

관성을 이용한 이동체 데이터베이스의 디클러스터링

서영덕* 김진덕** 홍봉희*

*부산대학교 컴퓨터공학과, **동의대학교 컴퓨터공학과

ydseo@pusan.ac.kr, jdk@dongeui.ac.kr, bhhong@pusan.ac.kr

Declustering of Moving object database based on Inertia

Youngduk Seo*, JinDeog Kim**, BongHee Hong*,

*Dept. of Computer Eng. Pusan Nat'l Univ., **Dept of Computer Eng. DongEui Univ.

요 약

이동체의 궤적을 저장하는 대용량 이동체 DB는 대규모의 이동 객체 궤적의 효과적인 검색을 위하여 디클러스터링 기법을 통한 객체 궤적의 분산 배치가 필수적으로 요구된다. 그러나 기존 공간 객체의 디클러스터링 기법은 이동체의 특성과 시간 영역에 대한 고려 없이 디클러스터링을 수행한다. 또한, 단순히 현재 시점에서 색인 노드의 공간 관련성만을 판단의 근거로 삼고 있어서 효과적인 디클러스터링이 되지 않는 단점이 있다. 이러한 이유로 이동체 데이터베이스에서 빠른 질의 수행을 위한 디클러스터링 기법이 필요하다.

이 논문에서는 이동체 궤적에 대한 질의 시 빠른 응답 시간을 얻고 전체 시스템의 처리를 향상을 위한 디클러스터링 방법을 제시한다. 제시되는 방법은 이동체의 진행 방향에 대하여 이동 시간에 의한 이동 궤적의 관성을 정의하고, 이를 색인의 노드 단위로 확장한 노드의 관성을 정의한다. 정의된 관성을 이용하여 이동체 궤적의 노드가 저장될 디스크를 정의함으로써 궤적 데이터의 디클러스터링을 효과적으로 수행할 수 있다.

1. 서론

디클러스터링은 무선 통신과 GPS(Global Positioning System)의 보급이 늘어남에 따라 위치 기반 서비스, 물류관제 서비스 등의 이동체와 관련된 다양한 서비스에서 효과적인 데이터 처리를 위하여 많은 필요성이 제기되고 있다. 그러나 기존의 공간 객체를 위한 디클러스터링 기법은 이동체의 특징인 시간 도메인에서의 이동체의 이동에 대한 고려가 없고, 이동체의 특성이 고려되지 못하고 있다. 또한, 단순히 현재 시점에서의 이동체의 궤적만을 판단의 근거로 삼고 있어서 정확한 디클러스터링을 위한 지표가 생성되지 못하는 단점이 있다. 그러므로 위와 같은 문제점을 해결 하기 위한 새로운 이동체 디클러스터링 방법의 제시가 필요하다.

이 논문에서는 이동체에 대한 궤적 질의와 영역 질의 시 빠른 응답 시간을 얻고 전체 시스템의 처리를 향상을 위하여 이동체의 관성을 이용한 디클러스터링 방법을 제시한다. 즉, 이동체의 진행 방향에 대한 관성을 통하여 다음 시점에서의 이동체의 진행 방향에 대한 판단하고, 이를 통하여 이동체 궤적의 노드가 저장될 디스크를 정의한다. 이를 통하여 이동 객체가 가지는 특성을 고려한 디클러스터링 기법을 생성할 수 있다.

이 논문의 구성은 다음과 같다. 먼저 2장에서는 기존의 디클러스터링 기법과 이동체 데이터베이스와 관련된 연구를 소개한다. 3장에서는 이동체의 관성에 관하여 소개하고 있으며 5장에서는 관성을 이용한 디클러스터링 기법을 제시한다. 마지막으로 6장에서 결론과 향후 연구 과제에 대하여 정리한다.

2. 관련 연구

기존의 연구에서 디클러스터링은 공간 색인에서 하나의 질의에 동시에 검색될 것이 예측되는(즉, 근접성이 높은) 색인의 노드 혹은 데이터 페이지들을 서로 다른 디스크에 둬으로써, 영역 질의 시 넓은 영역에 대한 병렬성을 높이는 방법이 제시되었다. 이러한 이유로 근접성은 디클러스터링에서 노드를 분산 배치하기 위한 기준으로 활용된다.

