그리드 분할에 의한 다차원 데이터 디클러스터링 성능 분석

김 학 철[†] · 김 태 완[††] · 이 기 준[†††]

요 약

대규모의 데이터를 다루는 여러 시스템에서 데이터를 다수의 병렬 디스크에 분산시켜 저장한 후 질의 처리시 동시에 여러 개의 디스크를 접근함으로써 입출력 성능의 향상을 위한 많은 노력들이 행해져 왔다. 대부분 이전 연구들은 데이터 공간을 이루는 각 차원이 겹치지 않는 여러 개의 구간으로 나누어져 전체 데이터 공간이 그리드 형태로 분할되어 있다는 가정하에 각 차원의 구간 번호로 결정되는 그리드 셀에 대해서 효과적으로 디스크 번호를 할당하는 알고리즘 개발에 집중되었다. 하지만, 그들은 데이터 공간을 그리드 형태로 분할하는 방법이 전체 디클러스터링 알고리즘 성능에 미치는 영향을 간과하였다. 본 논문에서 우리는 효과적인 그리드 분할을 통하여 매핑 함수를 이용하는 디클러스터링 알고리즘의 성능을 향상 시켰다. 이를 위하여 영역 질의 크기가 주어졌을 때 겹치는 그리드 셀의 수를 예측하는 모델을 제시하였으며 이를 이용하여 가능한 그리드 분할 방법들 중에서 질의 크기를 감소시키는 분할 방법을 선택하였다. 일반적으로, 다차원 데이터에 대해서는 이진 분할을 하지만 본 논문에서는 더 작은 수의 차원을 선택해서 여러 번 분할함으로써 질의를 만족하는 그리드 셀의 수를 감소시켰다. 다양한 실험 결과에 의하면 본 논문에서 제시한 예측 모델은 질의 크기와 차원에 관계없이 0.5% 이내의 에러율을 보이는 것으로 나타났다. 또한 효과적인 그리드 분할을 통하여 다차원 데이터에 대해서 가장 성능이 좋은 것으로 소개 되고 있는 Kronecker sequence 매핑 함수를 이용하는 디클러스터링 알고리즘의 성능을 최대 23배까지 향상시킬 수 있음을 알 수 있었다.

Performance Analysis on Declustering High-Dimensional Data by GRID Partitioning

Hak-Cheol Kim[†] · Tae-Wan Kim[††] · Ki-Joune Li[†††]

ABSTRACT

A lot of work has been done to improve the I/O performance of such a system that store and manage a massive amount of data by distributing them across multiple disks and access them in parallel. Most of the previous work has focused on an efficient mapping from a grid cell, which is determined by the interval number of each dimension, to a disk number on the assumption that each dimension is split into disjoint intervals such that entire data space is GRID-like partitioned. However, they have ignored the effects of a GRID partitioning scheme on declustering performance. In this paper, we enhance the performance of mapping function based declustering algorithms by applying a good GRID partitioning method. For this, we propose an estimation model to count the number of grid cells intersected by a range query and apply a GRID partitioning scheme which minimizes query result size among the possible schemes. While it is common to do binary partition for high-dimensional data, we choose less number of dimensions than needed for binary partition and split several times along that dimensions so that we can reduce the number of grid cells touched by a query. Several experimental results show that the proposed estimation model gives accuracy within 0.5% error ratio regardless of query size and dimension. We can also improve the performance of declustering algorithm based on mapping function, called Kronecker Sequence, which has been known to be the best among the mapping functions for high-dimensional data, up to 23 times by applying an efficient GRID partitioning scheme.

키워드 : 디클러스터링(Declustering), 병렬 입출력(Parallel I/O), 다차원 데이터(High-Dimensional Data), 그리드 분할(GRID Partition), 성능분석 모델(Estimation Model)

1. 서 론

지리 정보 시스템(GIS : Geographic Information System), 정보 검색 시스템(Information Retrieval System), CAD, Remote Sensing Database 등 대용량의 다차원 데이터를 다루는 시스템에서 전체 성능은 디스크 입출력 성능에 의해서 크게 영향을 받는다. 저장 시스템의 발달로 인해 위와 같은 시스템에서 대용량의 데이터를 저장하는 기술은 가능해 졌지만 여전히 효과적인 디스크 접근 문제는 해결해야 할 과제로 남아 있다. 이를 해결하기 위해 데이터를 여러 개의 병렬 디스크에 분산 배치한 후 질의 처리시 동시에 여러 개의 디스크를 접근함으로써 디스크 접근 시간(seek time)을 줄이기 위한 연구들이 많이 행해 졌으며 우리는 이를 "디클러스터링

※ 본 연구는 산업자원부의 지역혁신 인력양성사업의 연구결과로 수행되었음. 본 논문은 과학기술부 한국과학재단 지정 한국항공대학교 부설 인터넷정보검색연구센터의 지원에 의함.
† 준 회 원 : 부산대학교 대학원 전자계산학과
†† 정 회 원 : 행정자치부 전자정부전략개발실 전문위원
††† 정 회 원 : 부산대학교 정보 컴퓨터공학부 교수
논문접수 : 2004년 5월 12일, 심사완료 : 2004년 7월 1일

(declustering)"이라고 부른다.

데이터 디클러스터링을 위해서는 전체 데이터 집합을 디스크 물리적 블록 크기 단위로 분할하는 과정과 각각의 그룹을 이용 가능한 디스크에 효과적으로 분산하여 저장하는 2 단계의 과정이 필요하다. 이 때, 디클러스터링 알고리즘의 성능은 첫 번째 단계에서 결정되는 질의를 만족하는 데이터 블록의 수와 두 번째 단계에서 결정되는 디스크 접근의 병렬성에 의해서 결정된다. 하지만, 대부분의 이전 연구들은 분할 방법이 디클러스터링 성능에 미치는 영향은 간과한 채 분할 결과에 대해서 효과적으로 디스크 번호를 할당하는 알고리즘 개발에 집중되었다[1-8, 12]. 그들은 데이터 공간을 이루는 각 차원이 겹치지 않는 여러 구간으로 분할되어 전체 데이터 공간이 그리드 형태로 분할되어 있다는 가정하에 각 차원의 구간 번호로 결정되는 그리드 셀에 대해서 효과적으로 디스크 번호를 할당하는 매핑 함수 연구에 치중하였다.

Disk Modulo[5], Field-wise Xor[8], Error Correcting Code [7], Hilbert Curve Allocation Method[6], Cyclic Allocation Method[12], Coloring[1], Golden Ratio Sequence[2], Discrepancy based mapping[4], Kronecker Sequence[3] 등 다양한 매핑 함수들이 제시되고 평가되었다. 이 중에서 [3]에서 제시된 Kronecker Sequence에 기초한 매핑 함수가 다차원 데이터에 대해서는 가장 성능이 우수한 것으로 소개되었다. 그들은 질의를 만족하는 그리드 셀들이 최대한 서로 다른 디스크에 저장되도록 하는 알고리즘 연구에 집중하였다. 하지만, 우수한 매핑 함수를 적용하더라도 질의를 만족하는 데이터 블록의 수가 증가할 때에는 전체 성능의 저하를 초래하게 된다.

