GR_7 = \{F, G, L\}$	$R_7 = 1/9$
$GR_8 = \{K, I, J\}$	$R_8 = 1/12$

그림 8 GR 집합의 구성 예제

표 4 index list의 구성 예제

Data Set	index list
$GR_1 = \{E\}$	<1, 2>
$GR_2 = \{P\}$	<3>
$GR_3 = \{A, D\}$	<4, 6>
$GR_4 = \{N\}$	<5, 8>
$GR_5 = \{O, B, C\}$	<9>
$GR_6 = \{H, M\}$	<7, 10, 11>
$GR_7 = \{F, G, L\}$	<12, 13>
$GR_8 = \{K, I, J\}$	<14, 15>

스트 프로그램 길이 중 3회 반복된다. 즉, 1/2의 반복주기를 갖는다. 반복주기가 가장 큰 값부터 내림차순으로 GR을 구성하도록 한다. 그림 6에서 할당된 데이터를 반복주기에 따라 GR로 구성한 결과는 그림 8과 같다. 이때 각 데이터를 포인터 하는 인덱스 트리에 의한 index list는 다음과 표 4와 같이 구성된다.

제안하는 데이터 할당에서는 브로드캐스트 디스크 기술에 따라 데이터 반복주기가 같은 데이터 집합의 구성



그림 9 GR의 구성 예제

표 5 인덱스 접근시간 계산을 위한 파라미터

NG : 전체 GR_i 의 갯수
t_{G_i} : 전체 인덱스 그룹 할당 사이클 내에서의 GR_i 의 할당 횟수
r_i : GR_i 집합의 원소 데이터들의 반복도
N_{GT} : 한 사이클에 전송될 전체 GR_i 의 개수 $\sum_{i=1}^{NG} t_{G_i}$

이 SIRAH와는 달리 채널의 개수와 무관하게 이루어진다. 이러한 인덱스 집합을 데이터 채널에서와 마찬가지로 접근 데이터의 할당주기를 적용하여 최적의 반복 할당횟수를 구하고자 한다. 즉, 각 다중채널에서 데이터가 할당되는 비율을 적용하여 인덱스 그룹의 최적의 반복 횟수를 결정하는 것이다. 데이터 채널에서의 데이터 할당 비율에 따라 그림 9와 같은 데이터 그룹의 구성이 가능하다.

인덱스 리스트에 대한 평균 인덱스 접근시간을 나타내면 다음과 같다. 먼저 변수를 표 5와 같이 정의하도록 한다.

계속해서 GR 그룹들에 대하여 데이터의 반복주기를 반영한 인덱스 아이템 접근시간을 최소로 하는 t_{G_i} 를 구하고자 한다. 이것은 기존의 S2AP 알고리즘에서 브로드캐스트 디스크 구성을 적용한 데이터 아이템의 최적의 반복횟수를 계산하는 정리를 응용할 것이다. 정의한 변수에 따라서 데이터들의 반복주기를 적용한, 인덱스

그룹의 평균 기대 지연은 $\frac{N_{GT}}{2} \sum_{i=1}^{NG} \frac{r_i}{t_{G_i}}$ 이다[8].

위는 함수,

$$f(t_{G_1}, t_{G_2}, \dots, t_{G_{NG}}) = \frac{N_{GT}}{2} \left(\frac{r_1}{t_{G_1}} + \frac{r_2}{t_{G_2}} + \frac{r_3}{t_{G_3}} + \dots + \frac{r_{NG}}{t_{G_{NG}}} \right)$$

로 나타낼 수 있으며, 이것의 최소값은 산술·기하 평균 정리에 의해,

$$\frac{N_{GT}-t_{G_1} * r_1}{t_{G_1}} = \frac{N_{GT}-t_{G_2} * r_2}{t_{G_2}} = \frac{N_{GT}-t_{G_3} * r_3}{t_{G_3}} = \dots = \frac{N_{GT}-t_{G_{NG}} * r_{NG}}{t_{G_{NG}}}$$

일 때 최소값을 갖는다. 여기서, $\sum_{i=1}^{NG} r_i = k$ (상수)로 결정되므로,

$$2f-k = \left(\frac{N_{GT}-t_{G_1}}{t_{G_1}} + \frac{N_{GT}-t_{G_2}}{t_{G_2}} + \dots + \frac{N_{GT}-t_{G_{NG}}}{t_{G_{NG}}} \right)$$

이다. t_{G_i} 는 자연수이므로, 변수 N 개의 t_{G_i} 에 대한 산술 기하 평균을 적용하면,

$$2f-k \geq N \left(\frac{N_{GT}-t_{G_1}}{t_{G_1}} + \frac{N_{GT}-t_{G_2}}{t_{G_2}} + \dots + \frac{N_{GT}-t_{G_{NG}}}{t_{G_{NG}}} \right)^{\frac{1}{NG}}$$

이고 이 때, 등호는

$$\frac{N_{GT}-t_{G_1} * r_1}{t_{G_1}} = \frac{N_{GT}-t_{G_2} * r_2}{t_{G_2}} = \frac{N_{GT}-t_{G_3} * r_3}{t_{G_3}} = \dots = \frac{N_{GT}-t_{G_{NG}} * r_{NG}}{t_{G_{NG}}}$$

일 때 성립한다. 따라서 위의 식을 만족하는 $\frac{N_{GT}-t_{G_i} * r_i}{t_{G_i}} = m$ 으로 놓고, 실수 m 을 구한다면 모든 반복횟수 t_{G_i} 를 구할 수 있을 것이다. 위의 식에 따라서, 각 인덱스 리스트 그룹은 index 채널에서 브로드캐스트 될 최적의 반복횟수 t_{G_i} 가 결정되게 된다[8].

표 6 인덱스 리스트를 전송하기 위한 GR과 계산된 반복횟수

Data Set	index list	t_{G_i}
$GR_1 = \{E\}$	$\langle 1, 2 \rangle$	4
$GR_2 = \{P\}$	$\langle 3 \rangle$	4
$GR_3 = \{A, D\}$	$\langle 4, 6 \rangle$	2
$GR_4 = \{N\}$	$\langle 5, 8 \rangle$	2
$GR_5 = \{O, B, C\}$	$\langle 9 \rangle$	1
$GR_6 = \{H, M\}$	$\langle 7, 10, 11 \rangle$	1
$GR_7 = \{F, G, L\}$	$\langle 12, 13 \rangle$	1
$GR_8 = \{K, I, J\}$	$\langle 14, 15 \rangle$	1

각 그룹의 반복횟수가 결정되면 위와 같은 그룹의 인덱스를 각각의 반복횟수에 따라서 그림 10과 같이 브로드캐스트 하는 것이다.



