이동 환경에서 다중점 질의를 위한 효율적인 방송 데이타 클러스터링

(Efficient Broadcast Data Clustering for Multipoint Queries in Mobile Environments)

방 수 호 † 정 연 돈 †† 김 명 호 †††

(Su Ho Bang) (Yon Dohn Chung) (Myoung Ho Kim)

요 약 최근 뛰어난 성능의 휴대용 컴퓨터들이 등장하고 무선 통신 기술이 발달함에 따라 이동 컴퓨팅에 대한 관심이 급격히 증가하고 있다. 이동 컴퓨팅에서는 무선 통신이 가지는 통신 대역과 에너지의 제약 때문에 방송 기법을 많이 사용한다. 본 논문은 다수의 데이타를 참조하는 다중 점 질의에 대한 방송 데이타 클러스터링 기법에 대해 기술한다. 방송 데이타의 클러스터링을 통해 사용자는 보다 빨리 필요로 하는 데이타를 얻을 수 있다. 본 논문에서는 먼저 데이타 친화도와 세그먼트 친화도라는 기준을 제시하며 이에 기반한 클러스터링 방법을 제안한다. 데이타 친화도는 두 데이타 개체가 질의에서 같이 참조되는 정도를 나타내며, 세그먼트 친화도는 두 데이타 집합(세그먼트)이 질의에서 같이 참조되는 정도를 나타낸다. 제안하는 방법은 질의의 수가 증가에도 성능이 크게 저하되지 않는 특징을 지닌다.

Abstract Mobile computing has become a reality thanks to the convergence of two technologies : powerful portable computers and the wireless networks. The restrictions of wireless network, such as bandwidth and energy limitations make data broadcasting an attractive data communication method. This paper addresses the clustering of wireless broadcast data for multipoint queries. By effective clustering of broadcast data, the mobile client can access the data on the air in short latency. In the paper, we define the data affinity and segment affinity measures. The data affinity is the degree that two data objects are accessed by queries, and the segment affinity is the degree that two sets of data (i.e., segments) are accessed by queries. Our method clusters data objects based on data and segment affinity measures. We show that the performance of our method is scarcely infuenced by the growth of the number of queries.

1. 서 론

최근 휴대 전화, PDA, 노트북과 같은 휴대용 컴퓨터의 성능이 향상되고, 무선 통신 기술이 발달함에 따라 이동 컴퓨팅에 대한 관심이 급격히 증가하고 있다.[1, 2] 이동 컴퓨팅(mobile computing)이란 사용자가 자유로이 이동하면서 무선 통신을 통해 다양한 정보 서비스를 제공받는 환경을 말한다.

일반적으로 이동 컴퓨팅 환경에서 가장 큰 제약점으로 인식되는 점들은 에너지 사용의 제한, 통신 대역(bandwidth)의 협소, 무선 통신의 안전성 결여, 단말기의 화면 크기 등이다. 이 중에서 에너지 사용의 제한과 통신 대역의 협소함을 극복하기 위해 방송(broadcasting) 이라는 통신 방법을 많이 사용한다.[3] 방송이란 각 사용자가 서버에게 요구(request)를 하지 않고 일방적으로 서버가 데이타를 전달하는 통신 방법을 말한다. 방송 기법에서는 일정량의 채널을 다수의 사용자가 공유하여 사용하기 때문에 주파수의 효율성 측면에서 우수한 특징을 가진다. 그리고, 무선 통신의 특성상 전파를 송신하는데 필요한 에너지는 수신의 그것에 비하여 아주 크기 때문에 방송 기법을 사용할 경우 에너지

· 본 연구는 HVcenter의 지원으로 수행되었음.
† 비 회 원 : 한국무역정보통신 연구원
shbang@dbserver.kaist.ac.kr
†† 비 회 원 : 한국과학기술원 전산학전공 연구 교수
ydchung@dbserver.kaist.ac.kr
††† 종신회원 : 한국과학기술원 전산학전공 교수
mhkim@dbserver.kaist.ac.kr
논문접수 : 2000년 12월 26일
심사완료 : 2001년 10월 23일

의 사용도 줄일 수 있다.

방송 기법을 통한 데이타 전송 시 고려하여야 하는 성능 요소들에는 데이타의 접근 시간(access time)과 튜닝 시간(tuning time)이 있다. 접근 시간은 사용자가 방송 채널을 통해 특정 데이타를 수신하기를 희망한 시점부터 원하는 데이타를 전부 받을 때까지의 시간을 나타낸다. 튜닝 시간은 방송 데이타를 수신하기 위해 에너지를 사용해야 하는 시간을 나타낸다.

방송 기법의 사용을 위한 기존의 연구로는 접근 시간을 줄이기 위한 방송 데이타 스케줄링 기법[4]과 튜닝 시간을 줄이기 위한 색인 기법[5, 6], 그리고 접근 시간과 튜닝 시간을 모두 줄일 수 있는 캐싱 기법[7, 8] 등이 있었다.

본 논문에서는 무선 데이타 방송 환경에서 다중점 질의를 위한 방송 데이타 클러스터링에 대해 살펴본다. 다중점 질의란 하나의 사용자 질의가 둘 이상의 데이타를 검색하는 질의로서, 같은 질의에 나오는 데이타들을 클러스터링 함으로써 접근 시간을 줄일 수 있다. 관련 연구로는 [9, 10]이 있으며, 기본적으로 접근 빈도가 높은 질의부터 우선적으로 클러스터링 하는 기법을 사용한다. 따라서 질의들의 접근 빈도의 차이가 크고, 질의의 수가 적을 때 더 좋은 성능을 나타낸다. 그러나, 방송 기법이 사용되는 환경과 다중 점 질의의 특성을 고려하면 위와 같은 조건이 충족되기는 어렵다. 방송 기법은 데이타의 수에 비해 사용자의 수가 많은 경우에 주로 사용되며, 따라서 가능한 다중 점 질의의 수가 매우 커질 수 있기 때문이다.