동일한 매핑 함수를 적용할 때 전체 디클러스터링 알고리즘의 성능은 질의를 만족하는 데이터 블록(그리드 셀)의 수에 의해 결정됨은 명백하다. 본 논문에서는 그리드 분할 방법이 매핑 함수를 이용하는 디클러스터링 알고리즘의 성능에 미치는 영향을 실험을 통해서 보이고 가능한 여러 분할 방법 가운데 질의를 만족하는 그리드 셀의 수를 최소로 하는 분할 방법을 선택하는 알고리즘을 제시한다. 일반적으로 고차원 데이터 공간에 대해서 그리드 형태로 분할할 때 이진 분할(binary partition)을 가정하지만 질의창이 클 경우(질의 창의 한 변의 길이가 도메인 길이의 1/2보다 큰 경우) 모든 데이터 블록을 접근해야 하는 단점이 있다. 차원이 높아지면 극히 낮은 선택률에 대해서도 질의 창은 매우 커진다. 따라서, 고차원 데이터에 대한 이진 분할(binary partition)은 많은 문제점을 야기한다. 이를 해결하기 위해서 이진 분할을 위해서 필요한 차원의 수보다 더 작은 수의 차원을 선택해서 여러 번 분할할 경우 질의를 만족하는 그리드 셀의 수를 감소시킬 수 있다.

고차원 데이터에 대해서 그리드 형태의 분할 방법은 여러 가지가 가능하며 이 가운데 효과적인 분할 방법을 결정하는 것이 그리드 분할 후 매핑 함수를 이용하는 디클러스터링 알고리즘의 성능 향상을 위해서 필수적이다. 이를 해결하기 위하여 본 논문에서는 주어진 그리드 분할 방법에 대해서 영역 질의를 만족하는 그리드 셀의 수를 예측하는 분석 모델을 제시하였다. 이를 이용하면 가능한 분할 방법 가운데 질의창과 겹치는 그리드 셀의 수를 최소로 하는 분할 방법을 선택할 수 있다. 다양한 차원과 질의 크기에 대한 실험 결과에 의하면 본 논문에서 제시한 질의결과 예측 모델은 0.5% 이내의 에러율을 보이는 것으로 나타났다. 또한 가능한 분할 방법 가운데 예측 모델을 이용하여 질의 결과를 최소로 하는 분할 방법을 선택했을 때 이진 분할에 비해서 동일한 디스크 매핑 알고리즘을 적용할 때 최대 23배의 성능 향상을 가져옴을 알 수 있었다.

본 논문은 다음과 같이 구성된다. 2장에서는 디클러스터링 문제와 관련된 개념 및 성능 척도를 재시한다. 3장에서는 대부분 이전 연구들에서 가정하고 있는 그리드 분할 방법의 문제점 및 개선점을 제시하고 4장에서는 그리드 분할에 대해서 질의 결과 크기를 예측하는 모델을 제시하고 이를 이용하여 매핑 함수를 이용하는 디클러스터링 알고리즘의 성능 향상 방법을 제시한다. 5장에서는 실험을 통하여 제시한 예측 모델의 정확성을 검증한 후 분할 방법이 매핑 함수를 이용하는 디클러스터링 알고리즘의 성능에 미치는 영향을 보이도록 한다. 마지막으로 6장에서는 본 논문의 결론을 맺는다.

2. 개념 설명

2장에서는 디클러스터링 문제와 관련된 개념 설명과 함께 성능 척도에 대해서 설명한다. <표 1>은 본 논문에서 앞으로 사용할 기호와 그 의미를 나타낸 것이다.

〈표 1〉 기호 및 의미

기 호	의 미
d	데이터 차원
N	전체 데이터 수
M	디스크 수
B	1개의 디스크 블록에 저장할 수 있는 최대 데이터 수
P	생성된 전체 데이터 블록의 수(그리드 셀의 수)
λ_i	i번째 차원의 분할 횟수
q	영역 질의 Q의 한 변의 길이 ($0 \le q < 1$)
B_Q	질의 Q를 만족하는 데이터 블록(그리드 셀)의 수
α	최적의 디스크 접근 횟수에 대한 추가적인 디스크 접근 횟수
V	d차원 그리드 셀 1개의 체적

디클러스터링의 대상인 데이터를 물리적 디스크에 저장하기 위해서는 디스크 블록 단위로 분할해야 한다. 우리는 데이터 집합 $S=\{p_1, p_2, \cdots, p_N\}$에 대해서 데이터 분할 $\pi(N, B)$를 다음과 같이 정의한다.

[정의 1] 데이터 분할 $\pi(N, B)$

$$\pi(N, B):\{v \mid v \in S\} \rightarrow \{G_1^\pi, G_2^\pi, \cdots, G_P^\pi\},$$

여기서 $|G_i^\pi| \le B$, $\sum_{i=1}^{P} |G_i^\pi| = N$, 그리고 $i \neq j$에 대해서 $G_i^\pi \cap G_j^\pi = \emptyset$ □

이를 바탕으로 M개의 디스크에 대해서 디클러스터링 알고리즘은 다음과 같이 2단계로 정의할 수 있다.

- 단계 1 : 데이터 분할 과정 : $\{v \mid v \in S\} \rightarrow \{G_1^\pi, G_2^\pi, \cdots, G_P^\pi\}$
- 단계 2 : 디스크 할당 과정 : $\{G_1^\pi, G_2^\pi, \cdots, G_P^\pi\} \rightarrow \{0, 1, 2, \cdots, M-1\}$

이 때, 데이터 집합을 분할한 후 M개의 디스크에 분산하여 저장하였을 경우 질의 Q를 처리하기 위한 디스크 접근 횟수 $DA(Q)$는 다음과 같이 결정된다.

[정의 2] 질의 Q를 처리하기 위한 최대 디스크 접근 횟수 : $DA(Q)$

$$DA(Q) = \max_{i=0}^{M-1} DA_i(Q),$$

여기서, $DA_i(Q)$는 질의 Q를 처리하기 위한 i번째 디스크 접근 횟수 □

[정의 2]에 의하면 디클러스터링 알고리즘의 성능 향상을 위해서는 질의를 처리하기 위해서 이용 가능한 모든 디스크에 대한 접근 횟수가 동일해야 함을 알 수 있다. 즉, 질의 조건 Q를 만족하는 데이터 블록이 M개의 디스크에 균일하게 분포되어 있어야 함을 의미한다. 이를 바탕으로 완전한 최적의 디클러스터링 알고리즘을 다음과 같이 정의한다.

[정의 3] 엄격히 최적인 디클러스터링 알고리즘(strictly optimal declustering algorithm)

다음 조건을 만족하는 디클러스터링 알고리즘은 엄격히 최적이다.

$$\forall_Q, DA(Q) = \left\lceil \frac{B_Q}{M} \right\rceil \quad \square$$

특별한 경우를 제외하고 모든 질의에 대해서 위의 조건을 만족하는 디클러스터링 알고리즘은 존재하지 않는다고 알려져 있으며[15] 모든 디클러스터링 알고리즘의 성능은 다음과 같이 결정된다.

$$DA(Q) = \left\lceil \frac{B_Q}{M} \right\rceil + \alpha, \text{ 여기서 } \alpha > 0 \qquad (1)$$

디클러스터링 알고리즘의 성능을 나타내는 식 (1)에서 전체 성능에 영향을 주는 요소는 질의를 만족하는 블록의 수 B_Q와 추가적인 디스크 접근 횟수 α이다. 이 중에서 B_Q는 데이터 분할 방법에 의해서 영향을 받으며 α는 디스크 할당 정책에 의해 결정된다. 따라서, 전체 디클러스터링 알고리즘 성능의 향상을 위해서는 질의처리를 위해서 접근해야 하는 데이터 블록의 수를 감소시켜야 하며 동시에 접근해야 하는 데이터 블록이 전체 디스크 상에 최대한 균일하게 분포해야 한다.

3. 효과적인 그리드 분할

대부분 이전의 디클러스터링 문제에 대한 연구들은 데이터 분할 방법에 대한 연구보다는 효과적인 디스크 할당 알고리즘 연구에 집중되었다[1-8, 12]. 그들은 데이터 공간을 이루는 각 차원이 겹치지 않는 여러 개의 구간으로 나누어져 전체 데이터 공간이 그리드 형태로 분할되어 있다는 가정하에 각 차원의 구간 번호로 결정되는 그리드 셀에 대해서 디스크 번호를 할당하는 알고리즘 연구에 집중하였다. (그림 1)은 2차원 8×8 그리드 분할에 대해서 디스크 수가 4일 때, 다양한 매핑 함수를 적용했을 때 각각의 그리드 셀에 대해서 할당된 디스크 번호를 나타낸 것이다.