그림 10 인덱스 리스트를 전송하기 위한 GR의 스케줄링 예제

이때, 반복횟수가 같은 그룹은 인접해서 전송된다. 각 그룹은 포인터(pointer)하는 데이터의 전송빈도의 순위로 정렬되어 있기 때문이다. 즉, GR_1 과 GR_2 , GR_3 와 GR_4 , 마지막으로 GR_5 , GR_6 , GR_7 , GR_8 은 하나의 그룹과 같이 묶여서 전송된다. index list에 따라서 index block을 구성할 때, index list의 구성원소가 많을수록 전체 index block의 수를 줄이면서 구성할 수 있다. 위의 표 8에서 구성된 GR의 index list에 따라서 index block을 구성하면 다음과 같다. 다음 표 7은 2개의 인덱스 채널

표 7 인덱스 리스트를 전송하기 위한 index block의 구성 예제

Data Set	index list	index block
GR ₁ = {E}	<1, 2>	<1, 2>
GR ₂ = {P}	<3>	<3, Null>
GR ₃ = {A, D}	<4, 6>	<4, Null>, <6, Null>
GR ₄ = {N}	<5, 8>	<5, Null>, <8, Null>
GR ₅ = {O, B, C}	<9>	<9, Null>
GR ₆ = {H, M}	<7, 10, 11>	<7, Null>, <10, 11>
GR ₇ = {F, G, L}	<12, 13>	<12, 13>
GR ₈ = {K, I, J}	<14, 15>	<14, Null>, <15, Null>

을 가정할 때, 위의 결과에 대한 index block의 구성 예제를 보여준다.

index block은 인덱스 채널에서 인덱스를 전송하기 위한 인덱스 아이템의 최소 전송단위이다. 즉, 각 그룹의 index list들은 index block으로 재구성해야 한다. 이때 index block은 제공된 인덱스 채널의 개수에 따라서 구성된다. 한 개의 인덱스 채널을 사용하는 경우는 한 개의 index block안에 index list의 원소 중 한 개의 인덱스 아이템만을 포함시킨다.

2개 이상의 채널에서 인덱스 정보를 전송하기 위한 index block의 구성은 집합의 index list 원소를 Alphabetic huffman tree의 구조에 따라서 묶는 것으로 일어난다. index list에 더 많은 원소를 포함시키는 것이 제공된 인덱스 채널을 효율적으로 사용하는 index block의 구성을 가능하게 한다. 즉, index block 안에 더 많은 index list의 원소를 포함시키며 전체 index block의 개수를 줄이는 구성을 가능하게 한다는 뜻이다. 같은 반복주기를 가지는 GR을 묶어 index list를 구성하면 그림 11과 같다.

index 1 group IS ₁ = (GR ₁ , GR ₂)	<1, 2, 3>
index 2 group IS ₂ = (GR ₃ , GR ₄)	<4, 5, 6, 8>
index 3 group IS ₃ = (GR ₅ , GR ₆ , GR ₇ , GR ₈)	<7, 9, 10, 11, 12, 13, 14, 15>

그림 11 같은 반복주기의 GR을 집합으로 묶은 후 index list 구성 예제

그림 11의 같은 반복주기의 GR을 묶기 이전에는 모두 12개의 index block이 구성되었다. 그러나 같은 반복주기를 같은 GR을 집합으로 묶어 index list를 구성한 후, index block을 구성하여서는 모두 9개의 index block이 구성되었다. 이를 나타내는 것이 그림 12이다.

그림 12와 같은 index block의 구성은 인덱스 할당의 빈 공간을 줄이고 index block의 개수를 줄임으로써 인덱스 접근시간을 단축하게 된다. 그림 13은 그림 12에서 구한 인덱스 블록을 두개의 인덱스 채널에 할당하는 것을 보여준다. 인덱스 채널과 데이터 채널의 분배에 대해서는 SIRAH[9]에서 제시된 방법에 의하여 최적의 인덱스 채널의 개수를 찾게 된다.

5. 성능 평가

이 장에서는 기존의 할당 알고리즘들과 ZGMD 알고리즘의 성능을 실험을 통하여 비교할 것이다. 성능 비교에 대해서는 FLAT, VF^k, Y_greedy 세 개의 알고리즘을 비교할 것이다. 여기에 ZGMD 알고리즘과 DP방법으로 MCAZD를 최소화 하는(DP + MD로 표기) 알고리즘의 실험을 추가하여 성능을 평가한다. ZGMD 알고리즘은 인덱스를 추가할 때, 다른 기존의 알고리즘들에 비해 접근시간에 대한 최상의 성능을 보장받는다. 따라서 인덱스 기술을 추가하는 부분에 대한 성능비교와 함께