본 논문의 구성은 다음과 같다. 먼저, 2장에서는 본 논문에서 다루는 방송 데이타 클러스터링에 대해 설명하고, 기존의 연구들에 대해 알아본다. 3장에서는 기존 연구의 문제점을 지적하고, 새로운 방송 데이타 클러스터링 기법을 제안하며 4장에서는 실험을 통해 제안된 방법의 성능을 평가한다. 마지막으로 5장에서는 결론과 향후 연구 방향을 기술한다.

2. 배경 및 관련 연구

2.1 방송 데이타의 클러스터링

본 논문에서 가정하는 무선 데이타 방송 환경은 하나의 방송 서버가 전달하는 데이타를 다수의 사용자들이 수신하는 것이다. 서버는 같은 데이타 집합을 주기적으로 방송하여 사용자들이 원하는 데이타를 수신할 수 있도록 한다. 그림 1은 이동 사용자가 방송 스트림으로부터 자신이 원하는 데이타에 접근하는 모습을 나타내고 있다.

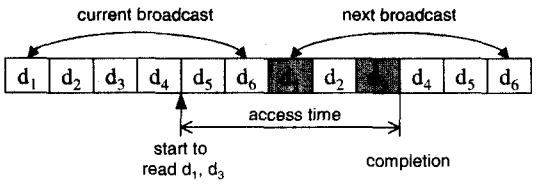

그림 1 무선 데이타 방송에서 데이타 수신 과정

방송 데이타 클러스터링 기법에서 성능 평가의 기준으로 주로 사용되는 것은 접근 시간(access time)이다. 접근 시간은 질의가 시작된 시점부터 질의가 참조하는 데이타를 전부 받을 때까지의 시간이다. 접근 시간은 질의가 시작된 시점에 따라 변화하기 때문에, 일반적으로 평균 접근 시간을 사용한다.

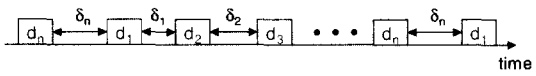

그림 2 방송 스트림에서 질의의 데이타 집합의 배치

기존 연구 [9]에서, 다중점 질의에 대한 평균 접근 시간을 다음과 같이 계산하였다. 주어진 질의 q_i가 $q_i=\{d_1, d_2, ..., d_n\}$ 이고, δ_i를 스케줄 σ에서 데이타 개체 d_i와 d_{i+1}사이의 간격이라고 하면 방송 스케줄 σ에서 질의 q_i의 평균 접근 시간 $AT^{avg}(q_i, \sigma)$는 다음과 같다. (BSize는 방송 스케줄의 길이이다.)

$$AT^{avg}(q_i, \sigma) = BSize - \frac{1}{2 \cdot BSize} \sum_{j=1}^{n} (\delta_j) \quad (1)$$

무선 방송 데이타의 클러스터링 문제는 각 사용자들이 제기하는 질의들에 대하여 최소의 접근 시간을 갖도록 하는 방송 스케줄을 찾는 것으로 각 질의들의 평균 접근 시간의 합인 전체 접근 시간(TAT: Total Access Time)을 최소화하는 것을 목적으로 한다. 최적의 스케줄을 찾는 것은 NP-Complete임이 증명되었으며, 따라서 최적에 가까운 스케줄을 찾을 수 있는 방송 클러스터링 기법이 필요하다.

$$TAT(\sigma) = \sum_{q_i \in Q} AT^{avg}(q_i, \sigma) \times freq(q_i) \quad (2)$$

[9]에서는 다중 점 질의에 대한 새로운 측정 기준으로 질의 거리(QD: Query Distance)를 제안하였다. 질의 거리란 방송 스케줄에서 주어진 질의에 포함된 데이타 집합이 가지는 최소 길이(minimal distance)이다. 주어진 질의 q_i가 접근하는 데이타 집합이 $\{d_1, d_2, ..., d_n\}$ 이고, δ_i를 스케줄 에서 데이타 개체 d_i와 d_{i+1}사이의 간격이라고 할 때, 방송 스케줄 σ 에서 질의 q_i의 질의

거리 $QD(q_i, \sigma)$는 다음과 같이 정의된다.

$$QD(q_i, \sigma) = BSize - MAX(\delta_1, \delta_2, \ldots, \delta_n) \quad (3)$$

2.2 스케줄 확장 방법 (SEM: Schedule Expansion Method)

SEM[9]은 각 질의들을 참조 빈도수에 따라 정렬한 후 빈도 수가 높은 질의부터 각 질의가 접근하는 데이타 집합을 스케줄에 첨가하면서 스케줄을 점차 확장해 나가는 방식으로 방송 스케줄을 구성한다. SEM에서 취하는 기본 전략은 다음과 같다.

- 빈도 수가 높은 질의의 질의 거리를 우선하여 최소화 시킨다.
- 빈도 수가 높은 질의의 질의 거리를 증가시키지 않는 범위에서 빈도 수가 낮은 질의의 질의 거리를 최소화 시킨다.

즉, SEM에서는 빈도 수가 높은 질의에 포함된 데이타들의 질의 거리를 고정시킨 후, 빈도 수가 낮은 질의에 포함된 데이타들을 스케줄에 추가한다. 따라서, 질의의 수가 적고 질의들의 빈도 수가 크게 차이가 날 경우 좋은 성능을 나타낸다. 반대로 질의 수가 많아지면 성능 향상이 감소하며, 빈도 수가 낮은 질의들이 반영되기 전에 전체 스케줄이 결정되는 경우도 생길 수 있다.