2차원 공간상의 공간 데이터베이스에 관한 기존의 연구는 [1]을 들 수 있다. [1]에서는 공간 객체의 근접성을 제안하고, 제안된 근접성을 이용한 parallel R-Tree를 설계하였다. parallel R-Tree는 노드의 근접성에 따라 가장 근접성이 떨어지는 디스크에 배치함으로써, 공간 질의의 minLoad와 높은 처리율을 유지할 수 있도록 제시되었다.

또한, [2]에서는 일 차원 데이터베이스에서의 기존에 알려진 다양한 방법(record distribution, large page B-tree, page distribution)에 관한 다양한 성능 평가를 수행하였으며, 일 차원 데이터베이스를 위한 색인 디클러스터링 기법에 관한 연구를 수행하였다.

시간 데이터베이스 상에서의 디클러스터링은 [3]을 들 수 있다. [3]에서는 TSB-Tree를 이용하여 디클러스터링을 수행하였으며, 시간 영역 질의와 속성 영역 질의에 대하여 빠른 질의 수행을 위하여 비슷한 부류의 데이터를 서로 다른 디스크에 적재하는 기법을 제시하였다

이와 같이 기존의 연구에서의 디클러스터링은 다음과 같은 큰 3가지의 기준을 이용하여 수행하는 것이 제시되었다.

- (1) 공간 근접성: 두 개의 공간 객체가 있을 때, 두 객체 간에 겹치는 면적에 대한 근접성을 연산하여 디클러스터링을 수행
- (2) 시간 근접성: 시간 축 + 공간 축에 대하여 시공간의 입체의 용적의 겹침을 이용하여 근접성을 정의하여 디클러스터링을 수행.
- (3) 기타 근접성을 이용하지 않는 방법: Round Robin, Minimum Area, Disk Modular등의 방법이 있다.

3. 이동체의 관성

이 장에서는 이 논문의 기본 아이디어인 이동체의 관성을 정의한다. 또한 이를 색인의 노드 관점으로 확장하여 이동체의 이동에 따른 색인 노드의 진행 방향을 예측할 수 있는 노드의 관성을 정의한다.

3.1 이동체의 관성

이동체는 일정한 속도와 진행하고자 하는 방향을 가지는 특성을 가진다.[4] 또한 이동체의 궤적은 이동체 데이터베이스에서 이동체가 보고한 하나의 간격에서의 이동 경로를 의미하는 세그먼트의 집합이다. 그러므로, 이동체의 방향과 속도가 일정하고자 하는 성질에 의해 이동체의 궤적 역시 일정한 속도와 방향으로 진행하고자 하는 관성을 가지게 되며, 이를 다음과 같이 정의한다.

정의 1) 관성의 영향 범위(P)(Influence scope of Inertia):

궤적 segment s_i 와 s_{i+1} 의 속도가 반전되는 지점 즉, 각 축의 속도가 $+ \rightarrow -$ 가 되거나 $- \rightarrow +$ 로 변하는 i 로부터 현재 지점 관성의 영향 범위는 짧은 시간 간격에서 정지, 역방향 진행 등의 순간적인 돌출 행동을 무시하고 색인이나 데이터 페이지의 단위를 넘지 않는 범위 내의 이동체의 진행 방향과 속도에 의해 다음 방향과 속도를 예측하고자 함이다.

정의 2) 궤적의 관성: 데이터베이스 내의 임의의 이동체 A의 궤적 $trjA$ 가 존재하고, 이의 세그먼트S가 있을 때, 시점 n에서 이동체의 궤적 세그먼트 S_n 는 $s_{i'}$ 로부터 $s_{i''}$ 까지의 방향과 속도를 따르고자 하는 성질을 가진다. 여기서 P는 관성 영향 범위이다.

이 논문에서는 궤적의 관성을 궤적에 대한 색인의 노드에 대하여 적용한다. 색인의 노드는 색인의 종류와 삽입 정책에 따라 동일한 구성요소를 가지고 있더라도, 색인을 형성하는 궤적집합은 달라질 수 있다. 그러나, 색인의 노드는 색인의 식별력(discrimination)을 높이기 위한 삽입 정책에 의하여 색인의 동일 노드내의 궤적은 다른 노드내의 궤적들에 비하여 유사한 성격을 가지게 된다.