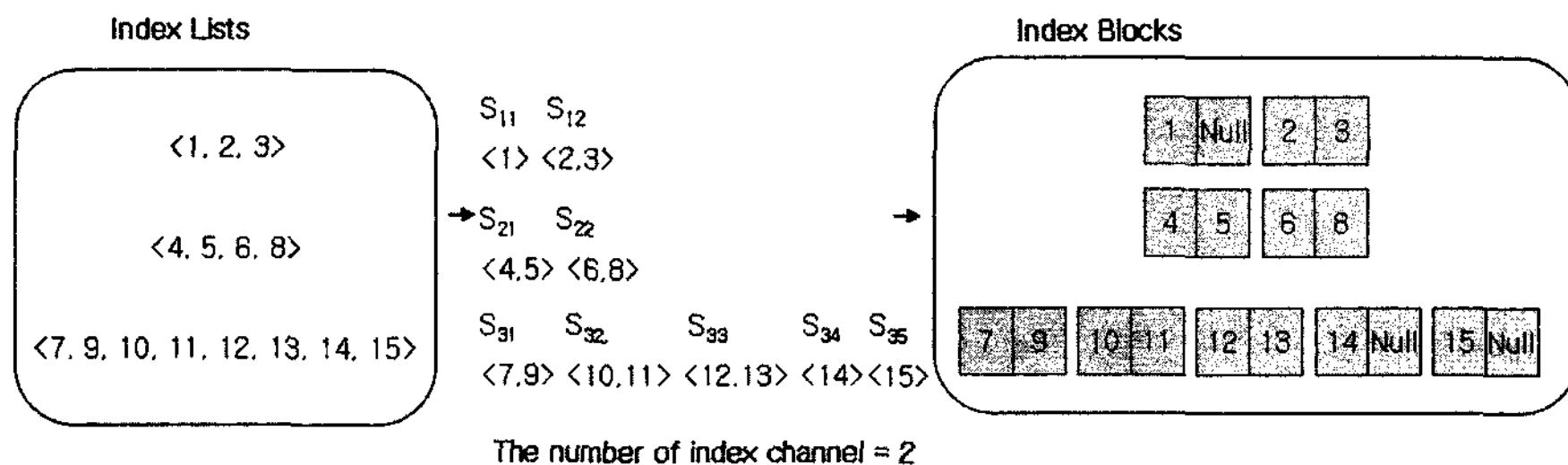


그림 12 두 개의 index channel 제공시 인덱스 블록 구성 결과

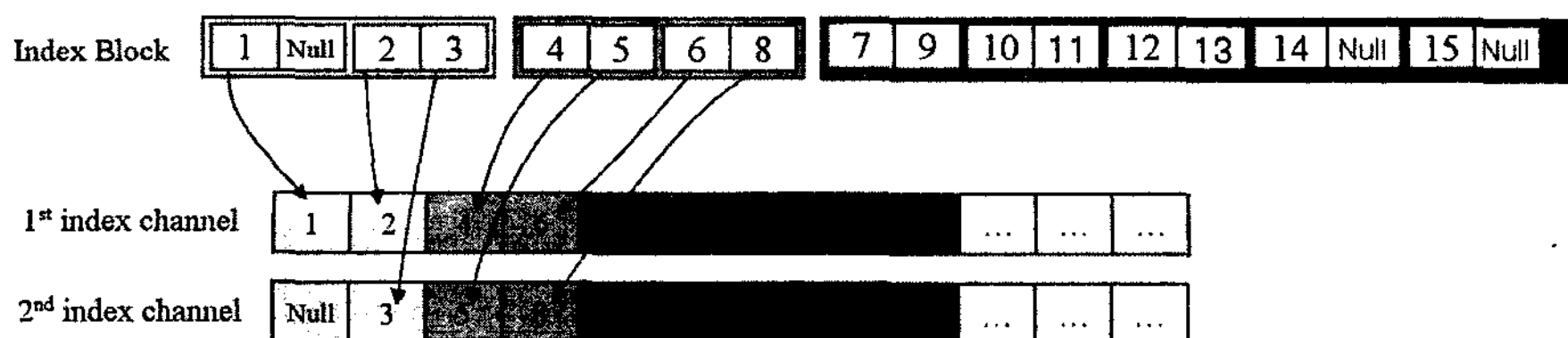


그림 13 index schedule이 생성되는 과정 (인덱스 채널의 수가 2일 경우)

인덱스 환경에 대한 추가 부담에 대한 문제도 함께 고려할 것이다. 비교하는 인덱스 알고리즘은 기존의 동등채널에 대한 브로드캐스트 디스크 기반의 데이터 구성을 고려하지 않는 데이터 할당 환경에서 Alphabetic Huffman Tree를 이용하여 구성된 인덱스를 데이터가 전송되는 것과 같은 반복주기를 고려한 SIRAH의 알고리즘이다.

실험 환경은 실제 모바일 무선 환경과 근접하도록 하여 현실적인 환경에서 제안하는 알고리즘의 우수성을 평가하도록 할 것이다. 제안하는 알고리즘은 SIRAH의 인덱스 할당 알고리즘을 기반으로 한다. 제안하는 알고리즘은 제공된 채널의 수가 작을수록 우수한 성능을 발휘한다. 서버측의 데이터베이스 크기는 기존의 알고리즘과의 공정한 비교를 위하여 최대 5000개의 데이터 유닛을 전송할 수 있다고 가정한다. [1]의 단일채널에서 브로드캐스트 디스크를 실험한 환경은 1000개 데이터 유닛으로 데이터베이스의 최대 크기를 가정하였고, 다중채널에서의 [4-6,8,9]은 최대 5000개의 데이터 유닛을 데이터베이스의 최대 크기로 고려하여 실험을 진행하였다. 제안하는 데이터 할당방법은 채널의 개수가 작을수록 우수한 성능을 보인다. 전송할 데이터 유닛의 최대 개수와 이전 연구들에서 사용한 채널 환경을 고려하여 다중채널의 최대 개수는 10개라 가정한다. [3-6,8]의 기존 데이터 할당 알고리즘들은 지프팩터 0.95를 데이터 접근빈도의 분포가 편향된 실제적 환경이라 가정하였으며, 역시 기존의 데이터 할당 알고리즘들과의 공정한 비교를 위하여 본 실험에서도 지프팩터 0.95를 기본 값으로 사용하도록 하겠다.

독립된 인덱스 채널의 사용으로 인덱스 접근 시간과 데이터 접근 시간의 통합된 비교를 위해 인덱스 unit은 데이터 unit의 1/10 크기라고 가정하였다[8]. 클라이언트는 동시에 여러 채널에 접근할 수 없고 하나의 채널에만 수신 가능하다고 가정한다. 시뮬레이션을 위한 매개 변수는 표 8과 같다. 성능 분석을 수행하는 환경은 3.2 인텔 펜티엄 4 프로세서 3.2GHz의 CPU에 1024MB의 RAM, Microsoft Windows XP Professional의 OS가 설치된 desk-top에서 시뮬레이션 되었다.

제안하는 알고리즘의 성능 평가를 위해 먼저 알고리즘의 수행시간을 비교한다. 계속해서 인덱스가 없는 환경에서의 제안하는 알고리즘의 접근 시간 성능을 평가한다. 환경 변화에 따른 제안하는 알고리즘의 성능 평가를 위해 채널의 수와 데이터 수, 데이터 접근빈도의 분포를 변화시키며 접근 시간에 대한 성능을 평가할 것이다. 또한, 인덱스 기술이 추가된 환경에서의 변화를 측정하기 위해, 제안하는 인덱스 기술과 SIRAH 인덱스 알고리즘의 인덱스 접근 시간에 대한 성능을 비교하며, 인덱스 기술이 추가된 데이터 할당 알고리즘의 접근 시간에 대

표 8 시뮬레이션 매개 변수

시뮬레이션 매개 변수	범 위
전체	
채널의 수	1~10
Zipf 매개 변수 θ	0.0~1.0
데이터베이스 내의 데이터의 수	0~5000
데이터 오브젝트의 크기	1 unit
인덱스 오브젝트의 크기	1/10 unit
기본 환경	
데이터 채널의 수	8
인덱스 채널의 수	2
Zipf 매개 변수 θ	0.95
데이터베이스 내의 데이터의 수	2000

한 성능을 인덱스가 없는 환경에서와 같은 지표를 사용하여 측정할 것이다. 데이터와 인덱스에 대한 접근 시간의 측정 단위는 데이터 1unit을 기준으로 하였다.