2.3 그레이 코딩 방법 (GCM: Gray Coding Method)

GCM[10]은 그레이 코드(gray codes)의 클러스터링 특성을 이용한 방법이다. 그림 3에 나타나듯이 이진 코드보다 그레이 코드에서 숫자들이 모여 있으며, 하위 비트보다 상위 비트에서 더 잘 모여있음을 알수 있다. GCM은 이러한 그레이 코드의 특성을 이용한 방법으로 먼저 각 데이타를 비트-벡터(bit-vector)로 표현한 다음, 그레이 코드 순으로 정렬하면 그 결과가 방송 스케줄이 된다. 비트-벡터의 각 비트는 하나의 질의에 대응되며 상위 비트일수록 참조 빈도가 높은 질의에 대응된다. 따라서 참조 빈도가 높은 질의의 데이타 집합일수록 우선적으로 클러스터링 된다.

Gray value	Gray code	Binary code
0	000	000
1	001	001
2	011	010
3	010	011
4	110	100
5	111	101
6	101	110
7	100	111

그림 3 3비트 그레이 코드와 이진 코드

SEM과 마찬가지로, GCM도 기본적인 접근 방법은 접근 빈도가 높은 질의에 포함된 데이타 집합부터 클러스터링하는 것이다. 따라서, GCM의 성능 향상 정도도 SEM의 그것와 비슷한 양상을 보인다. 전체적인 성능은 SEM보다 조금 낮은데 이것은 방법이 단순한 데서 기인한 것으로 분석된다.

3. 제안하는 방송 데이타 클러스터링 방법

본 장에서는 본 논문의 연구 동기를 밝히고, 새로운 방송 데이타 클러스터링 기법을 제안한다. 다음은 앞으로 방송 데이타 클러스터링 방법을 설명하면서 사용할 기호들과 각각에 대한 설명이다.

- d_i : 방송을 통해 전달되는 데이타 개체를 나타낸다.
- $|d_i|$: 데이타 개체 d_i 의 길이를 나타낸다.
- D : 방송되는 데이타 집합을 의미한다. 즉, 데이타의 개수가 N이면 $D = \{d_1, d_2, \ldots, d_N\}$ 이다.
- $BSize$: 방송 데이타 스케줄의 길이를 나타낸다. $Bsize$는 다음과 같이 계산된다. $BSize = \sum_{\forall d_i \in D} |d_i|$
- q_i : 사용자가 요구하는 질의를 나타내며, 데이타들의 집합으로 표현할 수 있다. 예를 들어 데이타 개체 d_1, d_3을 요구하는 질의 q_1는 $q_1 = \{d_1, d_3\}$로 표현된다. 데이타 개체들간의 순서는 고려하지 않는다.
- $N_{data}(q_i)$: 질의 q_i 가 포함하는 데이타 개체의 개수를 의미한다. $q_i = \{d_1, d_2, d_3\}$이면 $N_{data}(q_i) = 3$ 이다.
- $freq(q_i)$: 질의 q_i 의 참조 빈도를 나타낸다.
- Q : 전체 사용자가 요구하는 질의들의 집합을 의미한다. 전체 질의의 개수가 M이면 $Q = \{q_1, q_2, \ldots, q_M\}$ 이다.
- σ : 데이타 집합 D의 방송 스케줄을 의미한다. 즉 방송 데이타의 전달 순서를 나타내며, $\sigma = <<d_i, d_j, \ldots, d_k>>$로 표기한다.

3.1 연구 동기(Motivation)

개별 통신 기법에 비해, 방송 기법이 가지는 가장 큰 장점은 사용자의 수가 많아져도 그에 따른 오버로드(overload)가 전혀 생기지 않는다는 점이다. 방송 기법에서는 제공자가 사용자의 요구를 예측하여 일방적으로 데이타를 전달하기 때문에 사용자의 수에 전혀 영향을 받지 않는다. 이런 점을 고려하면 방송 기법은 주로 사용자의 수가 많은 경우에 사용된다는 것을 알 수 있다. 사용자의 수가 많다는 것은 다시 말하면 사용자의 질의가 상당히 많다는 것을 의미한다.

다음으로 다중 점 질의의 특성을 살펴보자. 다중 점 질의에서 전체 데이타의 수를 N, 하나의 질의가 가질 수 있는 데이타 수의 최대값을 k라고 하면, 가능한 질의의 수는 다음과 같이 계산된다.

$$(\text{가능한 질의 수}) = \sum_{i=1}^{k} ({}_{N}C_{i})$$

예를 들어, 데이타 수가 100 개이고 하나의 질의가 접근하는 데이타 개수의 최대값을 5라고 하면, 가능한 질의의 수는 79375495 이다. 즉, 다중 점 질의에서 가능한 질의의 수는 매우 크다는 것을 알 수 있다. 이것은 같은 질의가 나올 확률이 그만큼 적다는 것을 의미하기도 한다.