궤적을 포함하는 노드에서 특정 축에 대한 관성은 포함된 이동체의 궤적들 중 그 축에 대하여 가장 빠른 객체의 관성 방향으로 확장/축소된다. 그러나, 가장 빠른 객체가 진행 방향으로 멀리 있어서 다음 노드를 가득 채우는 시점까지 해당 노드의 방향을 결정하지 못할 경우도 존재한다. 이러한 이유로 궤적의 확장 방향에 대하여 결정적인 영향을 미칠 수 있는 이동체의 궤적을 관성 결정 궤적(trjD)이라 정의한다.

관성 결정 궤적은 해당 노드가 생성되는 시점에 정의되며, 새로이 생성된 노드의 최종 진행 방향과 속도를 결정한다. 주어진 노드N에 관성 결정 범위 P내의 궤적 $trjA_0, trjA_1, \dots, trjA_n, trjB_0, trjB_1, \dots, trjB_m, \dots, trjX_0, \dots, trjX_x$ 가 있을 때, 각 객체의 궤적에 대한 관성은 $\{a_i, b_i, \dots, x_i\}$ 가 된다. 또한, 이 노드가 가득 차는데 걸린 시간을 T일때, 다음 현재 관성에 따라 이동체가 이동할 때, 이동체의 위치는 $\{trjA_n + 2T_i * a_i, trjB_m + 2T_i * b_i, \dots, trjX_x + 2T_i * x_i\}$ 가 된다. 여기서 관성 결정 궤적 $TrjD = \text{Max}\{trjA_n + 2T_i * a_i, trjB_m + 2T_i * b_i, \dots, trjX_x + 2T_i * x_i\}$ 로 정의된다. 궤적 포함 노드의 관성은 이동체 궤적의 관성과 달리 각 축에 대하여 양의 방향에 대한 $TrjD^+$ 와 음의 방향 $TrjD^-$ 의 2개의 값을 가진다.

궤적을 포함하는 노드는 관성 결정 궤적에 의하여 해당 노드의 관성이 결정된다. 이것은 노드의 MBB는 포함하는 객체

집합 중 가장 변두리(Max)의 객체를 포함하는 근접 사각형이기 때문이다. 즉, 이 논문에서 궤적 포함 노드의 관성은 다음과 같다.

정의3) 궤적 포함 노드의 관성: 이동체 궤적을 포함하는 색인 노드에서 관성 결정 궤적의 관성

문제를 단순화하기 위하여 1차원 공간상에서의 이동체의 이동으로 가정한다.

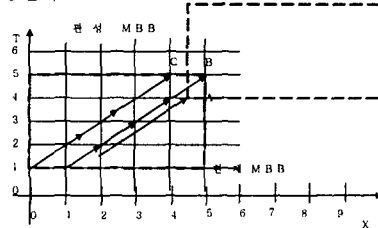


그림 1) X축의 양의 방향에 대한 이동체의 이동

예를 들어, 노드의 최대 포함 객체수가 8인 색인일 때, 그림 1에서 X축에 대한 $TrjD^+$ 는 다음과 같다.

$$MBB(X) = (0,5)$$

$$MBB(T) = (1,5)$$

$$IMBB(X) = MBB + \text{Speed}(trjD) * \text{Time_Of_Node_Full} = 5 + 16/3$$

$$IMBB(T) = 5 + 4$$

즉, 현재 노드내의 객체들이 현재 시점의 속도와 방을 기준으로 진행하는 경우 다음 노드가 가득차는 시점 IMBBT에서의 X축은 (5.31/3)의 위치함을 예측할 수 있다.

2차원 상에서의 관성은 1차원과 독립적인 연산으로 볼 수 있으며, T와 관계된 연산이므로 별개의 연산으로 처리 가능하다.

3.2 이동체 색인 노드의 관성

색인 노드 중 비단말노드의 관성 역시 궤적 포함 노드의 관성과 비슷하게 볼 수 있다. 즉, 포함하는 하위 노드의 MBB 중 관성 결정 노드(NodeD)로 볼 수 있는 노드가 존재하며, 현재 노드의 관성은 이 관성 결정 노드에 의하여 결정된다.

4. 이동체의 관성에 의한 디클러스터링

이 장에서는 이동체 데이터베이스 시스템에서 기존 연구의 디클러스터링 정책의 문제점을 제시하고, 3장에서 기술한 이동체의 관성을 이용하여 디클러스터링을 수행하는 방법을 제시한다. 궤적 포함 노드와 색인 노드를 이용한 디클러스터링 기법을 제시하며, 관성에 의한 근접성 생성 방법을 제시한다.