5.1 알고리즘 수행 시간에 대한 성능 평가

접근 시간의 측정 이전에 알고리즘 수행 시간의 평가 또한, 다중채널에서의 데이터 할당 알고리즘의 성능 측정에 중요한 부분이 된다. 기존의 MCAED의 최소화를 위한 DP 알고리즘은 접근 시간의 우수한 성능에 비해 알고리즘 수행 시간에 대한 부담이 크게 작용하였기 때문이다. 제안하는 알고리즘 수행 시간에 대한 성능 평가는 표 8의 기본 환경 변수에 따라 데이터 할당 알고리즘에 대한 수행시간을 평가하였다. 먼저 1개의 인덱스 채널과 4개의 데이터 채널을 사용하는 환경에 대하여 수행시간을 분석한 그림이 그림 14이며, 채널 증가에 따른 수행시간 성능의 비교를 위해 2개의 인덱스 채널과 8개의 데이터 채널을 사용하는 시뮬레이션 분석이 그림 15이다.

FLAT 알고리즘은 채널의 개수에 따라 데이터를 균등 분할하는 단순한 알고리즘이다. 데이터의 증가에 따른 수행시간의 변화가 없으며 따라서, 좌표의 x축과 일치하는 가장 월등한 수행시간의 성능을 보인다. Y_greedy 알고리즘은 접근 시간의 성능 대비 수행시간의