즉, 방송 데이타들에 대한 다중 점 질의는 그 수가 매우 많고, 접근 빈도의 차이가 크지 않을 것으로 예상된다. 각 질의의 데이타 집합 단위로 클러스터링 하는 방법은 질의 단위로 클러스터링이 수행되므로 질의의 수에 크게 영향을 받게 되며, 질의의 수가 많은 경우 전체 질의를 고려하기 전에 스케줄이 결정되는 경우도 생길 수 있다. 이에 대해 대안으로 생각할 수 있는 방법으로 전체 질의를 분석하여 데이타간의 상관 관계를 결정한 다음, 이를 이용해 데이타 단위로 클러스터링 하는 방법이 있다. 이 경우 클러스터링 전에 전체 질의에 대해 분석을 수행하기 때문에 클러스터링 과정 자체는 질의의 수에 영향을 받지 않게 되며, 전체 클러스터링 과정 동안 모든 질의의 정보를 이용할 수 있다. 이 때 중요한 것은 데이타간의 상관 관계를 정의하는 것으로 전체 질의의 특성을 반영할 수 있도록 해야 하겠다. 본 논문에서는 이와 같은 방법으로 무선 방송 데이타를 클러스터링 하는 방법을 제안한다.

3.2 데이타 친화도(Data Affinity)

본 논문에서는 전체 질의를 통해 나타난 데이타간의 상관 관계를 데이타 친화도(data affinity)라고 정의한다. 데이타 친화도는 전체 질의에서 각각의 데이타 쌍들이 같이 나타나는 정도를 나타낸다.

정의 1. 데이타 개체 d_i와 d_j의 데이타 친화도 $aff(d_i, d_j)$는 다음과 같이 정의 된다.

$$aff(d_i, d_j) = \begin{cases} \sum_{\forall q_k \in Q} \frac{has(q_k, d_i, d_j) \times freq(q_k)}{N_{data}(q_k) C_2}, & \text{if } i \neq j \\ 0, & otherwise \end{cases}$$

$$has(q_k, d_i, d_j) = \begin{cases} 1, & \text{if } d_i \in q_k \wedge d_j \in q_k \\ 0, & otherwise \end{cases}$$ □

데이타 개체 d_i와 d_j의 데이타 친화도는 기본적으로 d_i와 d_j를 모두 포함한 질의들의 참조 빈도의 합으로 볼 수 있다. 따라서 다수의 질의에 같이 포함된 데이타 쌍일수록 데이타 친화도 값이 커진다. 이 때 각 질의의 참조 빈도를 그 질의의 데이타 집합 내에서 구성 가능한 데이타 쌍의 수인 ${}_{N_{data}(q_k)}C_2$로 나누어 준다. 이렇게 하는 이유는 질의에 포함된 데이타의 수가 많을수록 각 데이타 쌍이 그 질의의 접근 시간에 미치는 영향이 감소하기 때문이다. 데이타 2개를 포함한 질의의 경우 접근 시간이 하나의 데이타 쌍의 클러스터링 결과에 의해 결정되므로 질의의 참조 빈도가 전부 그 데이타 쌍의 데이타 친화도에 반영된다. 반면에 질의가 포함하는 데이타의 수가 커질수록 접근 시간은 많은 수의 데이타 쌍의 클러스터링 결과에 의해 결정된다. 따라서, 각 데이타 쌍이 질의의 접근 시간에 기여하는 정도가 감소하므로, 질의의 참조 빈도를 데이타 쌍들이 나누어 가지도록 한다.

성질 1. 데이타 친화도는 다음과 같은 성질을 가진다.

1) 데이타 쌍 d_i, d_j를 모두 포함하는 질의가 없으면 $aff(d_i, d_j) = 0$ 이다.
2) 데이타 친화도는 대칭(symmetric)이다. 즉, $aff(d_i, d_j) = aff(d_j, d_i)$ 이다.
3) 각 데이타 간의 데이타 친화도를 모두 더하면 두 개 이상의 데이타를 가진 모든 질의들의 참조 빈도의 합과 같다.

   즉, $\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} aff(d_i, d_j) = \sum_{q_k \in Q \wedge N_{data}(q_k) > 1} freq(q_k)$이다. □

즉, 데이타 친화도는 전체 질의의 참조 빈도를 각 데이타 쌍에 분배한 것으로 큰 값을 가지는 데이타 쌍일수록 전체 질의의 접근 시간에 많은 영향을 미친다. 따라서 데이타 친화도가 높은 데이타들을 우선적으로 클러스터링 하여야 한다.

전체 데이타들의 데이타 친화도를 행렬로 표현한 것이 친화도 행렬이다.

정의 2. 전체 데이타의 수를 N이라고 했을 때, 친화도 행렬(AM: Affinity Matrix)은 N x N 의 크기를 가진 행렬로 각 항의 값은 다음과 같이 정의 된다.

$$AM[i][j] = aff(d_i, d_j)$$ □

3.3 세그먼트 친화도(Segment Affinity)

제안하는 클러스터링 기법은 세그먼트(segment)들을 병합(merging)해 나가는 방식으로 스케줄을 생성한다. 초기에는 각 세그먼트가 하나의 데이타를 가지며, 병합이 일어나면 세그먼트는 다수의 데이타를 가지게 된다. 이 때, 세그먼트 내의 데이타들은 이미 클러스터링 되어 순서가 고정되어 있다.

정의 3. 세그먼트란 하나 이상의 순서가 고정된 데이타들의 집합이며 다음과 같이 표기한다.

$$S_i = (d_1, d_2, \ldots d_k) \quad \square$$

세그먼트를 병합하는 데 있어 취하는 전략은 다음과 같다.

- 각 세그먼트는 자신이 포함하고 있는 데이타와 데이타 친화도가 높은 데이타를 가진 세그먼트와 우선적으로 병합된다.
- 세그먼트를 병합할 시에는 데이타 친화도가 큰 데이타 쌍일수록 가까이 위치하도록 병합한다.

본 논문에서는 위와 같이 세그먼트의 병합을 수행하기 위해 세그먼트간의 친화도를 정의한다.

