Journal of Digital Contents Society Vol. 14 No. 3 Sep. 2013(pp. 313-321)
http://dx.doi.org/10.9728/dcs.2013.14.3.313

MapReduce 환경에서의 실시간 LBS를 위한 이동궤적 데이터 색인 및 검색 시스템 설계

정재화*

요 약

최근 모바일 스마트 기기의 보급으로 스마트 기기에 탑재된 다양한 센서에서 수집되는 대량이 데이터를 분석하여 처리하는 빅 데이터의 시대는 위치기반 서비스(LBSs: Location-Based Services)에 까지 확대대고 있다. 이동궤적에 대한 데이터도 초 대용량으로 증가하고 있다. 초 대용량 이동궤적 데이터 처리를 위해서는 클라우드 컴퓨팅 기술 및 맵리듀스와 같은 병행처리 플랫폼에 대한 연구가 필요하다. 최근 대용량 데이터의 병렬처리를 위해 맵리듀스 기반의 연구는 진행되고 있으나, 일괄처리 및 키-값 데이터 구조에 적합한 맵리듀스는 실시간 LBS에 적용에 적합하지 않다. 따라서 본 연구는 맵리듀스 특성을 면밀히 분석하고 실시간적 서비스에 적합하도록 모듈 단위로 효율적인 색인 기법 및 검색에 대한 시스템 설계를 제시한다.

키워드 : 이동궤적 데이터 쿼리, 데이터 색인, 맵리듀스, LBS

Design of Trajectory Data Indexing and Query Processing for Real-Time LBS in MapReduce Environments

Jaehwa Chung*

Abstract

In recent, proliferation of mobile smart devices have led to big-data era, the importance of location-based services is increasing due to the exponential growth of trajectory related data. In order to process trajectory data, parallel processing platforms such as cloud computing and MapReduce are necessary. Currently, the researches based on MapReduce are on progress, but due to the MapReduce's properties in using batch processing and simple key-value structure, applying MapReduce framework for real time LBS is difficult. Therefore, in this research we propose a suitable system design on efficient indexing and search techniques for real time service based on detailed analysis on the properties of MapReduce.

Keywords : Trajectory Data Querying, Data Indexing, MapReduce, LBS

1. 서론

최근 센서 기능의 발전과 센서의 소형화는 센서를 응용한 스마트 기기의 대중화을 앞당겼으며 일상 생활에서 수집되는 데이터를 가공하여 산업 및 사회 전반에 이용하는 서비스가 등장하기 시작하였다. 이러한 서비스 중 위치기반 서비스(LBSs: Location-Based Services)는 사용자 또는 사물의 위치를 이용하여 관련된 데이터를 제공하는 서비스로 GPS와 자이로스코프 등 위치 및 공간을 인식할 수 있는 스마트 기기의 대중화로 국내·외에서 폭발적인 관심받는 서비스가 되었다[1]. 특히 소셜 네트워킹 서비스(SNSs: Social Networking Services)와 같이 다른 형태의 스마트폰 서비스와 LBS가 결합하는 새로운 모바일 위치기반 서비스 시장을 창출하였다.

※ 교신저자(Corresponding Author): Jaehwa Chung
접수일:2013년 07월 11일, 수정일:2013년 7월 28일
완료일:2013년 09월 25일
* 한국방송통신대학교 컴퓨터과학과
jaehwachung@knou.ac.kr
▣ 이 논문은 2012학년도 후기 한국방송통신학교 학술연구비 지원을 받아 작성된 것임.

전통적인 LBS는 2차원 또는 3차원 공간에서 좌표계의 값으로 표현되는 특점 시점의 위치 데이터 사용하여 조건에 일치하는 거리(distance) 또는 면적(area) 데이터를 제공하는 공간 질의처리 기술을 활용하는 서비스를 일컫는다. 그러나 단순한 위치에서 벗어나 한 객체에 대하여 특정 시간, [t_1, t_2], 동안 움직인 자취를 알 수 있으면 객체에 대한 특성 및 기호, 행적 등을 추가적으로 판단할 수 있어 기존의 LBS 보다는 한 단계 진보한 서비스 개발이 가능하다. 이렇게 객체가 이동하는 경로를 단위 시간별 위치정보를 연속적으로 기록한 데이터를 이동궤적(trajectory) 데이터라고 하며, 객체의 현재와 과거의 모든 자취 정보를 모두 포괄·관리하는 이동궤적 DBMS를 이용하여 서비스가 가능하다.