3	0	1	2	3	0	1	2
2	3	0	1	2	3	0	1
1	2	3	0	1	2	3	0
0	1	2	3	0	1	2	3
3	0	1	2	3	0	1	2
2	3	0	1	2	3	0	1
1	2	3	0	1	2	3	0
0	1	2	3	0	1	2	3

(a) DM[5]

3	2	1	0	3	2	1	0
2	3	0	1	2	3	0	1
1	0	3	2	1	0	3	2
0	1	2	3	0	1	2	3
3	2	1	0	3	2	1	0
2	3	0	1	2	3	0	1
1	0	3	2	1	0	3	2
0	1	2	3	0	1	2	3

(b) FX[8]

3	2	1	0	3	0	3	2
0	1	2	3	2	1	0	1
3	0	3	0	1	2	3	2
2	1	2	1	0	3	0	1
1	2	1	2	3	0	3	2
0	3	0	3	2	1	0	1
3	2	1	0	1	2	3	2
0	1	2	3	0	3	0	1

(c) HCAM[6]

3	1	2	0	3	1	2	0
2	0	1	3	2	0	1	3
1	3	0	2	1	3	0	2
0	2	3	1	0	2	3	1
3	1	2	0	3	1	2	0
0	3	0	3	2	1	0	1
1	3	0	2	1	3	0	2
0	2	3	1	0	2	3	1

(d) GRS[2]

(그림 1) 매핑 함수에 의한 디스크 할당 예 : 디스크 수는 4

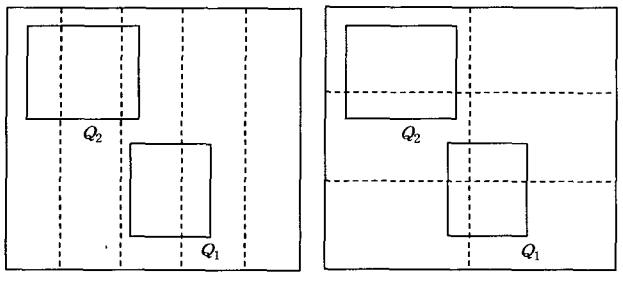
N개의 데이터를 저장하기 위해서는 적어도 $\lceil N/B \rceil$개의 데이터 블록이 필요하며 우리는 d차원 데이터에 대한 그리드 분할을 다음과 같이 정의한다.

[정의 4] 그리드 분할(d, N, B)

다음 조건을 만족하도록 i번째 차원을 λ_i개의 구간으로 나눈다.

$$\prod_{i=1}^{d} \lambda_i \ge \left\lceil \frac{N}{B} \right\rceil, \text{ 여기서 } 1 \le \lambda_i \le \left\lceil \frac{N}{B} \right\rceil \quad \square$$

[정의 4]를 만족하는 각 차원의 분할 횟수의 조합은 여러 가지가 있으며 이는 영역 질의와 겹치는 그리드 셀의 수를 결정한다. (그림 2)는 2차원 데이터에 대해서 필요한 전체 데이터 블록의 수가 5일 때 동일한 영역 질의에 대해서 분할 방법이 미치는 영향을 보여 준다.



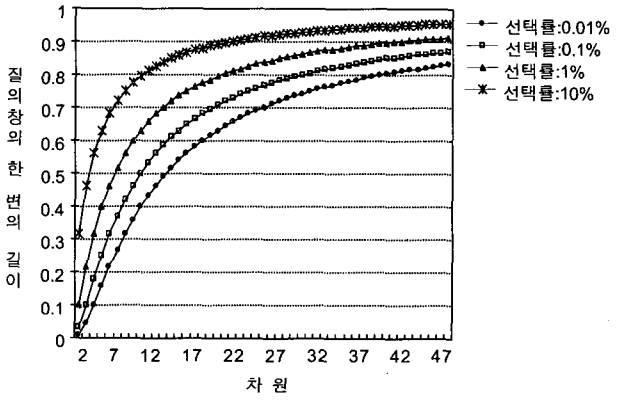
(a) $\lambda_1=5, \lambda_2=1$ (b) $\lambda_1=2, \lambda_2=3$

(그림 2) 분할 방법에 따른 질의를 만족하는 그리드 셀의 수 : $\lceil N/B \rceil = 5$

분할을 위해서 차원을 하나만 선택 했을 때에는 5×1 형태의 그리드 분할이 가능하며((그림 2)(a)). 모든 차원을 사용할 경우 2×3 형태의 그리드 분할이 가능하다((그림 2)(b)). 이 때, 질의 Q_1과 Q_2를 만족하는 그리드 셀의 수는 분할 방법에 따라 다름을 알 수 있다. (그림 2)에서 알 수 있듯이 분할 방법은 생성되는 전체 그리드 셀의 수와 영역 질의를 만족하는 그리드 셀의 수를 결정하며 디클러스터링 알고리즘의 성능 향상을 위해서는 효과적인 분할 방법을 적용하는 것이 필수적이다.

일반적으로 다차원 데이터에 대해서는 이진 분할(binary partition)을 가정한다. 하지만, 차원이 높아질수록 '차원의 저주(Curse of Dimensionality)'현상 때문에 작은 선택률(selectivity)에 대해서도 질의 창이 매우 커진다. (그림 3)은 차원이 증가에 따른 질의창의 크기 변화를 나타낸 것이다.

선택률이 0.1%인 경우에 대해서도 10차원 이상의 데이터에 대해서는 질의창의 한 변의 길이가 0.5이상이 됨을 알 수 있다. 따라서, 이진 분할을 하였을 경우에는 모든 그리드 셀과 겹치게 되며 이 때에는 순차적으로 디스크 번호를 할당하는 것이 가장 효과적이다.



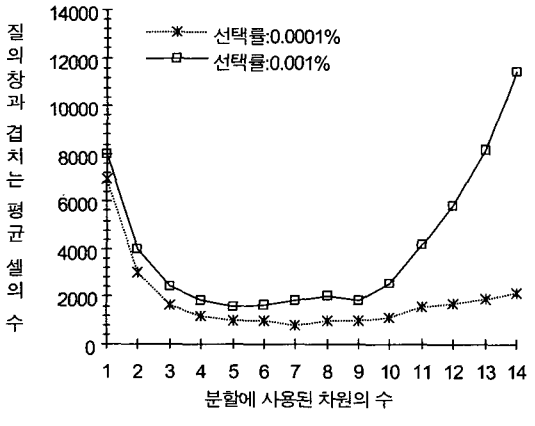
(그림 3) 차원의 변화와 선택률(selectivity)에 따른 질의창의 크기 변화

본 논문에서는 이를 해결하기 위해서 이진 분할에 필요한 차원의 수보다 적은 수의 차원을 선택해서 여러 번 분할하도록 한다. (그림 4)는 필요한 데이터 블록(그리드 셀)의 수가 2^{14}개일 때, 분할에 사용된 차원의 수와 선택률에 따라 질의창과 겹치는 평균 그리드 셀의 수를 나타낸 것이다.