정의 4. 임의의 세그먼트 S_i와 S_j의 세그먼트 친화도 $SegAff(S_i, S_j)$는 다음과 같이 정의 된다.

$$SegAff(S_i, S_j) = \begin{cases} \sum_{d_k \in S_i} \sum_{d_l \in S_j} aff(d_k, d_l) \\ \times \dfrac{MaxDist(d_k, d_l) - dist(d_k, d_l)}{MaxDist(d_k, d_l)}, & \text{if } i \neq j \\ 0, & otherwise \end{cases} \quad \square$$

세그먼트 친화도는 기본적으로 각 세그먼트에 속한 데이타들간의 데이타 친화도의 합이다. 따라서 각 세그먼트에 속한 데이타들간의 데이타 친화도가 클수록 세그먼트 친화도 값은 증가한다. 이 때, 친화도 값에 데이타간의 간격에 의해 결정되는 인수 $\dfrac{MaxDist(d_k, d_l) - dist(d_k, d_l)}{MaxDist(d_k, d_l)}$ 값을 곱해주게 된다.

$dist(d_k, d_l)$는 방송 스트림에서 d_k과 d_l의 최소 거리(minimal distance)를 의미하며 $MaxDist(d_k, d_l)$ 는 $dist(d_k, d_l)$ 의 최대값을 의미한다. 각각은 다음과 같이 계산될 수 있다

정의 5. d_k와 d_l 사이의 간격을 δ라 하면 d_k와 d_l의 최소 거리 $dist(d_k, d_l)$와 $dist(d_k, d_l)$의 최대값 $MaxDist(d_k, d_l)$은 다음과 같이 정의된다.

$$dist(d_k, d_l) = \begin{cases} |d_k| + \delta + |d_l|, & \text{if } \delta < \dfrac{BSize - |d_k| - |d_l|}{2} \\ BSize - \delta, & otherwise \end{cases}$$

$$MaxDist(d_k, d_l) = \frac{BSize + |d_k| + |d_l|}{2} \quad \square$$

이와 같이 방송 스케줄 내에서의 거리가 아닌 방송 스트림에서의 최소 거리를 고려하는 이유는 방송 스케줄이 주기적으로 반복되어 방송되기 때문이다. 실제로 방송 스케줄의 맨 첫 데이타와 맨 마지막 데이타는 하나의 방송 스케줄 안에서는 가장 멀리 위치하지만 방송 스트림에서는 인접한 위치에 있게 된다.

$\dfrac{MaxDist(d_k, d_l) - dist(d_k, d_l)}{MaxDist(d_k, d_l)}$ 값은 데이타 쌍이 방송 스트림에서 인접하게 될수록 1에 가까운 값을 가지고, 방송 스트림에서 데이타 쌍의 거리가 멀어질수록 0에 가까운 값을 가지게 된다. 따라서 세그먼트 친화도 값은 데이타 친화도가 높은 데이타들이 방송 스트림에서 인접하게 될수록 큰 값을 가진다.

데이타 친화도와 달리 세그먼트 친화도는 비대칭(asymmetric)이다. 이것은 세그먼트의 배열 순서에 따라 데이타 간의 간격이 달라지기 때문이다. 따라서 세그먼트의 병합 과정에서는 병합될 세그먼트들의 결정 뿐만 아니라 세그먼트의 배열 순서도 결정해야 함을 알 수 있다.

병합을 수행할 때마다 세그먼트 친화도를 계산하는 것은 비효율적이므로 전체 세그먼트간의 세그먼트 친화도를 행렬의 형태로 유지한다.

정의 6. 전체 세그먼트의 수를 N이라고 했을 때, 세그먼트 친화도 행렬(SAM: Segment Affinity Matrix)은 N x N 의 크기를 가진 행렬로 각 항의 값은 다음과 같이 정의 된다.

$$SAM[i][j] = SegAff(S_i, S_j) \quad \square$$

3.4 역 세그먼트 (Inverse Segment)

임의의 방송 스케줄은 대칭인 스케줄과 접근 시간이 같은 특징이 있다. 대칭인 스케줄이란 데이타를 원래 스케줄의 역순으로 배치한 스케줄을 말한다. 즉,

$\sigma_1 = << d_1, d_2, \ldots, d_{i-1}, d_i, d_{i+1}, \ldots, d_{n-1}, d_n >>$

$\sigma_2 = << d_n, d_{n-1}, \ldots, d_{i+1}, d_i, d_{i-1}, \ldots, d_2, d_1 >>$

이면 σ_1와 σ_2 는 전체 접근 시간이 같다. 접근 시간을 계산한 식 (1) 에 나타나듯이 접근 시간은 질의에 포함된 데이타들의 간격에 의해 결정된다. 대칭인 스케줄은 원래 스케줄과 비교했을 때 모든 데이타간의 간격이 그대로 유지되므로 , 임의의 방송 스케줄은 대칭인 스케줄과 접근 시간이 같은 것이다.

이와 비슷한 성질을 세그먼트에서도 찾을 수 있다. 같은 데이타들을 가지고 있으나, 순서가 역순인 세그먼트를 역 세그먼트(Inverse Segment)라고 한다.

정의 7. 임의의 세그먼트 S_i 가 $S_i = (d_1, d_2, \ldots, d_n)$ 일 때, S_i의 역 세그먼트 S_i^{-1}은 다음과 같이 정의된다.

$$S_i^{-1} = (d_n, d_{n-1}, \ldots, d_1) \quad \square$$

역 세그먼트는 세그먼트 내의 데이타들의 간격을 그대로 유지하므로 기존의 클러스터링 결과를 그대로 보존한다. 따라서 세그먼트의 병합 과정에서 세그먼트의 역을 취해 병합하는 방법을 고려해 볼 수 있다. 역 세그먼트는 다음과 같은 성질을 가진다.

