이동체 데이터의 근접성을 이용한 디클러스터링 방법

홍 은석[0], 서 영덕, 홍 봉희
부산대학교 컴퓨터공학과
{eshong, ydseo, bhhong}@pusan.ac.kr

Spatial-temporal Declustering Method Using Proximity of Moving Object Data

Eun-Seok Hong[0], Young-Duk Seo, Bong-Hee Hong
Dept. of Computer Engineering, Pusan National University

요 약

컴퓨터와 무선 통신 기술의 발달로 인하여 LBS(Location based Service)와 같은 새로운 이동체 관련 서비스가 생겨나고 있다. 이와 같은 서비스들은 이동체들이 일정 주기를 가지고 자신의 정보를 서버로 전송하는데 이는 많은 디스크 입출력을 요구하게 된다. 그러므로 이동체 데이터에 대하여 다중 디스크를 이용한 병렬 입출력이 요구되고 있다. 그러나 기존의 디클러스터링 방법은 시간 도메인을 고려하지 않거나 공간 관련성만을 고려하여 디클러스터링을 하므로, 하나의 디스크에 특정 이동체의 궤적이 집중 되는 문제점이 있다. 이 문제점은 디스크의 병목현상으로 인한 느린 응답시간과 낮은 처리율의 결과를 발생시킨다. 그러므로 이동 객체의 빠른 질의 처리를 위한 새로운 디클러스터링 기법이 필요하다.

이 논문에서는 다중 디스크 기반의 시스템에서 이동 객체에 대한 영역질의시 빠른 응답시간과 높은 처리율을 얻기 위하여 새로운 디클러스터링 기법을 제시한다. 이동체 데이터의 궤적 MBB중 공간 좌표로부터 Predefined Disk를 생성하고 PDT-Proximity를 이용하여 시간 도메인을 고려하는 방법이다. 위와 같이 이동 객체의 특성을 고려한 새로운 디클러스터링 방법으로 시스템의 성능을 향상시킬 수 있다.

1. 서론

무선 통신과 GPS(Global Positioning System)의 보급이 늘어남에 따라 위치기반 서비스, 물류관제 서비스 등 이동체와 관련된 여러가지 새로운 서비스들이 점점 생겨나고 있다. 이러한 서비스와 관련하여 각 서버 시스템으로 보내지는 이동체 데이터의 양은 방대하다. 그리고 이동체 데이터는 시간에 따라 계속 서버로 보고 되기 때문에 아주 많은 데이터의 처리가 필요하다. 그리고 이동체의 데이터에 대한 수정, 검색시 빠른 처리에 대한 요구가 많아지고 있다.

그러나 기존의 디클러스터링 방법을 이동체 데이터에 대하여 그대로 적용할 경우 시간, 공간 도메인 모두에 대한 고려가 없어 적절한 디클러스터링이 이루어지지 않는 문제점이 있다. 특히 이동 객체가 보고하는 데이터는 시간에 따라 연속적인 형태로 보고되기 때문에 시간 도메인에 대한 고려는 중요한 문제가 된다. 그러므로 위와 같은 문제점을 해결 하기 위한 새로운 이동체 디클러스터링 방법의 제시가 필요하다.

이 논문에서는 이동체에 대한 영역 질의시 빠른 응답시간을 얻고 전체 시스템의 처리율 향상을 위한 이동체 Proximity 방법을 제시한다. 즉 이 논문에서는 이동체 데이터의 공간 좌표를 이용한 공간 Proximity와 시간 도메인을 함께 고려하여 새로운 Proximity을 제안한다. 이렇게 함으로서 이동체 데이터에 대하여 시공간 도메인 모두를 고려하여 디클러스터링 할 수 있다.

이 논문의 구성은 다음과 같다. 먼저 2장에서는 관련 연구를 소개하고 3장에서는 대상 환경 및 문제 정의를 기술한다. 4장에서는 이동체 Proximity 방법에 대해서 설명하고 5장에서는 결론 및 향후 연구를 기술한다.