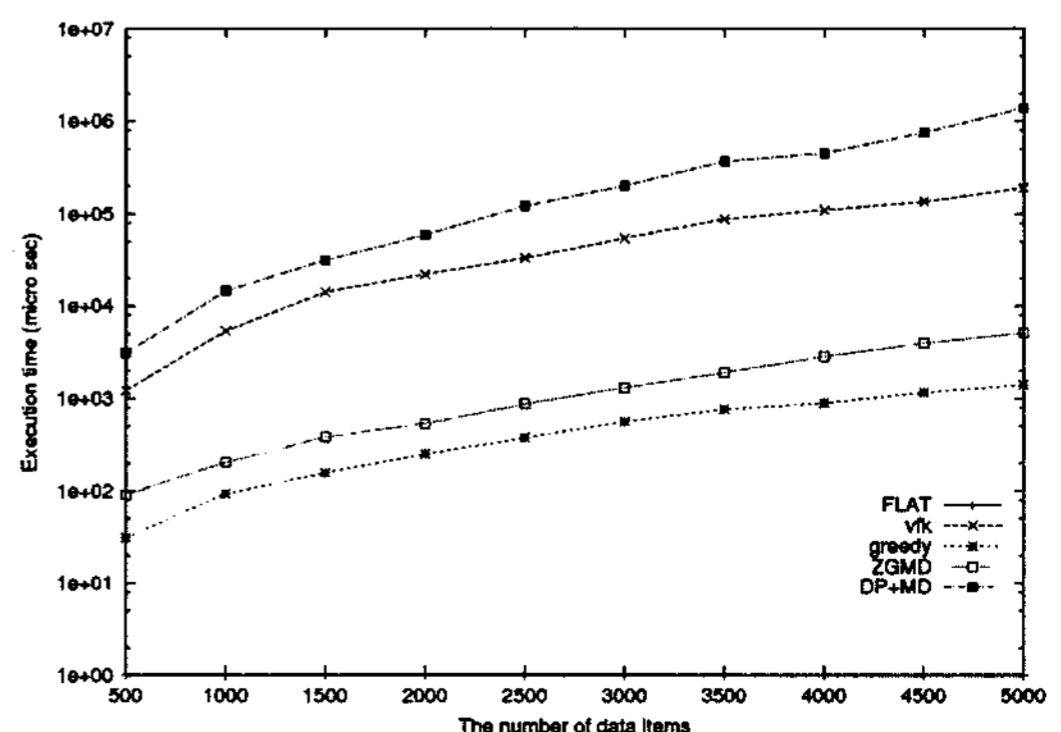


그림 14 데이터 변화에 따른 수행시간-1

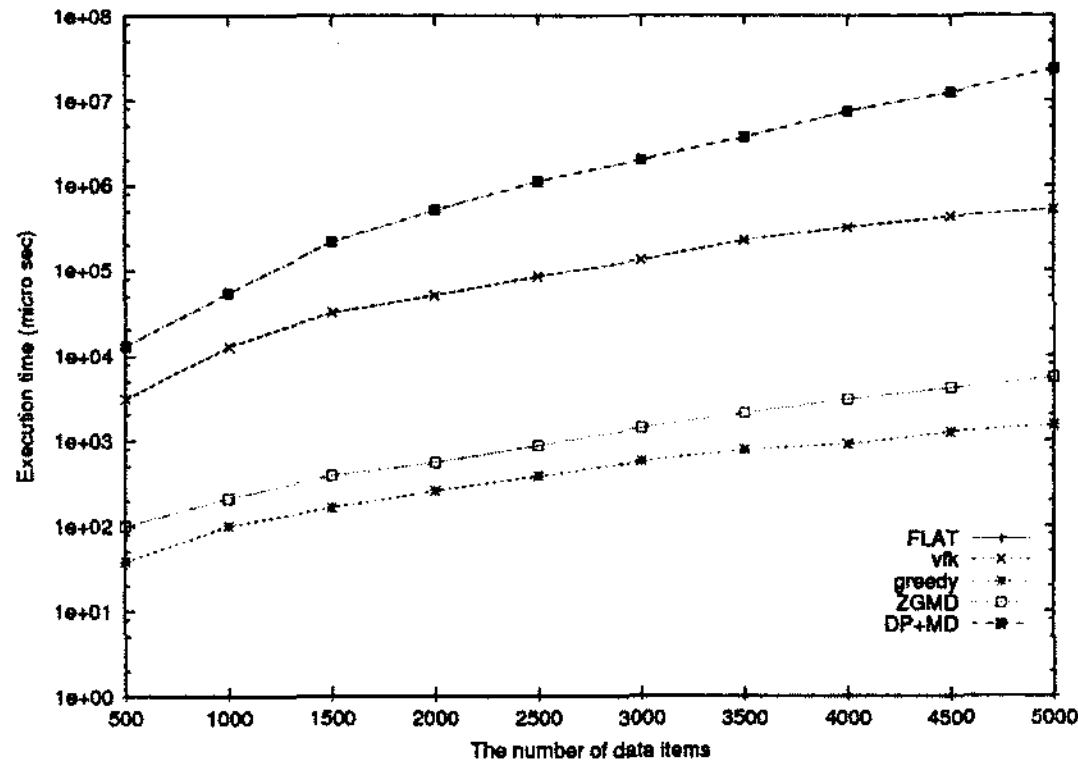


그림 15 데이터 변화에 따른 수행시간-2

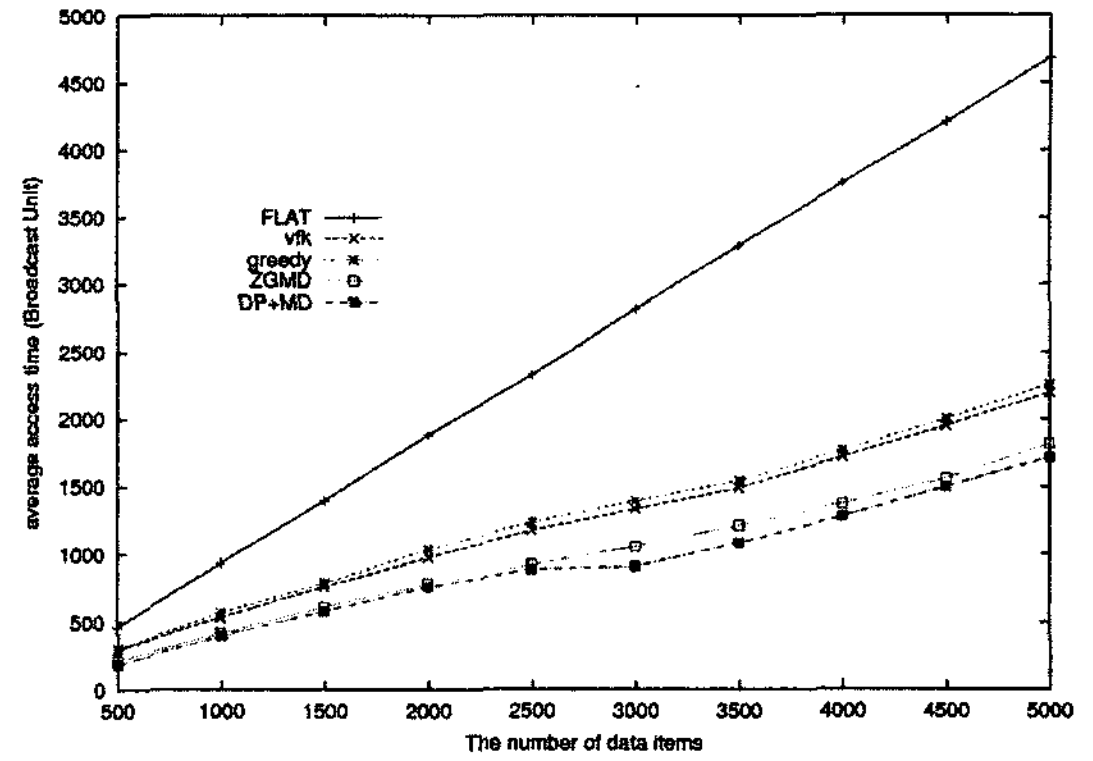


그림 16 데이터 변화에 따른 접근시간

성능이 우수하다고 평가 받는 알고리즘이다. 제안하는 방법은 채널에의 데이터 할당에 있어서 수행시간이 빠르나 채널에 할당된 데이터에 대한 반복주기 t_r 를 계산하는 추가 지연으로 Y_greedy 알고리즘에 비해 수행시간이 느려졌다. 그러나 그 차이가 크지 않고, VF^K 의 알고리즘에 비교한다면 월등히 우수함을 그림 14에서 확인할 수 있다. VF^K 알고리즘은 접근 시간 성능의 장점에도 불구하고 수행시간에 있어서 큰 부담을 가지고 있는 알고리즘이다. DP방법으로 MCAZD를 최소화 하는 할당은 수행시간 $O(NK^K)$ 로 이러한 VF^K 알고리즘보다 월등히 높은 수행 시간을 보인다. 또한 채널의 수에 지수배의 수행 시간의 증가를 보이기 때문에 그림 15에서 따라 다수의 채널이 제공된 환경에서는 수행 시간의 성능이 더욱 악화됨을 알 수 있다.

5.2 인덱스 환경에 대한 데이터 할당 알고리즘의 성능 평가

제안하는 데이터 할당 알고리즘의 성능 보장을 위해서는 인덱스 기술이 필요함을 확인하였다. 따라서 다음의 실험은 인덱스 기술을 적용한 상황에서의 접근 시간의 성능 변화를 측정할 것이다.

먼저 인덱스 접근시간을 포함한 전체 데이터의 접근 시간을 측정하도록 하겠다. 인덱스 유닛의 크기는 데이터 유닛 크기의 1/10로 가정하여 전체 접근 시간을 측정하였다. 그림 14와 15의 시뮬레이션 환경은 2개의 인덱스 채널, 8개의 데이터 채널을 사용하였다. 가정하는 실험환경의 데이터 수의 변화 범위는 500에서 5000까지이다. 데이터 수의 변화 범위를 고려할 때, 기본 채널 수 10개는 데이터 할당 알고리즘의 접근시간에 대한 성능 변화 정도를 가장 잘 측정할 수 있는 수치가 된다. 10개 이상의 기본 채널 수에서는 데이터 할당 알고리즘들의 성능 차이가 줄어들기 시작한다. 또한, 10개 이하의 채널 수 변화에 따른 성능 차이는 채널 수 변화에 따른 성능 분석으로 새로이 측정하기로 한다.