성질 2. 임의의 세그먼트 S_i, S_j의 세그먼트 친화도와 S_j^{-1}, S_i^{-1}의 세그먼트 친화도는 같다.

$$SegAff(S_i, S_j) = SegAff(S_j^{-1}, S_i^{-1}) \quad \square$$

이것은 S_i와 S_j를 병합한 뒤 역을 취하면 S_j^{-1}와 S_i^{-1} 이 병합된 세그먼트와 같은 결과가 나오기 때문이다.

세그먼트의 배열 순서와 역 세그먼트를 고려하면, 임의의 두 세그먼트 S_i, S_j에 대해 병합이 가능한 경우는 8 가지이며 각 경우의 세그먼트 친화도는 다음과 같다.

$SegAff(S_i, S_j)$, $SegAff(S_j, S_i)$
$SegAff(S_i^{-1}, S_j)$, $SegAff(S_j^{-1}, S_i)$
$SegAff(S_i, S_j^{-1})$, $SegAff(S_j, S_i^{-1})$
$SegAff(S_i^{-1}, S_j^{-1})$, $SegAff(S_j^{-1}, S_i^{-1})$

이 때, 성질 2 에 의해,

$SegAff(S_i, S_j) = SegAff(S_j^{-1}, S_i^{-1}), SegAff(S_i^{-1}, S_j)$
$= SegAff(S_j^{-1}, S_i)$

$SegAff(S_i, S_j^{-1}) = SegAff(S_j, S_i^{-1}), SegAff(S_i^{-1}, S_j^{-1})$
$= SegAff(S_j, S_i)$

이므로, 위의 4 가지 경우에 대한 세그먼트 친화도 값만 알고 있으면 모든 세그먼트의 병합 형태에 대해 세그먼트 친화도 값을 얻을 수 있다. $SegAff(S_i, S_j)$와 $SegAff(S_j, S_i)$의 경우 세그먼트 친화도 행렬을 통해 얻을 수 있으므로, $SegAff(S_i^{-1}, S_j)$와 $SegAff(S_i, S_j^{-1})$ 값에 대해서만 추가적으로 정보를 유지하면 된다. 역 세그먼트 친화도 행렬은 병합 과정에서 $SegAff(S_i^{-1}, S_j)$와 $SegAff(S_i, S_j^{-1})$의 값을 유지하기 위한 행렬이다.

정의 8. 전체 세그먼트의 수를 N이라고 했을 때, 역 세그먼트 친화도 행렬(ISAM: Inverse Segment Affinity Matrix)은 N x N 의 크기를 가진 행렬로 각 항의 값은 다음과 같이 정의 된다.

$$ISAM[i][j] = \begin{cases} SegAff(S_i^{-1}, S_j), & \text{if } i > j \\ SegAff(S_i, S_j^{-1}), & otherwise \end{cases} \quad \square$$

3.5 알고리즘 (Algorithm)

본 논문에서 제안하는 방법은 먼저 각 세그먼트가 하나의 데이타를 갖도록 초기화한 뒤 세그먼트들을 병합해 나가는 방식으로 진행된다. 세그먼트 친화도 행렬과 역 세그먼트 친화도 행렬에서 최대값을 가지는 항을 통해 병합이 되는 세그먼트들과 병합되는 형태를 결정한다. 제안하는 방법의 기본적인 알고리즘은 그림 4와 같다.

먼저 데이타 집합과 질의 집합에서 데이타 친화도를 계산하여 친화도 행렬을 생성한다. 그리고 각 세그먼트가 하나의 데이타 개체를 가지도록 초기화한 뒤 세그먼트 친화도 행렬과 역 세그먼트 친화도 행렬을 생성한다. 역 세그먼트 친화도 행렬은 초기에는 세그먼트 친화도 행렬과 같은 값을 가진다. 다음으로 세그먼트 친화도 행렬과 역 세그먼트 친화도 행렬을 이용하여 세그먼트들을 병합해 나간다. 세그먼트 친화도 행렬과 역 세그먼트 친화도 행렬에서 최대값을 가지는 항을 통해 병합될 세그먼트와 병합되는 형태를 결정하며, 병합이 이루어질 때마다 새로운 세그먼트가 생겨나므로 새로운 세그먼트에 대해 세그먼트 친화도 행렬과 역 세그먼트 친화도 행렬의 해당 행과 열을 수정한다. 이와 같이 병합을 계속 진행해 나가다가, 모든 세그먼트 사이의 세그먼트 친화도가 0이 되면 병합 과정을 종료한다. 모든 세그먼트 사이의 세그먼트 친화도가 0이라는 것은, 서로 다른 세그먼트의 데이타들을 요구하는 질의가 없다는 것을 의미하므로, 세그먼트들을 어떤 순서로 병합하더라도 접근 시간은 같게 된다.

Algorithm Broadcast_Data_Clustering
INPUT : a set of data objects D, and a set of queries Q
OUTPUT : a broadcast schedule σ
METHOD :
1) Make AM from D and Q.
2) Initialize each segment S_i to have a data object d_i.
3) Make SAM and ISAM.
4) Do
5) Find Segment S_i, S_j and Merging order where the Segment Affinity is the maximum.
6) Merge S_i and S_j following the Merging order.
7) Recalculate SAM and ISAM for the new segment.
8) Until(All values of SAM are 0)
9) Merge remaining Segments

그림 4 제안하는 방법의 기본 알고리즘

알고리즘의 시간 복잡도(Time Complexity)는 병합 과정에 의해 결정된다. 전체 데이타의 수를 n이라고 했을 때, 병합이 가장 많이 일어나는 경우 n - 1번의 병합이 일어나게 된다. 그리고 병합이 일어날 때마다 n^2의 크기를 가지는 세그먼트 친화도 행렬과 역 세그먼트 친화도 행렬을 검색해야 하므로 전체 알고리즘의 시간 복잡도는 $O(n^3)$이 된다. 기존의 SEM과 GCM 방법의 시간 복잡도가 각각 $O(n^2)$, O(log n)인 것과 비교하여 복잡도가 증가하였지만, 방송 데이타의 클러스터링 작업이 방송 주기별로 한번씩 이루어지는 작업인 점을 고려하면, 사용할 수 있는 시간 복잡도라고 판단된다.