2. 관련연구

기존의 디클러스터링 방법을 분류하면 일반적인 관계형 데이터베이스에서 사용되는 방법과 공간 데이터베이스에서 사용되는 방법의 두가지 종류가 있다. 이 논문에서는 공간 데이터베이스를 대상으로 한다. 공간 데이터베이스에서 기존의 연구들을 보면 수학적 heuristic을 이용한 방법과 개념적 heuristic을 이용한 방법으로 나눌수 있다. 그중 수학적 heuristic을 이용한 방법은 Parallel R-Trees를 이용한 Proximity Index[1]방법과 MVAS(Multi Version Access Structure)[3] 을 이용한 KT-Proximity[2]방법이 있다. 또한 개념적 heuristic 방법으로는 Round Robin, Minimum Area, Minimum Intersection으로 나눌수 있다.

[1]에서는 새로 삽입되는 객체에 대하여 기존의 객체들과의 근접성(Proximity)을 조사하고 높은 근접성을 가지는 객체들을 서로 다른 디스크에 저장하는 방법이다. 이렇게 함으로써 디클러스터링의 기본요소인 MinLoad, UniSpread를 만족하고자 하였다. 그러나 이 방법은 2차원적인 공간 관련성만을 고려하고 있는 문제점을 안고 있다. 즉 시간 도메인에 대한 고려가 없어서 이동체 데이터를 처리하기에는 부족한 면이 있다.

[2]의 방법에서는 MVAS[3]를 사용하여 시간 도메인에 대해 기술하고 있다. 즉 Unique한 값인 K와 시간에따라 보고되는 Time과의 관련성을 보여주고 있다. 그러나 이 방법에서 T는 시

간 도메인을 고려한 것이지만 K는 공간 관련성이 없는 단순한 Key값을 표현하고 있기 때문에 시공간의 이동체 데이터에 대해서는 바로 적용하지 못하는 문제점을 안고 있다.

즉 기존의 디클러스터링 방법들은 대부분 1차원의 시간 도메인이나 2차원의 공간적인 측면만을 따로 고려한 근접성 방법을 적용하고 있어서 시공간의 이동체 데이터에 대해서는 바로 적용하지 못하는 문제점을 가지고 있다.

3. 대상 환경 및 문제 정의

이 논문은 GPS와 같은 위치 확인 장치를 가진 이동체가 주기적으로 자신의 위치 정보를 다중 디스크 기반 데이터베이스 서버로 전송하는 환경을 대상으로 하고 있다. 또한, 이동체 데이터는 이동체의 궤적으로 정의하고, 색인으로서 TB-Tree[4]를 사용한다.

이동체 데이터는 시간에 따라 연속적인 형태로 보고되기 때문에 시간 도메인에 대한 고려가 있어야 한다. 그러나 이동체 데이터를 기존의 공간 Proximity[1]를 이용하여 디클러스터링을 수행할 경우 특정 궤적의 이동체 데이터가 하나의 디스크로 집중되는 문제점이 생길 수 있다.

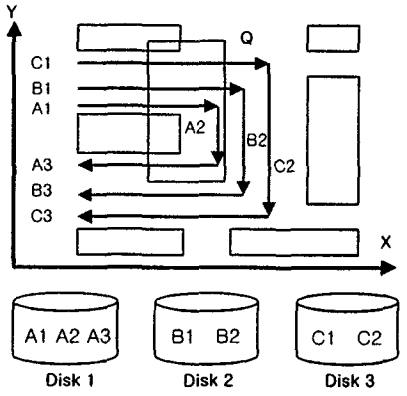

그림 1) 공간 디클러스터링

예를 들어 그림 1)에서와 같은 상황을 고려해보자. 이와 같은 상황에 대하여 관련연구 [1]의 근접성 방법을 적용할 경우를 고려해 보자. A1, B1, C1이 순서로 데이터가 입력되고 A2가 그 다음 시점에 보고될 때 A2는 B1과 C1과 Y축에 대하여 겹치는 부분이 있지만 A1과는 근접성에서 차이가 있다. 즉 A2는 공간 근접성에서 B1, C1에 비해 가장 낮은 값을 가지게 된다. 이러한 형태로 데이터가 계속 입력이 된다면 디스크 1에 하나의 궤적 노드 A가 집중되는 현상이 발생한다. 이때 Q와 같은 영역질의가 요청된다면 궤적 A로 인하여 하나의 디스크1에서 많은 I/O 가 일어 날것이다. 즉, 디스크 1에 병목현상이 일어나 전체 시스템의 늦은 응답시간과 처리율의 결과를 가지게 된다. 그러므로 시간 도메인에 대한 고려 없이 기존의 디클러스터링 방법을 그대로 적용하는 것은 문제점을 안고 있다.