이동궤적 데이터는 일반적인 객체 데이터와는 달리 객체의 수 변화와 관계없이 시간에 따라 데이터가 지속적으로 누적되어 초 대용량적으로 증가하는 특징을 가지고 있다. 최근 진보적인 LBS를 가능하게 하기위해 이동궤적 데이터 관리 기법에 대한 연구가 진행되고 있으나 복잡한 데이터 구조와 방대한 데이터 양 등의 이동궤적 데이터의 속성으로 인해 아직까지는 연구 및 개발 결과가 제한적이다.

따라서 본 논문은 이동궤적 데이터의 초대용량 특성으로 비롯되는 서비스 및 처리기술의 제한점을 해소하여 차세대 LBS에서 중요하게 요구되는 실시간적 기술을 구현하기 위한 시스템 아키텍처를 제안한다.

2. 관련연구

이동궤적 데이터를 효율적으로 색인 및 질의처리하기 위해 [2]의 경우 기존 색인 기법이 가진 이동궤적 데이터에 대한 질의 처리 시 비효율성을 해소하기 위하여 이동궤적을 기준으로 노드를 분할하는 R-트리 기반의 TB-트리와 STR-트리를 제안하였다. 또한 [3]의 연구는 이동궤적 정보를 관리하기 위하여 SETI(scalable and efficient trajectory index) 색인 구조를 제안하고 시간 정보와 위치 정보를 나누어 색인하였다. 이동궤적 데이터는 위치 정보에 기반하여 셀 단위로 분할하여 튜플 형태로 저장하고, 시간 정보는 각각의 시간별로 별도의 R-트리를 구축하여 정보를 관리한다. 위의 연구는 중앙집중식 환경에서 효율적인 이동궤적 데이터 관리 및 처리에 대한 연구로 데이터의 크기가 하나의 서버에서 처리가 불가능할 경우 적용하기가 어려운 문제점을 가지고 있다.

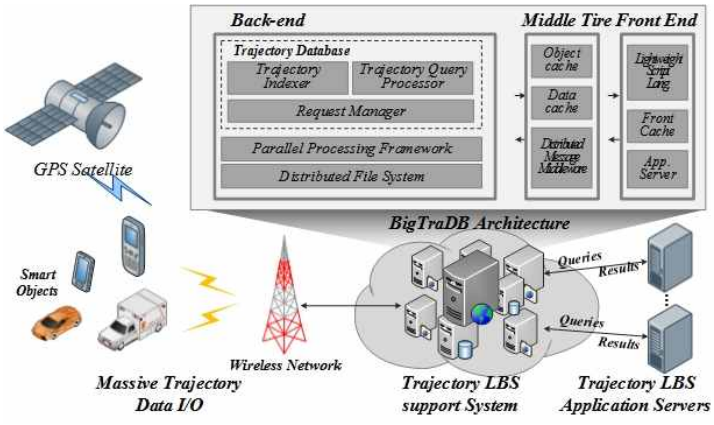
최근 이동궤적 데이터의 양이 폭발적으로 증가함에 따라 클라우드 컴퓨팅 및 병행 처리 모델을 기반으로 하는 이동 궤적 데이터의 맵리듀스 기반의 분산처리 연구로서 [4]는 위치 포인트들로 구성되는 공간 데이터 질의 처리를 병행 프로그래밍 기법인 맵리듀스를 사용하여 효율적으로 해결하고자 한 연구이다. 해당 연구에서 고려하는 질의 처리는 세 가지로써 한 지점(point) 혹은 지역(region)에 대한 공간 선택 질의(spatial selection query), 공간 결합 질의(spatial join query) 그리고 근접 이웃 질의(nearest neighbor query)이며 이들 중 검색 질의와 결합 질의를 맵리듀스 프레임워크 상에 적용하였다. [4]는 초 대용량 이동궤적 데이터의 효율적인 처리를 위해 맵리듀스 프레임워크를 구성하여 대규모 클러스터 환경에서의 병행 질의 처리를 연구하였다. 이동궤적 데이터는 두 개의 이동점을 잇는 직선 단위인 세그먼트로 분할되어 관리되며 세그먼트들은 시간 간격별로 정적 색인 기법인 SETI에 기반하여 색인된다. [6]의 연구는 지리적 근접성을 통하여 쿼드트리 기반의 글로벌 색인구조와 파일단위의 지역 색인구조를 바탕으로 맵리듀스 기반 데이터 관리 및 처리 기법을 제안하였다.