4. 성능 평가

이 장에서는 기존의 클러스터링 방법과 본 논문에서 제안한 클러스터링 방법의 성능을 실험을 통해 비교 평가 한다. 비교 대상이 되는 클러스터링 방법은 3장에서 소개한 SEM (Schedule Expansion Method)과 GCM (Gray Coding Method)이며, 성능은 전체 질의의 평균

접근 시간으로 평가하였다. 평균 접근 시간은 전체 접근 시간을 전체 질의의 참조빈도로 나눈 것이다.

질의에 나타나는 데이타들의 접근 빈도는 *Zipf* 분포를 따르도록 하였다. *Zipf* 분포는 데이타들에 대한 접근 빈도를 나타내는 데 일반적으로 많이 사용되며, 데이타 d_i의 접근 빈도는 $(1/i)$ 에 비례한다.[11, 12] *Zipf* 분포에 따라 데이타 d_1 이 접근 빈도가 가장 크며, $d_{DataRange}$ 가 접근 빈도가 가장 작은 데이타가 된다. 실험을 수행하는 데 있어 고려한 패러미터들은 다음과 같다.

- **데이타 개체의 수** : 방송 채널을 통해 사용자에게 전달되는 데이타 개체의 수로서 각 데이타 개체들은 최소한 하나 이상의 질의에 의해 참조된다.
- **질의의 수** : 방송 데이타를 참조하는 질의 유형의 수로서 각 질의는 최소한 하나 이상의 방송 데이타를 참조한다.
- **질의가 포함하는 데이타의 수** : 방송 스트림 중에서 한 질의에 의해 참조되는 데이타의 수이다. 본 논문에서는 한 질의가 포함하는 데이타의 수가 다양한 경우를 가정하고 있으므로, 각 질의가 포함하는 데이타의 수의 평균을 변화시키면서 실험을 수행하였다.

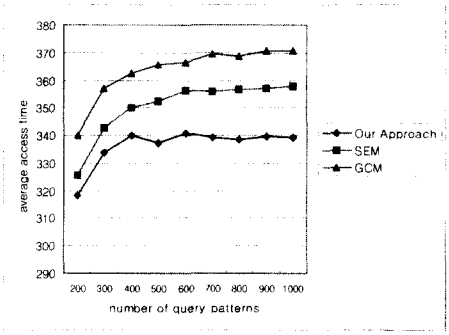

그림 5 질의의 수의 변화에 따른 실험 결과

그림 5는 방송 데이타를 참조하는 질의 유형의 수의 변화시키면서 실험한 결과를 나타낸다. 데이타 개체의 수는 500 개이며, 각 질의는 평균적으로 5개의 데이타를 갖도록 생성하였다.

그림에서 나타나듯이 본 논문에서 제안한 방법이 가장 좋은 성능을 가지며 그 다음으로 SEM, GCM 순서로 성능이 좋은 것을 알 수 있다. 그리고, 기존의 방법들이 질의의 수가 증가함에 따라 접근 시간이 조금씩 증가하는 데 반해 본 논문에서 제안한 방법은 접근 시간이 일정한 선에서 계속 유지 됨을 알 수 있다. 이것은 질의 단위로 클러스터링을 수행하는 기존 방법과 데이타 단위로 클러스터링을 수행하는 본 논문의 방법의 접근 방식의 차이에서 기인한 것으로 데이타 단위로 클러스터링을 수행했을 경우 질의의 수의 증가에 크게 영향을 받지 않는다는 것을 알 수 있다.

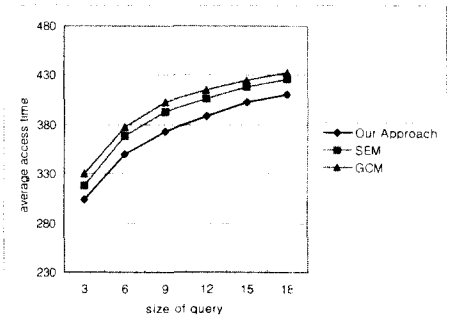

그림 6 질의가 포함하는 데이타 수의 변화에 따른 실험 결과

그림 6은 한 질의에 포함된 데이타 수를 변화시키면서 실험한 결과를 나타낸다. 데이타 개체의 수와 질의 유형의 수는 각각 500 개이며, 한 질의가 포함하는 데이타 수의 평균을 변화시키면서 실험을 수행하였다.

모든 방법에서 질의가 포함하는 데이타의 수가 증가할수록 접근 시간이 증가하였다. 이는 접근 시간에는 데이타를 수신하는 시간이 포함되므로 데이타의 수가 증가할수록 접근 시간이 증가하기 때문이다. 기존 방법들에 비해 제안하는 방법이 가장 좋은 성능을 나타내었으며 성능의 차이는 일정한 간격을 유지하였다.