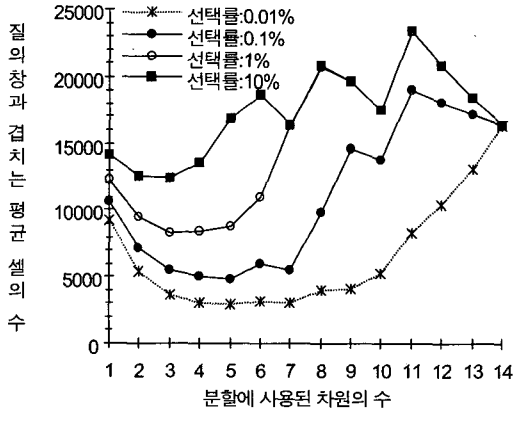
(그림 4)에서 알 수 있듯이 이진 분할(binary partition)에 필요한 차원의 수보다 적은 수의 차원을 선택해서 여러 번 분할하는 것이 오히려 성능의 향상을 가져옴을 알 수 있다. 또한, 질의 크기에 따라서 영역 질의와 겹치는 그리드 셀의 수를 최소로 하는 분할 차원의 수는 다름을 알 수 있다.

다음 장에서 우리는 질의 크기가 주어졌을 때 영역 질의와 겹치는 그리드 셀의 수를 예측하는 모델을 제시한다.

이를 이용하여 다차원 데이터에 대해서 가능한 그리드 분할 방법 가운데 질의를 만족하는 데이터 블록의 수를 감소시키는 것을 선택함으로써 이전 디클러스터링 연구의 대부분



(a) $q < 0.5$



(b) $q \geq 0.5$

(그림 4) 분할에 사용된 차원의 수에 따른 영역 질의를 만족하는 평균 셀의 수

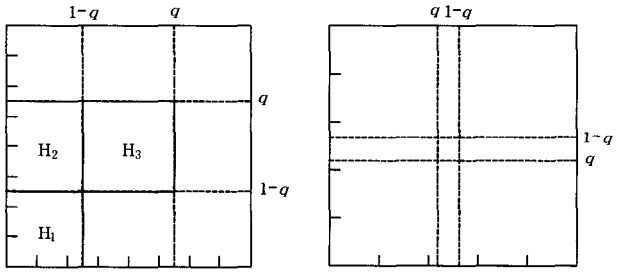
을 차지하고 있는 매핑 함수를 이용하는 디클러스터링 알고리즘의 성능을 향상시킬 수 있다.

4. 효율적인 그리드 분할에 의한 디클러스터링 알고리즘 성능 개선

4장에서는 가능한 분할 방법들 가운데 효율적인 그리드 분할 방법을 선택함으로써 매핑 함수를 이용하는 디클러스터링 알고리즘의 성능을 개선하는 방법을 제시한다.

4.1 그리드 분할 성능 예측 모델

먼저, 그리드 분할에 대해서 영역 질의를 만족하는 그리드 셀의 수를 예측하는 모델을 제시한다. 이를 위해서 d차원 데이터 공간을 이루는 각 차원이 λ개의 구간으로 나누어져 λ^d개의 그리드 셀로 이루어져 있다고 가정한다. 본 논문에서는 민코프스키 합 모델[29]에 기초하여 각각의 그리드 셀에 대해서 기대값을 계산한다. 이 모델에 의하면 각 셀의 기대값은 겹치거나 포함되는 다차원 입방체(hypercube)의 모양에 의해서 달라짐을 알 수 있다. (그림 5)는 2차원 공간에 대해서 3가지 서로 다른 모양이 존재함을 보여준다.



(a) $q \geq 0.5$ (b) $q < 0.5$

(그림 5) 2차원 공간에서의 질의 영역에 의해 분할된 그리드 영역의 유형 예

즉, 질의창의 한 변의 길이 q가 0.5보다 클 경우에는 모든 변의 길이가 $1-q$인 경우(H_1), 한 변은 $1-q$, 다른 한 변은 $2q-1$인 경우(H_2) 그리고 모든 변의 길이가 $2q-1$인 경우(H_3)의 3가지이다. 질의창의 한 변의 길이 q가 0.5보다 작을 경우에는 $2q-1$을 $1-2q$로 대체하면 된다. 각각의 면적은 $(1-q)^2$, $(1-q)(2q-1)$, $(2q-1)^2$이다.

각 차원의 도메인의 길이는 길이가 $2q-1$인 구간 1개와 길이가 $(1-q)$인 구간 2개의 합으로 나타낼 수 있다. 따라서, d차원 $[0,1]^d$ 데이터 공간의 체적 V는 질의창의 한 변의 길이 q를 이용하여 다음과 같이 표현할 수 있다.

$$\begin{aligned} V &= [(2q-1)+2(1-q)]^d \\ &= \sum_{i=0}^{d} C(d,i)\cdot(2q-1)^i\cdot(2(1-q))^{d-i} \\ &= \sum_{i=0}^{d} C(d,i)2^{d-i}(1-q)^{d-i}(2q-1)^i \end{aligned} \quad (2)$$

식 (2)는 $d+1$개의 서로 다른 형태가 가능하며 i번째 형태를 가지는 입방체의 체적은 $(1-q)^{d-i}(2q-1)^i$이고 그 개수는 $C(d,i)2^{d-i}$임을 의미한다. 지금부터 우리는 d차원 데이터에 대해서 i번째 모양을 가지는 입방체의 수를 다음을 만족하는 $COEF(d,i)$로 표기하도록 한다.

$$COEF(d,i) = C(d,i)\cdot 2^{d-i}, \text{ 여기서 } 0 \leq i \leq d \quad (3)$$

영역 질의 Q를 만족하는 셀의 수를 계산하기 위해서 i번째 형태를 가지는 $COEF(d,i)$개의 입방체에 포함되거나 겹치는 그리드 셀 수의 기대값을 계산한다. 먼저, d차원 데이터에 대해서 각 차원이 $\lambda(\geq 2)$번 분할되어 $\prod_{i=1}^{n}\lambda = P(n \leq d)$라고 가정하자. $q \geq 0.5$일 때 $|c| = \left\lfloor \frac{1-q}{1/\lambda} \right\rfloor$ 이고 $|s| = \lambda - 2|c|$로 정한다.

[보조 정리 1] 질의창의 한 변의 길이가 $q(\geq 0.5)$인 영역 질의에 의해서 만들어지는 $d+1$개 유형의 입방체에 대해서 길이가 $2q-1$인 변의 개수가 $i(0 \leq i \leq d)$개이고 길이가 $1-q$인 변의 개수가 $d-i$인 ((그림 5) (a))의 H_1에 포함되거나 (그림 5)(a)의 H_2, H_3와 겹치는 그리드 셀들의 기대값의 합은 다음과 같다.

$$E(H,Q) = |s|^i \cdot \left(\frac{1/\lambda}{1-q} \cdot \sum_{j=0}^{|c|} j \right)^{d-i} \quad (4)$$

[증명] 먼저 2차원 데이터의 경우에 기대값을 제시하고 d차원 경우로 확장한다. 문제를 간단히 하기 위해서 $|c|>0, |s|>0$이라고 가정한다.