4. 시공간 Proximity

3장에서 본 바와 같이 기존의 디클러스터링 방법은 시공간 도메인을 모두 고려하지 않아 이동체 데이터가 특정 한 개의 디스크에 집중되는 문제점이 발생하였다. 이 장에서는 시공간 모두를 고려한 디클러스터링 방법을 제시한다.

4.1 Predefined Disk

Predefined Disk는 공간도메인에 대한 가상디스크를 말한다. 그리고 이동 객체의 3차원 MBB 데이터 중에서 공간 좌표에 고공간 근접성을 적용하는 것을 다음과 같이 정의한다.

정의 1. Spatial Proximity (SP) : 보고되는 이동체 데이터의 공간 좌표에 대한 근접성.

기존의 두 노드 R과 S가 있다고 가정하자. 이 두 노드에 대한 Proximity는 다음과 같이 정의 된다.[1]

Proximity(R,S) = #of queries retrieving both / total #of queries

또한 2차원의 경우 영역 질의시의 Proximity(R,S) 는

$$Prox(R,S) = \frac{4}{9}\int_{-\Delta}^{\Delta}\int_{0}^{\delta}(q_1-\Delta)(q_2+\delta)dq_1dq_2$$

로 정의한다.[2]

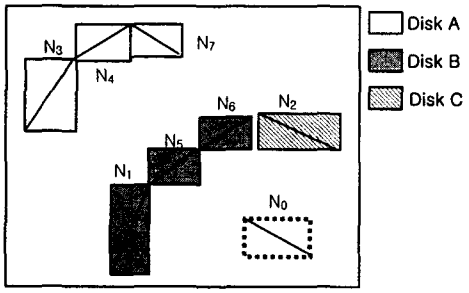

그림 2) 새로운 객체의 공간 좌표 N_0 보고시

Spatial Proximity는 특정시간에 이동객체가 자신의 위치 데이터를 보고하는 순간 적용한다.

예를 들어 TB-Tree의 한 노드의 최대 객체수를 3이라고 하고 그림 2)에서와 같이 데이터가 분포 되어 있다고 가정하자. 새로운 궤적 데이터 N_0가 삽입 되었다고 했을때 기존의 궤적과는 다르기 때문에 새로운 노드를 형성하게 된다. 그때 기존의 존재하는 노드들과 새로 보고된 N_0와 서로 공간 근접성을 적용하여 가장 근접성이 작은 디스크로 할당하는 것이다.

만약 같은 궤적의 데이터가 보고 된다면 한 노드가 다 채워질 때까지 TB-Tree에 삽입하고 노드가 다 찬 후 새로운 데이터가 보고되는 시점에 다시 SP를 적용한다.

그림 2)의 예에서 보면 N_0는 디스크 B, C보다 디스크 A에 있는 노드와의 근접성이 가장 작기 때문에 디스크 A에 할당 된다. 위와 같은 방법으로 결정되는 디스크를 다음과 같이 정의한다.

정의 2. Predefined Disk (PD) : 이동체 데이터의 MBR에 대하여 Spatial Proximity 방법 적용시 가장 낮은 Proximity값을 가지는 디스크

즉 Spatial Proximity 방법에 의하여 공간 관련성만이 고려되어 나온 디스크를 Predefined Disk라 정의 한다. 그리고 그 값을 PDT-Proximity 방법의 PD값으로 사용한다. 그림 1)에서는 Disk 1, Disk 2, Disk 3을 Predefined Disk라 할 수 있다.

Predefined Disk를 PDT-Proximity의 한 축으로 설정하는 근거는 다음과 같이 들 수 있다.

이동체 데이터 중에서 공간 차원만을 고려하여 Spatial Proximity를 적용한 값이므로, Predefined Disk라고 정의 되어 나온 디스크값은 현재 삽입되는 객체의 공간 관련성을 충분히 반영하고 있다

예를 들어 A, B, C 세개의 디스크가 있다고 가정하고 제일 마지막에 보고된 노드를 $NODE_t$ 라고 가정하면 기존의 데이터들과 공간 관련성을 비교한후 적절한 디스크를 선정하게 된다. 여기서 만약 디스크 B가 결정 되었다고 할 때 그 의미는 $NODE_t$가 기존의 디스크에 있는 객체들과의 공간 관련성에서 디스크 B와 가장 근접성이 작다는 것을 의미한다. 즉 $NODE_t$에 대하여 결정된 PD