그림 16의 실험은 데이터 수 변화에 따른 제안하는 알고리즘과 비교 알고리즘의 접근시간 성능 변화를 측정하기 위한 것이다. 제공된 채널수가 고정되어 있을 때, 데이터 수가 증가에 따른 접근 시간의 증가 비율이 낮을수록 접근시간 성능에 대해서 효과적인 데이터 할당 알고리즘이 되기 때문이다.

인덱스가 없는 환경에서 FLAT보다 높은 접근 시간을 보였던 ZGMD 알고리즘과 MCAZD의 DP 할당 알고리즘은 그림 15에서와 같이 가장 빠른 성능을 보이고 있다. 인덱스를 이용하여 데이터의 위치를 검색하고, 여러 채널을 탐색하는 지연이 없어졌기 때문에 인덱스 환경에서 제안하는 알고리즘은 우수한 성능을 보이게 된다.

그림 17과 그림 18은 인덱스 환경에서 채널수의 변화에 따른 접근 시간을 측정한 결과이다. 인덱스 채널의 수는 2개로 고정하였으며 데이터 채널의 수를 1개에서 10개까지 변화시키며 접근 시간의 변화를 측정하였다. 10개 이상의 채널 증가에 대해서는 데이터 할당 알고리즘들의 성능 차이가 줄어들면서 접근시간 0의 값으로 근사되기 시작한다. 따라서 10개 이상 채널 증가에 대한 실험은 제외하기로 한다.

데이터변화와 마찬가지로 제안하는 알고리즘은 채널이 증가함에 따라 우수한 접근 시간을 보이고 있다. 가장 빠른 접근 시간을 보이는 것은 DP방법으로 MCAZD를 최소화 하는 할당으로 제안하는 알고리즘과 비교하여 그 차이가 미비하다. 그러나 기존의 Y_greedy 방법과 VF^K 와 비교하여 월등한 성능을 보이고 있음을 알 수 있다.

마지막으로 그림 19는 zipf 변화에 따른 접근시간의 변화를 측정하였다. 역시 기본값의 상황인 2000개의 데이터에 2개의 인덱스 채널과 8개의 데이터 채널을 사용하는 환경을 시뮬레이션 한다. 이것은 데이터 채널을 8개 이상을 사용하는 환경에서 각 할당알고리즘의 성능 차이가 분명하게 드러났기 때문이다. 10개 이상의 채널 환경에서는 각 할당알고리즘의 성능 차이가 점차로 줄