$i=0$일 때, 면적은 $(1-q)^2$이며 이는 (그림 5)(a)의 H_1의 경우에 해당된다. 이 때, 이 입방체에 포함되는 셀의 기대값은 다음과 같다.

$$\begin{aligned} E(H,Q) &= \frac{1/\lambda}{1-q}\cdot\frac{1/\lambda}{1-q} + \frac{1/\lambda}{1-q}\cdot\frac{2/\lambda}{1-q} + \cdots + \frac{1/\lambda}{1-q} \\ &\quad \cdot \frac{|c|/\lambda}{1-q} + \frac{2/\lambda}{1-q}\cdot\frac{1/\lambda}{1-q} + \frac{2/\lambda}{1-q}\cdot\frac{2/\lambda}{1-q} + \cdots \\ &\quad + \frac{2/\lambda}{1-q} + \cdots + \cdot \frac{|c|/\lambda}{1-q}\,\frac{|c|/\lambda}{1-q}\cdot\frac{1/\lambda}{1-q} \\ &\quad + \frac{|c|/\lambda}{1-q}\cdot\frac{2/\lambda}{1-q} + \cdots + \frac{|c|/\lambda}{1-q}\cdot\frac{|c|/\lambda}{1-q} \\ &= \frac{1/\lambda}{1-q}\cdot\frac{1/\lambda}{1-q}\sum_{j=1}^{|c|} j + \frac{2/\lambda}{1-q}\cdot\frac{1/\lambda}{1-q}\sum_{j=1}^{|c|} j \\ &\quad + \cdots + \frac{|c|/\lambda}{1-q}\cdot\frac{1/\lambda}{1-q}\sum_{j=1}^{|c|} j \\ &= \frac{1/\lambda}{1-q}\sum_{j=1}^{|c|} j = \left(\frac{1/\lambda}{1-q}\sum_{j=1}^{|c|} j\right)^2 \end{aligned} \quad (5)$$

이를 귀납적으로 d차원으로 확장하면 기대값은 다음과 같다.

$$E(H,Q) = \left(\frac{1/\lambda}{1-q}\sum_{j=1}^{|c|} j\right)^d \quad (6)$$

$i=1$일 때, 면적은 $(2q-1)(1-q)$ 이며 이는 ((그림 5) (a))의 H_2에 해당한다. $(2q-1)$과 겹치고 $(1-q)$에 포함되는 셀의 기대값은 다음과 같다.

$$E(H,Q)=\frac{1/\lambda}{1-q}\cdot|s|+\frac{2/\lambda}{1-q}\cdot|s|+\cdots+\frac{|c|/\lambda}{1-q}\cdot|s| = \frac{1/\lambda}{1-q}\sum_{j=1}^{|c|} j\cdot|s| \quad (7)$$

$i=2$일 때, 면적은 $(2q-1)^2$ 이며 이는 (그림 5)(a)의 H_3에 해당한다. $(2q-1)$과 겹치는 셀의 기대값은 다음과 같다.

$$E(H,Q)=1\cdot|s|+1\cdot|s|+\cdots+1\cdot|s|=|s|^2 \quad (8)$$

d차원의 경우, $(1-q)$에 포함되는 구간들은 $\left(\frac{1/\lambda}{1-q}\sum_{j=1}^{|c|} j\right)$에 귀속되며 $(2q-1)$과 겹치는 셀들은 독립적으로 $|s|$에 귀속된다. 따라서, $d+1$유형들 가운데 1개의 입방체에 포함되거나 겹치는 셀들의 기대값은 다음과 같다.

$$E(H,Q)=|s|^i\left(\frac{1/\lambda}{1-q}\sum_{j=1}^{|c|} j\right)^{d-i} \quad (9)$$

□

이를 이용하여 질의 크기 q에 따라서 겹치는 그리드 셀의 수를 다음과 같은 정리를 할 수 있다.

[정리 1] $\prod_{i=1}^{d}\lambda=P$개의 그리드 셀에 대해서 질의 창의 한 변의 길이가 $q\geq 0.5$인 영역 질의 Q와 겹치는 그리드 셀의 수 $E(Q)$는 다음과 같다.

$$E(Q)=\sum_{j=0}^{d} COEF(d,i)|s|^i\left(\frac{1/\lambda}{1-q}\sum_{j=1}^{|c|} j\right)^{d-i}$$

[증명] $COEF(d,i)$는 $(1-q)^i(2q-1)^{d-i}$유형들의 입방체의 수를 나타내고 $(1-q)^i(2q-1)^{d-i}$입방체의 기대값을 알기 때문에 이는 유효하다. □

지금까지 우리는 $q\geq 0.5$일 때 기대값을 보였다. $q<0.5$일 때에는 $|c|=\lceil q/1/\lambda\rceil$이고 $|s|=\lambda-2|c|$이다.

[정리 2] $\prod_{i=1}^{d}\lambda=P$개의 그리드 셀에 대해서 질의 창의 한 변의 길이가 $q<0.5$인 영역 질의 Q와 겹치는 그리드 셀의 수 $E(Q)$는 다음과 같다.

$$E(Q)=\begin{cases}\sum_{i=0}^{d} COEF(d,i)\left(\frac{1/\lambda}{1-q}\sum_{j=1}^{|c|-1}\right)^{d-i} & |s|<0,\\ \sum_{j=0}^{d} COEF(d,i)\left(\frac{1/\lambda+q}{1-q}|s|\right)^i\left(\frac{1/\lambda}{1-q}\sum_{j=1}^{|c|} j\right)^{d-i} & |s|\geq 0\end{cases}$$

[증명] 증명은 [보조정리 1]과 [정리 1]과 유사하다. □

$|s|<0$인 특수한 경우의 예를 (그림 5)(b)에 나타내었다.

4.2 그리드 분할에 의한 디클러스터링 알고리즘 성능 개선 방법

앞 절에서 우리는 주어진 그리드 분할에 대해서 질의 크기가 주어졌을 때 영역 질의와 겹치는 그리드 셀의 수를 예측하는 모델을 제시 하였다. 4.2절에서는 이를 이용하여 그리드 분할 후 매핑 함수를 이용하여 디스크 번호를 할당하는 디클러스터링 알고리즘의 성능을 개선하는 방법을 제시한다. 1개의 그리드 셀에 포함될 수 있는 데이터의 수를 B라고 할 때, N개의 데이터를 저장하기 위해서 필요한 그리드 셀의 수 P는 $\lceil N/B\rceil$ 이상이다. 이를 위한 각 차원의 분할 횟수 λ_i의 조합은 여러 가지가 가능하며 이는 영역 질의와 겹치는 그리드 셀의 수를 결정한다.

P개의 그리드 셀을 생성하기 위해서 이진 분할을 할 경우 필요한 차원의 수는 $\lceil \log_2 P\rceil$이며 차원이 높아지면 이진 분할을 가정하더라도 분할이 발생하지 않는 차원이 존재하게 된다. 이진 분할의 경우 '차원의 저주(Curse of Dimensionality)' 현상 때문에 극히 작은 선택률의 영역 질의에 대해서도 모든 그리드 셀을 방문해야 하는 단점이 있다. 이를 해결하기 위해서 본 논문에서는 이진 분할에 필요한 차원의 수 보다 적은 수의 차원을 선택해서 여러 번 분할하도록 한다. 이 때, 여러 형태의 분할 방법이 존재하며 디클러스터링 알고리즘의 성능 향상을 위해서는 영역 질의와 겹치는 그리드 셀의 수를 최소로 하는 분할 방법을 선택해야 한다. 그리드 분할에 사용되는 차원의 수를 d_p라고 할 때, 선택된 차원의 분할 횟수는 $\lceil \sqrt[d_p]{P}\rceil$이 되며, 나머지 차원의 분할 횟수는 1이 된다. 각 차원의 분할 횟수가 결정되면, 앞 절에서 제시한 그리드 분할의 성능 예측 모델을 적용하여 영역 질의와 겹치는 그리드 셀의 수를 계산하여 최소로 하는 분할 방법을 선택한다.