4.1 기존 연구의 문제점

기존의 디클러스터링 기법은 색인의 노드가 생성되는 시점의 공간 관련성을 이용하여 근접성에 대한 판단을 수행한다. 그러므로, 공간 객체와 같이 정적이며, 삽입과 삭제의 방향이 예측 불가능한 데이터 집합에 대하여는 적절하나, 관성을 통한 효과적인 색인의 진행방향이 예측되는 집합에 대하여는 최적의 성능을 발휘하기 어렵다.

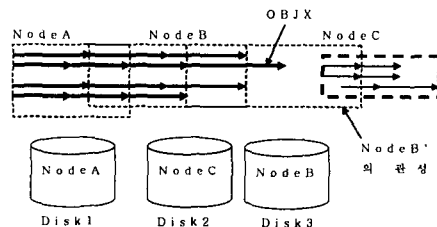


그림 2) 이동체의 관성에 의한 디스크 배치

예를 들어, 그림 2)의 궤적을 TB-Tree로 색인 할 때,

NodeB에서 분할되는 NodeB'은 기존의 방법에 의하면 공간 근접성이 가장 적은 NodeC가 Disk2에 적재될 것이다. 그러나, 이러한 방법은 NodeB'의 관성결정 객체인 ObjX의 관성에 의해 Node E'이 full이 되는 시점에 NodeC와 겹치게 될 가능성이 크다.

이 논문에서는 기존 연구의 이러한 문제점을 극복하고, 보다 효과적인 질의의 병렬화를 이루기 위하여 관성을 이용한 근접성을 제시한다.

4.2 객적 포함 노드에서의 관성 근접성

객적 포함 노드의 관성 근접성은 노드 내에서 관성 결정 객적을 찾는 방법과 이 관성 결정 노드를 이용하여 색인의 관성을 결정하기 위한 부분으로 나눌 수 있다. 관성 결정객적은 3.1절의 관성 결정객적의 정의에 의하여 객적을 포함하는 노드내의 객적집합 중 현재 노드가 가득 차는 기간을 다음 노드에서 동일하게 적용할 때, 생성되는 노드의 MBB를 결정할 수 있는 객적으로 결정될 수 있다. 이를 토대로 생성되는 노드의 관성에 의한 근접성은 알고리즘 1)에 의하여 계산할 수 있다.

```

Function TrijProximity(Node node1, Node node2){
    Proximity Prox;
    Double CriticalTrij1, CriticalTrij2;
    CriticalTrij1 = FindCriticalTrajectory(node1);
    CriticalTrij2 = FindCriticalTrajectory(node2);
    Prox = CalcProximity(CriticalTrij1, CriticalTrij2);
    Return Prox;
}
    
```

알고리즘 1) 노드 근접성 계산

알고리즘 1)에서 FindCriticalTrajectory는 주어진 노드에서 관성 결정 객적을 찾아내기 위한 함수이며, 3.1절의 관성에 관한 정의를 이용한다. 또한, CalcProximity는 주어진 두 개의 객적으로부터 근접성을 계산하기 위한 함수이다. 두 객체의 근접성은 기존의 연구[1]에서 작성된 방식을 이용할 수도 있다.

4.3 색인 노드의 관성 디클러스터링

이 논문에서는 기존 노드의 관성을 이용하여 새롭게 생성된 노드의 근접성을 결정하는 것을 제시한다. 그림 3에서 새로이 생성되는 객적 노드 G는 기존의 색인 노드 D의 관성이 IMBB에 의한 근접성이 적용되어야 한다.

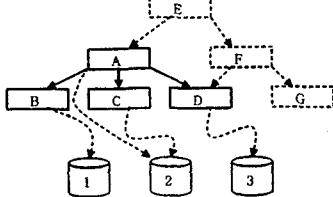


그림 3) 객적 노드의 분할과 디스크 할당

색인 노드의 디클러스터링은 노드의 관성에 의한 MBB에 의해 생성된 인접을 기준으로 적용된다. 즉, 그림 3의 노드 A, B, C, D가 그림 2의 색인 노드의 관성을 그대로 가진다고 가정할 경우 새로이 생성되는 노드G는 D의 관성을 통한 IMBB를 가질 것이다. 또한 F는 D와 G의 관성에 의한 노드 관성을 가지게 된다. 이 관성에 의한 진행 방향상의 IMBB는 기존의 근접성의 연산 방법과 다른 결론을 가지게 된다.