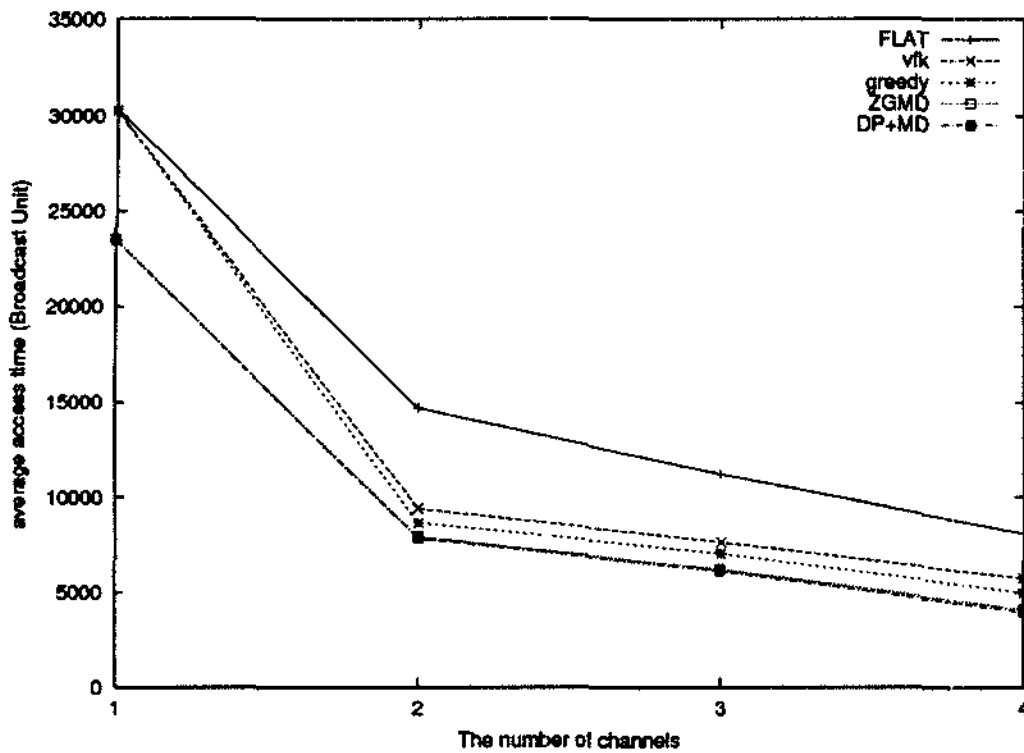


그림 17 채널 수 변화에 따른 접근시간-1

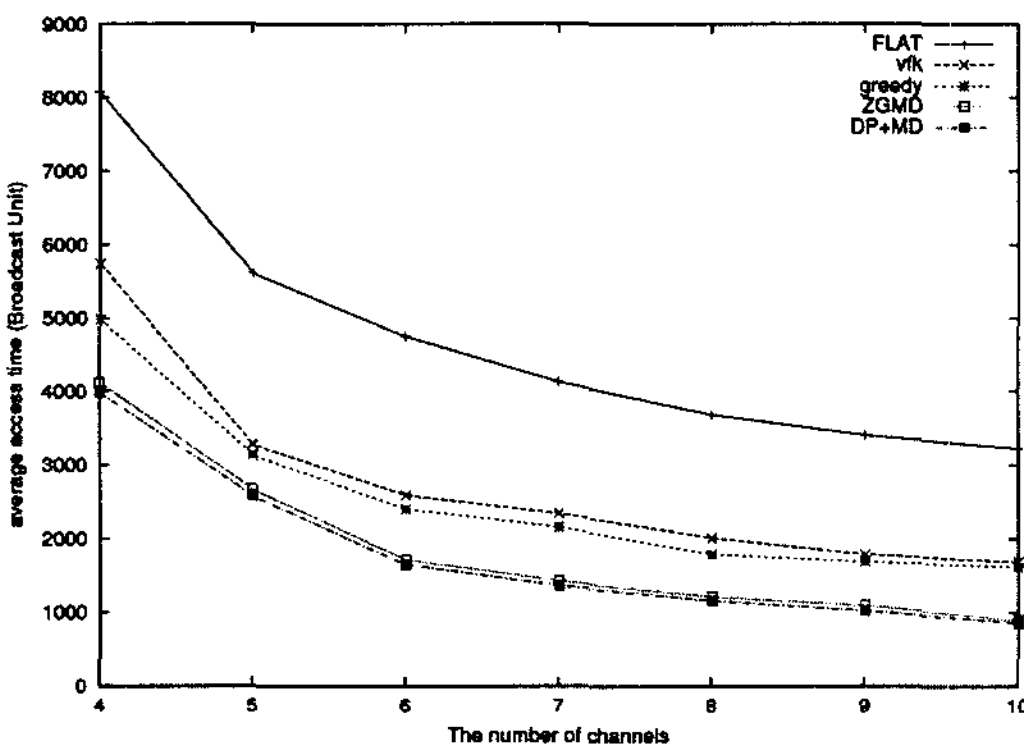


그림 18 채널 수 변화에 따른 접근시간-2

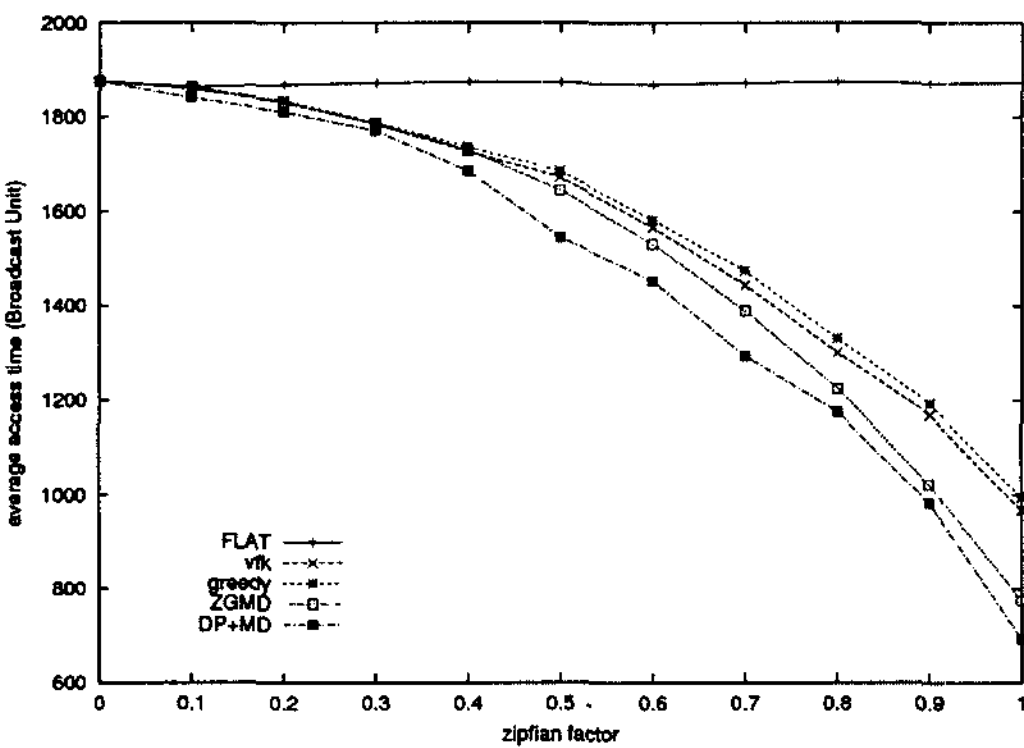


그림 19 zipf 변화에 따른 접근시간

어떻게 된다.

인덱스가 제공된 환경에서는 인덱스 없는 환경에서와는 반대로 데이터의 접근 빈도가 상위 소수의 데이터에 집중될수록 제안하는 알고리즘과 DP에 의한 할당 방법의 접근시간이 빠르게 단축되는 것을 확인할 수 있다. 결국 제안하는 데이터 할당 방법은 인덱스 기술이 적용될 경우 zipf 분포가 $\theta=1$ 에 가까워지는 현실적인 환경

에서 더욱 우수한 성능을 발휘할 것을 예상할 수 있다.

6. 결론

무선 환경에서 접근 시간의 성능은 무선기기를 이용하는 클라이언트의 서비스에 대한 만족도로 이어지기 때문에 싱글 채널에서부터 다중채널에서 데이터를 전송하는 방법까지 접근시간과 튜닝시간을 단축하기 위한 연구가 계속되어왔다. 본 논문은 다수의 클라이언트의 접근시간에 대한 서비스 만족을 위한 다중채널에서의 데이터 할당 방법을 제시하였다.

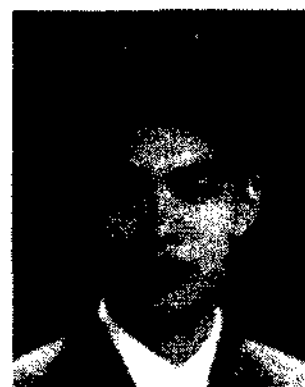
본 논문에서는 브로드캐스트 디스크 구성을 통해 접근 시간을 보다 효과적으로 단축시키기 위해 채널에 할당되는 데이터들의 접근 빈도의 분포를 수치화 시키도록 하였다. 접근 빈도의 분포를 수치화하여 이 수치를 최대화시키는 할당을 한 후 브로드캐스트 디스크 구성으로 인한 접근 시간의 단축을 하고자 하였다. 본 논문에서는 접근 빈도 분포의 편향성을 최대화 시키는 DP 기반의 방법을 제시하였으나 이러한 DP 방법의 알고리즘 수행 계산 시간은 현실적으로 부담이 되는 현저히 큰 수치를 보였다. 따라서 Y_greedy 알고리즘을 기반으로 접근 빈도 분포의 편향성을 최대치에 근사시키는 알고리즘을 제시하였다. 제시하는 알고리즘의 접근 시간 성능은 기존의 알고리즘과 비교하여 Y_greedy 알고리즘에 가까운 빠른 성능을 보이며, 접근 시간의 성능에 대해서 우수한 성능을 보이는 것을 실험을 통해 확인하였다.