Algorithm GRID-Declustering
입력 d: 차원, N: 데이터 수, B: 디스크 블로킹 인수,
M: 디스크 수, q: 질의창의 한 변의 길이,
F: 매핑 함수)
1: $P\leftarrow\lceil N/B\rceil$ /* 필요한 그리드 셀의 수 계산 */
2: d_p /* 그리드 분할에 사용되는 차원의 수 ($\leq d$) */
3: **For** each d_p { /* $1\leq d_p\leq \min(d, \lceil \log_2 P\rceil)$ */
4: $\lambda_i\leftarrow\lceil \sqrt[d_p]{P}\rceil$ *where* $1\leq i\leq d_p$
5: $\lambda_j\leftarrow 1$ *where* $d_p\leq j\leq d$ /* 나머지 차원은 분할에 사용되지 않음 */
6: $E(Q)\leftarrow$ ***Expected***$NQ(d,\lambda_i's,q)$/*각 차원의 분할 정보를 이용하여 질의창과 겹치는 셀의 수 예측*/
7: Choose $\lambda_i's$ such that minimize B_Q
8: }
9: Allocate a disk number to each cell by $F(I[d],M)$
/* $I[j]$는 그리드 셀의 j번째 차원의 구간 번호*/

(그림 6) 효율적인 그리드 분할에 의한 디클러스터링 알고리즘

위와 같은 방법으로 분할 형태가 결정되면, 이전에 제시된 다양한 매핑 함수들을 적용할 수 있다[1-8, 12]. 본 논문에서는 이 중에서 [3]에서 제시된 다차원 공간에 대해서 가장 좋은 성능을 보이는 것으로 소개되고 있는 Kronecker Sequence에 기초한 매핑 함수를 적용 하였다. (그림 6)은 본 논문에서 제시한 효율적인 그리드 분할을 통한 매핑 함수를 이용하는 디클러스터링 알고리즘을 기술한 것이다.

5. 실험 결과

5장에서는 실험 결과를 보이도록 한다. 먼저, 본 논문에서 제시한 예측 모델의 정확성을 다양한 실험을 통하여 검증하고 실제로 분할 방법이 매핑 함수를 이용하는 디클러스터링 알고리즘의 성능에 미치는 영향을 보이도록 한다.

5.1 분석 모델의 검증

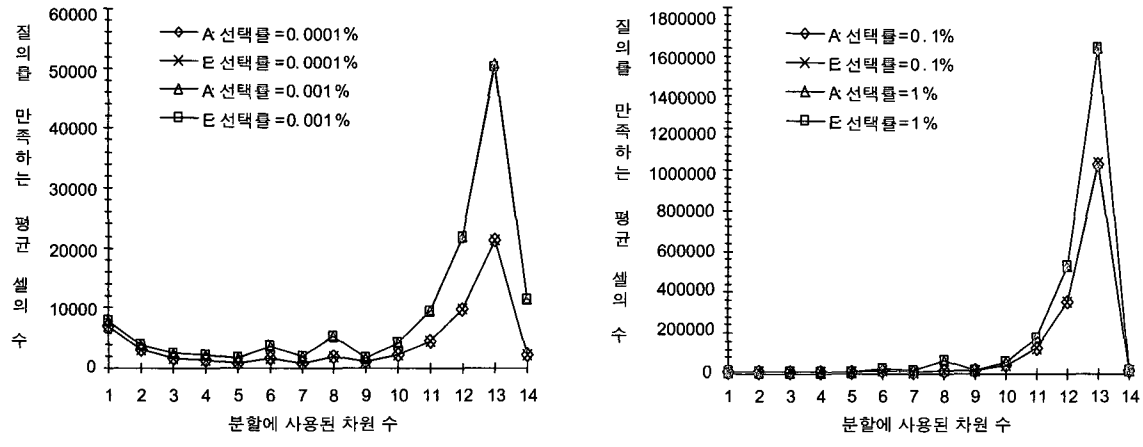
본 절에서는 제시한 예측 모델의 정확성을 실험을 통하여 보이도록 한다. 이를 위하여 생성되는 그리드 셀의 수가 동일한 여러 형태의 분할 방법에 대해서 다양한 크기의 영역 질의를 생성하여 실제로 겹치는 그리드 셀의 수와 예측 모델에 의한 결과를 비교하도록 한다.

이를 위하여 20차원 데이터에 대해서 동일한 수의 그리드 셀을 생성하는 서로 다른 분할 방법에 대해서 영역 질의에 대해서 실제로 겹치는 그리드 셀의 수와 본 논문에서 제시한 예측 모델에 의해서 계산한 결과를 제시하도록 한다. 2^{20}개의 데이터 블록(그리드 셀)을 생성하도록 분할하는 방법은 $4^{10}\times1^{10}$, $16^{5}\times1^{15}$, $32^{4}\times1^{10}$, $1024^{2}\times1^{18}$의 4가지 경우가 있다. 즉, 전체 20개의 차원 중 그리드 분할을 위해서 각각의 경우에 10, 5, 4, 2개의 차원을 사용하며 나머지 차원에 대해서는 분할이 발생하지 않는다. 질의 크기가 예측 모델에 미치는 영향을 분석하기 위하여 $[0, 1]^{20}$에 존재하는 다양한 선택률(selectivity)에 해당하는 10,000개의 정방형 질의를 생성하여 실제로 겹치는 그리드 셀 수의 평균값과 4장에서 제시한 예측 모델에 의한 예측 결과와 비교 하였다. <표 2>는 실험 결과를 나타내며 대부분의 경우 0.1% 이내의 에러율을 보이며 본 논문에서 수행한 모든 실험 결과에 대해서 0.5% 이내의 정확성을 나타냄을 알 수 있다.

(그림 7)은 16차원 10^{6}개의 데이터에 대해서 디스크 페이지 크기가 4KB일 때 가능한 분할 방법에 따른 성능 분석을 실제 결과와 본 논문에서 제시한 예측 모델에 의한 결과를 비교한 것이다.

〈표 2〉 분할 방법에 따른 질의를 만족하는 평균 그리드 셀의 수와 분석 모델에 의한 예측 결과

분할 형태	선택률 : 0.01%		선택률 : 1%		선택률 : 10%	
	실 제	예 측	실 제	예 측	실 제	예 측
2^{20}	1,048,576.000	1,048,576.000	1,048,576.000	1,048,576.000	1,048,576.000	1,048,576.000
$4^{10}\times1^{10}$	179,828.234	180,609.273	1,048,576.000	1,048,576.000	1,048,576.000	1,048,576.000
$16^{5}\times1^{15}$	218,824.385	218,738.123	625,454.272	626,101.769	1,075,106.099	1,075,662.848
$32^{4}\times1^{16}$	201,246.655	201,142.794	484,565.956	484,395.101	751,697.686	752,040.529
$1024^{2}\times1^{18}$	418,733.490	418,738.516	663,242.883	663,232.669	834,739.607	834,736.029



(a) $q < 0.5$ (b) $q \geq 0.5$

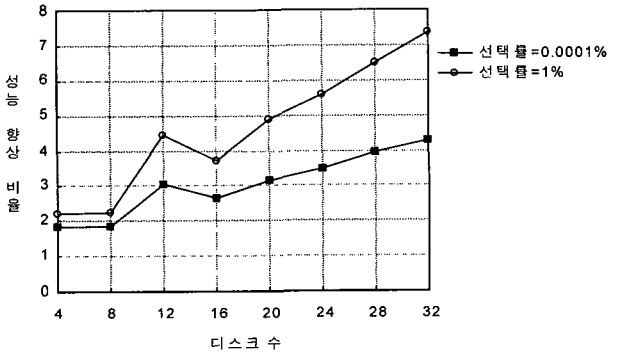
(그림 7) 분할 방법에 따른 질의 결과 크기와 분석 모델에 의한 비교

그림에서 문자 'A'와 'E'는 각각 실제 결과와 예측 결과를 나타낸다. 16차원 데이터에 대해서 이진 분할(binary partition)을 가정하더라도 14개의 차원이 필요하며 따라서 14개의 분할 방법이 존재한다.

(그림 7)에서 알 수 있듯이 본 논문에서 제시한 예측 모델은 실제 결과와 거의 동일한 결과를 보여 주며 우리는 이를 이용하여 질의 크기가 주어졌을 때 가능한 분할 방법들 중에서 질의를 만족하는 그리드 셀의 수를 최소로 하는 분할 방법을 선택할 수 있다.