이 논문에서는 이동체 색인 노드의 디스크 적재 위치를 정할 때, 다음과 같이 노드 적재시 관성에 의한 디스크 선택 방법을 제시한다.

임의의 주어진 노드 N_i 가 새로 발생하는 노드일 때, N_i 의 관성 $I(N_i)$ 와 형제 노드들의 관성과는 각각 서로 다른 근접성을 가지며 다음과 같은 집합이 된다.

$$P(I(N_i)) = \{p_i | P(I(N_i), I(N_k)), 1 \leq k < n\}$$

이때, N_i 가 적재될 디스크 $D(N)$ 은 $P(I(N_i))$ 중에서 가장 낮은

근접성을 가진 sibling node가 적재된 디스크로 결정된다. 즉, 다음과 같다.

$$D(N) = \{x | P(I(N_i)) = \min\}$$

이를 이용한 노드 분할시의 디스크 선택을 위한 방법은 알고리즘 2)와 같다.

```

Function FindDisk(Node node){
    Proximity MinProx;
    DiskID SelectedDisk;
    MinProx = MaxValue;
    For (snode in All SiblingNodesExceptNode(node)){
        nodeProx = TrijProximity(node, snode)
        if (nodeProx < MinProx){
            MinProx = nodeProx;
            SelectedDisk = snode;
        }
    }
    return SelectedDisk;
}
    
```

알고리즘 2) 디스크 선택을 위한 알고리즘

알고리즘 2)는 현재 분할된 노드의 형제 노드로부터 각각의 근접성을 계산하여 가장 낮은 근접성을 가지는 노드가 적재된 디스크를 찾는다. 또한 이 알고리즘에서는 적절한 디스크를 찾기 위하여 이웃 노드의 근접성을 조사하여 Proximity가 가장 낮은 디스크를 선정한다.

예를 들어 그림 2)에서 발생하는 새로운 노드인 NodeB'의 적재 위치는 알고리즘 2)에 의하면 관성에 의한 MBB의 겹침이 가장 적은 디스크인 Disk1에 적재될 것이다.

5. 결론 및 향후 연구

무선 통신과 GPS의 보급이 늘어남에 따라 위치기반 서비스, 물류관제 서비스 등 이동체와 관련된 여러 가지 새로운 서비스들이 점점 생겨나고 있다. 이 논문에서는 이러한 서비스에서 필수적으로 요청되는 이동체 객적에 대한 질의 시 빠른 응답 시간을 얻고 전체 시스템의 처리율 향상을 위한 이동체의 관성을 이용한 디클러스터링 방법을 제시하였다. 즉, 이동체 객적의 진행 방향에 대하여 관성을 이용하여, 다음 시점에서의 이동체의 진행 방향에 대한 예측을 수행하는 이동체 객적의 관성을 정의하였다. 또한 이를 이용하여 색인 노드의 관성을 정의하고, 이를 통한 이동체 데이터베이스에서의 색인의 디클러스터링 기법을 제안하였다.

향후 이 논문에서 제시된 기법을 실제 이동체 데이터베이스에 적용하여 성능평가와 비용모델에 대한 연구를 수행할 것이다. 또한 디클러스터링 시스템을 구현하여 삽입되는 이동체 자체의 정보와 지도의 topology정보를 포함한 디클러스터링 기법에 대한 연구를 수행할 것이다. 마지막으로 이동체에 적합한 근접성 설정 방법에 대하여 연구할 것이다.

6. 참고문헌

- [1] Ibrahim Kamel, Christos Faloutsos, "Parallel R-Trees" Proceedings of the 1992 ACM SIGMOD, pp. 195-204, 1992.
- [2] Bernhard Seeger and Per-Ake Larson, Multi-Disk B-trees, Proc. ACM SIGMOD p.p 138-147
- [3] Peter Muth, Achim Kraib, Gerhard Weikum, Lot:Dynamic Declustering of TSB-Tree Nodes for parallel access to Temporal Data, Extending Database Technology(EDBT) p.p.553-572
- [4] Simonas Saltenis, Christian S.Jensen Scott T.Leutenegger, Mario A. Lopez, "Indexing the positions of Continuously Moving Objects", {SIGMOD} Conference 2000