최근에는 B.Zheng에 의해 TOSA라는 접근시간을 줄이는 다중채널에서의 브로드캐스트 스케줄링 알고리즘이 제시되었다[7]. 기존에는 데이터 아이템의 크기와 채널의 대역폭이 일정하다고 가정하고 기존의 연구를 진행하였다. 하지만 B.Zheng은 보다 현실적인 환경을 고려하여 데이터 아이템의 크기와 채널의 대역폭이 각각의 환경에 의해 다를 수 있다고 가정하고 Y_Greedy를 개선하는 새로운 알고리즘을 제시한 것이다. 인덱스의 사용과 더불어 각각 상이한 데이터 아이템의 크기와 다중 채널의 서로 다른 대역폭을 고려하는 것은 데이터 스케줄링에 대한 연구에서 보다 현실적인 연구가 될 수 있다. 따라서 본 연구에서 이러한 실질적인 환경을 고려하도록 하여 다중채널에서의 데이터 할당에 대한 연구를 개선하는 것이 향후 연구과제로 남는다.

참고 문헌

[1] S. Acharya, R. Alonso, M. Franklin, and S. Zdonik, "Broadcast Disks: Data Management for Asymmetric Communication Environments," Proceedings of ACM SIGMOD Conference, pp. 199-210, 1995.
 [2] T. Imielinski and B. R.Badrinath, "Mobile Wireless Computing: Challenges in Data Management,"

- Communications of the ACM, Vol.37, No.10, pp. 19-28, Oct. 1994.
- [3] S. Ganguly and R.Alonso, "Query Optimization in Mobile Environments," Technical Report No. LCSR-TR-219, Department of Computer Science, Rutgers University, 1993.
- [4] W. Yee, S. Navathe, E. Omiecinski, and C. Jermaine, "Efficient Data Allocation over Multiple Channels at Broadcast Servers," IEEE Transactions on Computers, Vol.51, No.10, pp. 1231-1236, Oct. 2002.
- [5] K. Prabhakara, K.A. Hua, and J. Oh, "Multi-Level Multi-Channel Air Cache Designs for Broadcasting in a Mobile Environment," Proceedings of the 16th International Conference on Data Engineering, pp. 167, 2000
- [6] W.C. Peng and M.S. Chen, "Dynamic Generation of Data Broadcasting Programs for Broadcast Disk Arrays in a Mobile Computing Environment," Proceedings of the Ninth International Conference on Information and Knowledge Management, pp. 35-45, Nov. 2000.
- [7] Baihua Zheng, Xia , Xing Jin, and Dik Lum Lee, "TOSA: A Near-Optimal Scheduling Algorithm for Multi-Channel Data Broadcast," The 6th International Conference on Mobile Data Management (MDM'05), Ayia Napa, Cyprus, May 9-13, 2005.
- [8] S. Jung and S. Nam, "An Adaptive Data Allocation Method Based on Data Access Probability over Multiple Broadcasting Channels," Korea Information Science Society Database, Vol.33, No. 1, pp. 1229-7799, Feb. 2006.
- [9] S. Jung and B. Lee, S. Pramanik, "A Tree-Structured Index Allocation Method with Replication over Multiple Broadcast Channels in Wireless Environments," IEEE Transactions on Knowledge and Data Engineering, Vol.17, No.3, pp. 311-325, Mar. 2005.
- [10] C. Hsu, G. Lee, and A. Chen, "Index and Data Allocation on Multiple Broadcast Channels Considering Data Access Frequencies," Proceedings of the Third International Conference on Mobile Data Management, pp. 87-93, 2002.
- [11] Narayanan Shivakumar, Suresh Venkatasubramanian, "Energy Efficient indexing for Information Dissemination In Wireless Systems," ACM-Baltzer Journal of Mobile Networks and Nomadic Applications, Dec. 1996.
- [12] G. Forman and J. Zahorjan, "The Challenges of Mobile Computing," IEEE Computer, Vol.27, No.4, pp. 38-47, Apr. 1994.
- [13] S. Lo and A. Chen, "Optimal Index and Data Allocation in Multiple Broadcast Channels," Proceedings of the 16th International Conference on Data Engineering, pp. 293, 2000.
- [14] Dimitrios Katsaros, Yannis Manolopoulos, "Broadcast program generation for Webcasting," Data and Knowledge Engineering, Vol.49, No.1, pp. 1-21, Apr. 2003.
- [15] W.G. Yee, E. Omiecinski, and S.B. Navathe, "Efficient Data Allocation for Broadcast Disk Arrays," Technical Reprot GIT-CC-02-20, Georgia Institute of Technology, 2001.
- [16] K. Tan and J. Yu, "An Analysis of Selective Tuning Schemes for Nonuniform Broadcast," Data and Knowledge Engineering, Vol.22, No.3, pp. 319-344, 1997.
- [17] Mesquite Software Inc., "CSIM18 Simulation Engine USER'S GUIDE," 2000.
- [18] H. Leong and A. Si, "Data Broadcasting Strategies Over Multiple Unreliable Wireless Channels," Proceedings of the Fourth International Conference on Information and Knowledge Management, pp. 96-104, Dec. 1995.
- [19] A. Hurson, Y. Chehadeh, and J. Hannan, "Object Organization on Parallel Broadcast Channels in a Global Information Sharing Environment," Performance Computing and Communications Conference, 2000, Conference Proceeding of the IEEE International, pp. 347-353, Feb. 2000.
- [20] J. Gray, P. Sundaresan, and S. Englert, K. Baclawski, and P. Weinberger, "Quickly Generating Billion-record Synthetic Databases," Proceedings of ACM SIGMOD International Conference, Management of Data, May, pp. 243-252, 1994.
- [21] J. Juran, A.R. Hurson, N. Vijaykrishman, and S. Boonsiriwattanakul, "Data Organization and Retrieval on Parallel Air Channels, Performance and Energy Issues," Proceedings of International Conference High Performance Computing, Wireless Networks, Vol.10, No.1, pp. 501-510, Mar. 2004.



이 원 택

1997년 3월~2004년 2월 서강대학교 컴퓨터학과 학사. 2005년 3월~2007년 2월 서강대학교 컴퓨터학과 석사. 2007년 3월~현재 현대오트모터 시스템즈. 관심분야는 이동통신, 모바일 컴퓨팅 시스템, 모바일 데이터베이스, 모바일 트랜잭션



정 성 원

1988년 서강대학교 전자계산학 학사. 1990년 M.S. in Computer Science at Michigan State Univ. 1995년 Ph.D. in Computer Science at Michigan State Univ. 1997년~2000년 한국전산원 선임 연구원. 2000년~현재 서강대학교 컴퓨터

학과 부교수. 관심분야는 Mobile Databases, Spatial Databases, Telematics, GIS, LBS, Mobile Agents