5.2 그리드 분할 방법의 응답 시간에 대한 영향 분석

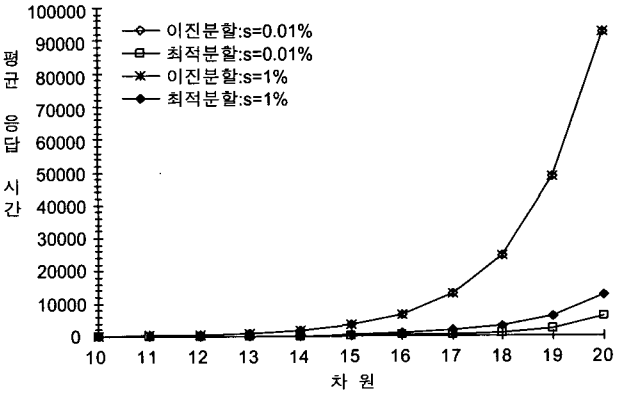
5.2절에서는 그리드 분할 방법이 구체적으로 매핑함수를 이용하는 디클러스터링 알고리즘의 응답 시간(response time)에 미치는 영향을 실험을 통하여 보이도록 한다. 본 절에서는 응답 시간을 [정의 2]에 의해서 질의 처리를 위해서 가장 많이 접근하는 디스크의 접근 횟수로 가정하였다. 다차원 그리드 분할에 대해서 적용 가능한 다양한 매핑 함수들이 제시되었으며 [3]에서 제시된 Kronecker sequence를 이용한 디스크 번호 할당 알고리즘이 가장 좋은 성능을 보이는 것으로 소개 되었다. 따라서, 본 논문에서는 다양한 그리드 분할에 대해서 Kronecker sequence 매핑 함수를 이용하여 디스크 번호를 할당할 때 응답 시간에 대한 성능 비교를 하였다. (그림 8)은 15차원 데이터에 대해서 2^{15}개의 데이터 블록(그리드 셀)이 필요할 때 이진 분할과 본 논문에서 제시한 예측 모델을 이용하여 가능한 분할 방법 가운데 질의를 만족하는 그리드 셀의 수를 최소로 하는 분할 방법을 선택했을 때 디스크 수의 변화에 따른 성능 비교 결과이다.



(그림 8) 디스크 수의 변화에 따른 Kronecker sequence 매핑 함수의 성능 변화

일반적으로 디스크 수가 증가함에 따라 이진 분할에 비해서 성능이 향상되며 (그림 8)에서 알 수 있듯이 디스크 수가 32이고 선택률이 1%일 때 최대 7.4배까지 성능 향상을 가져옴을 알 수 있다. 디스크 수가 증가할수록 성능 향상비율이 높아지는 이유는 이진 분할을 할 경우에는 적용한 매핑 함수의 특성에 따라서 일정 수준 이상에서는 디스크 수를 증가시키더라도 성능이 향상되지 않기 때문이다.

본 논문에서는 차원이 증가함에 따라 이진 분할을 할 때 성능의 변화를 나타내기 위한 실험을 하였다. 이를 위해서 10~20차원 공간에 대해서 2^d개의 데이터 블록이 필요할 때 예측 모델을 적용하여 최적의 분할을 하였을 때 성능 비교를 하였다. (그림 9)는 디스크 수가 40일 때 차원이 증가함에 따라 선택률이 10^{-5}, 10^{-2}일 때 평균 응답 시간을 나타낸 것이다.



(그림 9) 차원의 변화와 분할 방법에 따른 응답 시간에 대한 성능 비교 : 디스크 수는 40 (s : 선택률)

차원이 증가함에 따라 이진 분할을 할 경우 질의를 만족하는 그리드 셀의 수에 관계 없이 응답 시간이 급격하게 증가함을 알 수 있다. 이는 이진 분할의 경우 질의를 만족하는 그리드 셀의 수가 거의 기하급수적으로 증가하며 또한 전체 디스크를 효과적으로 활용하지 못하기 때문이다. 따라서, 지금까지 제시된 다양한 매핑 함수에 대해서 이진 분할의 경우 디스크 번호를 할당하는 알고리즘에 대한 연구가 더 요구된다. 이에 비해서 이진 분할에 필요한 차원보다 작은 수의 차원을 선택해서 여러 번 분할할 경우 차원의 증가함에 따라 응답 시간이 증가하는 정도가 매우 작음을 알 수 있다.

지금까지 우리는 다양한 실험을 통하여 본 논문에서 제시한 그리드 분할의 성능 예측 모델의 정확성을 보였다. 또한 동일한 매핑 함수를 적용할 때, (그림 9)의 18차원의 경우에서 알 수 있듯이 그리드 분할 방법의 개선만으로도 디클러스터링 알고리즘의 성능을 최대 23배 이상 향상시킬 수 있음을 알 수 있다.

6. 결론 및 향후 연구 과제

지금까지 데이터 분산을 통한 입출력 성능의 향상을 위한 디클러스터링 알고리즘 연구가 많이 행해졌다. 대부분의 이전 연구들은 데이터 공간이 그리드 형태로 분할되어 있다는 가정하에 각각의 그리드 셀에 대해서 부분 만족 질의(partial match query) 또는 영역 질의(range query)에 대해서 효과적으로 디스크 번호를 할당하는 알고리즘 개발에 집중되었다.

그들은 분할 방법이 디클러스터링 알고리즘 성능에 주는 영향은 간과한 채 효율적인 디스크 할당 정책 연구에 집중하

였다. 본 논문에서는 지금까지 제시 되었던 디클러스터링 알고리즘 연구의 대부분을 차지하는 매핑 함수를 이용하는 디클러스터링 알고리즘의 성능 향상을 위해서 지금까지 무시되었던 데이터 분할 측면에서 해결하고자 하였다. 본 논문의 의의는 다음과 같다.

- 일반적으로 고차원 데이터에 대해서 그리드 형태로 분할할 때 가정하는 이진 분할(binary partition)의 문제점을 제시하였다. 질의창의 한 변의 길이가 도메인 길이의 1/2보다 클 경우 모든 그리드 셀을 방문해야 한다. 이 때에는 지금까지 제시되었던 여러 매핑 함수가 의미가 없어진다. 즉, 순차적으로 디스크 번호를 할당하는 것이 가장 효과적이다. 다차원 데이터에 대해서는 극히 작은 선택률(selectivity)에 대해서도 질의 창의 길이는 1/2보다 커진다.
- 본 논문에서는 다차원 데이터에 대해서 이진 분할이 가지는 문제점은 분할에 사용되는 차원의 수를 줄임으로써 해결하고자 하였다. 즉, 이진 분할에 필요한 차원의 수보다 작은 수의 차원을 선택해서 여러 번 분할 하는 것이다. 실험 결과 차원 감소 방법을 적용할 경우 이진 분할에 비해서 방문해야 하는 그리드 셀의 수를 현저히 줄일 수 있음을 알 수 있었다.
- 차원 감소를 할 수 있는 방법은 여러 가지가 있으며 가능한 방법 가운데 최선의 방법을 선택하는 것이 중요하다. 이를 위하여 본 논문에서는 주어진 그리드 분할에 대해서 영역 질의를 만족하는 그리드 셀의 수를 계산하는 예측 모델을 제시하였다. 여러 차원과 다양한 크기의 영역 질의에 대한 실험 결과 제시한 예측 모델은 질의 크기와 차원에 관계 없이 0.5% 이내의 에러율을 보이는 것으로 나타났다.
- 다차원 데이터에 대해서 가장 좋은 성능을 보이는 Kronecker Sequence 매핑 함수[3]를 적용할 때 본 논문에서 제시한 예측 모델을 이용하여 적용 가능한 여러 분할 방법 가운데 질의를 만족하는 셀의 수를 최소로 하는 방법을 적용할 때 이진 분할에 비해서 최대 23배까지 성능의 향상을 가져온다.