그림 7은 방송을 통해 사용자에게 전달되는 데이타 개체의 수를 변화시키면서 실험한 결과를 나타낸다. 질의

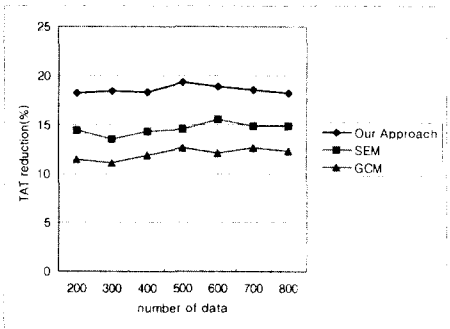

그림 7 데이타 개체 수의 변화에 따른 실험 결과

유형의 수는 각각 500 개이며, 한 질의가 포함하는 데이타 수는 평균적으로 전체 데이타 수의 1.5%가 되도록 설정하였다. 이 경우에 데이타 개체의 수가 변화함에 따라 방송 스케줄의 길이가 달라지기 때문에 TAT 감소 정도(TAT reduction)를 측정 기준으로 삼았다. TAT 감소 정도는 클러스터링 하지 않은 방송 스트림과 비교했을 때, TAT가 감소한 정도를 백분율로 나타낸 것이다.

그림에 나타나듯이 모든 방법에서 데이타 개체 수의 변화는 성능과 무관함을 알 수 있다. 모든 경우에 대해 본 논문에서 제안한 방법이 가장 좋은 성능을 나타내며 성능 향상 정도는 일정하게 유지된다.

5. 결 론

본 논문에서는 무선 데이타 방송에서 사용자가 둘 이상의 데이타를 요구하는 경우에 접근 시간을 줄이기 위한 방송 데이타 클러스터링 기법에 대해 논의하였다. 기존 방법들의 경우 참조 빈도가 높은 질의들부터 각 질의가 포함하는 데이타 집합 단위로 클러스터링하는 방식을 사용한다. 따라서 클러스터링을 수행하는 과정에서 현재 처리되고 있는 질의의 정보만 이용하기 때문에, 질의들의 참조 빈도의 차이가 적고, 질의의 수가 증가할수록 성능이 감소하는 단점이 있다.

본 논문에서는 기존 방식과 달리 먼저 전체 질의를 분석하여 각 데이타간의 데이타 친화도를 정의하고 이에 기반하여 데이타들을 병합하면서 클러스터링을 수행한다. 따라서 클러스터링 과정 전반에서 전체 질의의 정보를 이용하게 된다. 데이타 친화도란 데이타들이 같은 질의에 나타나는 정도를 나타내는 것으로 데이타 친화도가 클수록 전체 질의의 접근 시간에 미치는 영향이 커진다. 따라서 본 논문에서는 세그먼트 친화도라는 기준을 이용하여 데이타 친화도가 큰 데이타들이 방송 스케줄에서 가까이 위치하도록 클러스터링 한다. 실험을 통한 성능 평가에 나타나듯이, 제안한 방법은 질의의 수가 증가하여도 접근 시간을 일정하게 유지시켜준다. 따라서, 방송 기법과 다중 점 질의의 특성에 부합한 클러스터링 기법이라 할 수 있다.

참 고 문 헌

[1] D. Barbara, "Mobile Computing and Database : A Survey", *IEEE Trans. On Knowledge and Data Eng.*, pages 108-117, 1999.

[2] T. Imielinski, and B. Badrinath, "Mobile Wireless Computing : Challenges in Data Management", *CACM*, pages 18-28, Oct 1994

[3] T. Imielinski, and S. Viswanathan, "Adaptive wireless information systems", *Proceedings of SIGDBS Conference*, 1994

[4] S. Acharya, R. Alonso, M. Franklin, and S. Zdonik, "Broadcast Disks : Data Management for Asymmetric Communication Environments", *Proc.ACM SIGMOD Int'l Conf. Management of Data*, pages 199-210, May 1995.

[5] T. Imielinski, S. Viswanathan and B. R. Badrinath. "Data on air : Organization and access", *IEEE Trans. On Knowledge and Data Eng.*, 9(3), 1997.

[6] T. Imielinski, S. Viswanathan and B. R. Badrinath. "Energy efficient indexing on air", *Proceedings of ACM SIGMOD Conference*, pages 25-36, 1994.

[7] D. Barbara and T. Imielinski, "Sleepers and Workaholics : Caching Strategies in Mobile Environments", *Proc. ACM SIGMOD Int'l Conf. Management of Data*, pages 1-12, May 1994.

[8] K. Wu, P. S. Yu, and M. Chen, "Energy-Efficient Caching for Wireless Mobile Computing", *Proc. 12th Int'l Conf. Data Eng.*, pages 336-343, 1996.

[9] Y. D. Chung and M. H. Kim, "QEM: A Scheduling Method for Wireless Broadcast Data", *Proc. of 6th Int'l Conf. on Database Systems for Advanced Applications*, pages 135-142, April 1999.

[10] Y. D. Chung and M. H. Kim, "Linear Placement of Wireless Broadcast Data Using Gray Codes", *Journal of Electrical Engineering and Information Science*, pages 459-465, 1999.

[11] J. Gray, et al., "Quickly Generating Billion-Record Synthetic Databases", *Proc. ACM SIGMOD Conf.*, pages 243-252, May 1994.

[12] D. Knuth, "The Art of Computer Programming, Vol II", *Addison Wesley*, 1981.

방 수 호

1999년 연세대학교 컴퓨터과학과 학사. 2001년 한국과학기술원 전산학과 석사. 현재 한국무역정보통신 연구원. 관심분야는 Mobile Information Systems, Database Systems, eBusiness Frameworks 등임

정 연 돈

정보과학회논문지 : 데이타베이스 제 28 권 제 1 호 참조

김 명 호

정보과학회논문지 : 데이타베이스 제 28 권 제 1 호 참조