본 논문에서는 다차원 데이터에 대해서 효과적인 그리드 분할을 수행하는 알고리즘을 제시하였다. 하지만, 좋은 분할 방법을 적용하더라도 다차원 데이터에 대해서는 차원의 저주(Curse of Dimensionality) 현상 때문에 그리드 형태로 분할하는 것은 필연적인 약점을 가지고 있다. 따라서, 다차원 데이터에 대해서는 새로운 분할 방법이 필요하다. 향후 연구 과제는 다차원 공간에 대한 그리드 분할을 대체할 새로운 분할 방법과 그에 상응하는 디스크 할당 알고리즘을 개발하는 것이다.

참 고 문 헌

[1] M. J. Atallah and S. Prabhakar, (Almost) Optimal Parallel Block Access for Range Queries, *In Proc. PODS Conf*, pp. 205-215, 2000.

[2] R. Bhatia, R. K. Sinha and C.-M. Chen, Declustering Using Golden Ratio Sequences, *In Proc. ICDE Conf*, pp.271-280, 2000.

[3] C-M. Chen, R. Bhatia and R. K. Sinha, Multidimensional Declustering Schemes Using Golden Ratio and Kronecker Sequences, *IEEE TKDE*, Vol.15, No.3, pp.659-670, 2003.

[4] C. M. Chen and C. T. Cheng, From Discrepancy to Declustering : Near optimal multidimensional declustering strategies for range queries, *In Proc PODS Conf*, pp.29-38, 2002.

[5] H. C. Du and J. S. Sobolewski, Disk Allocation for Cartisian Files on Multiple-Disk Systems, *ACM Trans. Database Systems*, Vol.7, No.1, pp.82-102, 1982.

[6] C. Faloutsos and P. Bhagwat, Declustering Using Fractals, *In Proc. Parallel and Distributed Information Systems Conf*, pp.18-25, 1993.

[7] C. Faloutsos and D. Metaxas, Disk Allocation Methods Using Error Correcting Codes, *IEEE Trans on Computers*, Vol.40, No.8, pp.907-914, 1991.

[8] M. H. Kim and S. Pramanik, Optimal File Distribution For Partial Match Retrieval, *In Proc. SIGMOD Conf*, pp. 173-182, 1988.

[9] T-W. Kim, A Distance-Based Packing Method for High Dimensional Data, *PhD thesis*, Pusan National University, 2003.

[10] S-W. Kuo, M. Winslett, Y. Cho and J. Lee, New GDM-based Declustering Methods for Parallel Range Queries, *In Proc. IDEAS Symp*, pp.119-127, 1999

[11] D. R. Liu and S. Shekhar, Partitioning Similarity Graphs : A Framework for Declustering Problems : *International Journal Information Systems*, Vol.21, No.6, pp.475-496, 1996.

[12] S. Prabhakar, K. Abdel-Ghaffar and A. El Abbadi, Cyclic Allocation of Two-Dimensional Data, *In Proc. ICDE Conf*, pp.94-101, 1998.

[13] Y. Zhou, S. Shekhar and M. Coyle, Disk Allocation Methods for Parallelizing Grid Files, *In Proc. ICDE Conf*, pp. 243-252, 1994.

[14] S. Berchtold, C. Bŏhm, B. Braunmuller, D. A. Keim and H.-P. Kriegel, Fast Parallel Similarity Search in Multimedia Databases, *In Proc. SIGMOD Conf*, pp.1-12, 1997.

[15] K. Abdel-Ghaffar and A. E. Abbadi, Optimal Allocation of Two-Dimensional Data, *In Proc ICDT Conf*, pp.409-418, 1997.

[16] Y-L. Lo, K. A. Hua and H. C. Young. GeMDA : A Multidimensional Data Partitioning Technique for Multiprocessor Database Systems. *Distributed and Parallel Databases*, Vol.9, No.3, pp.211-236, 2001.

[17] S. Prabhakar, D. Agrawal and A. E. Abbadi, Disk Allocation for Fast Range and Nearest-Neighbor Queries, *Distributed*

and Parallel Databases, Vol.14, No.2, pp.107-135, 2003.

[18] C-.M Chen and R. K. Sinha, Analysis and Comparison of Declustering Schemes for Interactive Navigation Queries, *IEEE TKDE*, Vol.12, No.5, pp.763-778, 2000.

[19] M. T. Fang, R. C. T. Lee and C. C. Chang, The Idea of De-Clustering and Its Applications, *In Proc VLDB Conf*, pp.181-188, 1986.

[20] Kamel and C. Faloutsos, Parallel R-trees, In Proc SIGMOD, pp.195-204, 1992

[21] B. Chor, C. E. Leiserson, R. L. Rivest and J. B. Shearer, An Application of Number Theory to the Organization of Raster-Graphics Memory, Journal of ACM, Vol.33, No.1, pp.86-104, 1986.

[22] L. T. Chen and D. Rotem, Declustering Objects for Visualization, *In Proc VLDB Conf*, pp.85-96, 1993.

[23] C. Chang, B. Moon, A. Acharya and C. Shock, Titan : a High-Performance Remote-sensing Database, *In Proc ICDE Conf*, pp.375-384, 1997.

[24] R. Bhatia, R. K. Sinha and C-M. Chen, Hierachical Declustering Schemes for Range Queries, *In Proc EDBT Conf*, pp.525-537, 2000.

[25] D-R. Liu and M-Y. Wu, A Hypergraph Based Approach to Declustering Problems, *Distributed and Parallel Databases*, Vol.10, No.3, pp.269-288, 2001.

[26] T-W. Kim, H-C. Kim and K-J. Li, Analyzing the range query performance of two partitioning methods in high-dimensional space, *Technical Report, Department of Computer* Science, Pusan National University, 2003. http://isel.cs. pusan.ac.kr/paper/pdf/twkim_03_IPL.pdf.

[27] S. Berchtold, C. Bohm and H-.P. Kriegel, Improving the Query Performance of High-Dimensional Index Structures by Bulk Loading R-trees, *In Proc EDBT Conf*, pp.216-230, 1998.

김 학 철

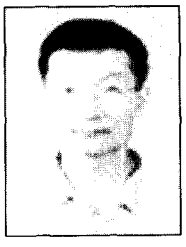
e-mail : hkckim@pnu.edu

1997년 부산대학교 전자계산학과(학사)

1999년 부산대학교 대학원 전자계산학과 (석사)

2002년 부산대학교 대학원 전자계산학과 박사과정 수료

관심분야 : 병렬 데이터베이스, 다차원 색인, GIS, 시공간 데이터베이스



김 태 완

e-mail : twkim@quantos.cs.pusan.ac.kr

1988년 연세대학교 경역학과(학사)

1991년 미국 피츠버그대학교 전자계산학과 (학사)

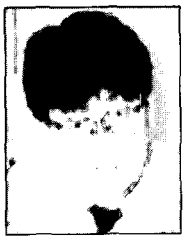
1994년 미국 텍사스 A&M 대학교 대학원 전자계산학과(석사)

2003년 부산대학교 대학원 전자계산학과(박사)

2003년~2004년 부산대학교 컴퓨터 및 정보통신 연구소 전임연구원

2004년~현재 행정자치부 전자정부전략개발실 전문위원

관심분야 : 다차원 색인, GIS, 시공간 데이터베이스



이 기 준

e-mail : lik@pusan.ac.kr

1984년 서울대학교 계산통계학과(학사)

1986년 서울대학교 대학원 계산통계학과(석사)

1992년 프랑스 국립응용과학원(INSA) 전자계산학과(박사)

1990년~1991년 프랑스 Logicim사 선임 엔지니어

1993년~현재 부산대학교 정보컴퓨터공학부 부교수

관심분야 : 시공간데이터베이스, 텔레메틱스, 유비쿼터스 컴퓨팅