기존의 연구들은 분산 환경에서 초 대용량의 데이터를 처리하기 위한 기법을 제안하고 있지만 모두 실시간적 특성을 향상시키기 위한 기법보다는 맵리듀스에 적용하는데 집중되었다. 따라서 본 논문에서는 실시간 이동궤적 LBS 시스템을 위한 아키텍처를 제안하고 클라이언트 미들웨어 모듈을 설계하고자 한다.

3. 시스템 설계

본 논문에서 제안하는 시스템은 분산된 스토리지 위에서 설치되는 실시간 질의처리 플랫폼 시스템이며 전체 데이터집합은 다수의 클러스터

(그림 1) 이동궤적 LBS 시스템 및 BigTran 아키텍처 개념도



(Figure 1) Overview of Trajectory LBS System and BigTraDB Architecture

서버에 분할되어 저장되는 환경을 가정한다.

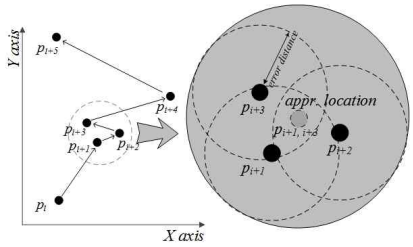
[정의 1] 이동궤적 데이터의 실시간적 질의처리 플랫폼 시스템(BigTraDB): BigTraDB는 (그림 1)과 같이 하나의 마스터 노드와 다수의 데이터 노드로 이루어지는 서버 클러스터 환경에서 분산 파일 시스템과 어플리케이션 사이에 존재하며 스토리지 모듈, 색인 모듈, 질의처리 모듈로 구성된 이동궤적 데이터 관리 및 처리를 지원하는 미들웨어 시스템으로 정의한다.

[정의 1]에 따라 스토리지 모듈은 사용자(이동객체)의 초대용량 이동궤적 데이터들을 분산된 클러스터 데이터 서버에 효율적으로 분할하여 저장하는 기능을 담당하는 모듈이며 색인 모듈은 인덱스를 생성하여 분산된 클러스터 데이터 서버에 있는 이동궤적 데이터들에 대한 관리 및 검색의 효율을 높이는 기능을 담당한다. 그리고 질의처리 모듈은 색인 모듈과 연동하여 위치기반 어플리케이션에서 요구하는 질의에 해당하는 데이터를 효율적으로 찾는 기능을 담당한다.

3.1 데이터 모델 설계

BigTraDB는 일정 시간 구간마다 이동 객체의 이동궤적을 라인 세그먼트 모델[1]에 따라 표현한다. 따라서 [정의 2]와 같이 객체의 위치를 포인트로 표현하고 두 개의 포인트를 라인으로 연결하여 하나의 이동궤적으로 정의한다.

(그림 2) 이동궤적 데이터 표현 모델



(Figure 2) Line Segment Model

[정의 2] 이동궤적 이력 데이터(Historical Trajectory Data, HTD): 이동궤적 이력 데이터는 이동 객체의 GPS 로그 데이터를 바탕으로 생성하며 다음과 같이 정의한다.

$$p_i^{oid} = (x, y, t_i)$$

단, x와 y는 위치정보, o는 이동 객체의 ID, i는 타임시퀀스를 나타낸다.

$$\lambda_{i,j}^{oid} = (p_i^{oid}, p_j^{oid})$$

단, i와 j는 타임시퀀스이고 i≠j이다.

따라서, 라인 세그먼트 $\lambda_{i,j}^{oid}$를 이용하여 이동궤적 데이터 HTD는 다음과 같이 정의된다.

$$HTD^{oid} = (\lambda_{i,j}^{oid}, \lambda_{j+1,k}^{oid}, \cdots)$$

BigTranDB는 이동궤적 데이터를 효과적으로

관리하기 위해 단순화한 근사 위치(approximated location) 기법을 적용한다. 근사 위치란 시스템에서 지정된 거리(Δ) 이내에서 객체의 위치가 특정 시간동안 변동되면 하나의 위치로 함축하여 표현한다. (그림 2)에서 [p_{i+1}, p_{i+3}]는 $p_{i+1,i+3}$으로 표현되며, $p_{i+1,i+3}$의 위치는 GPS의 오차범위를 고려한 모든 위치의 외접원의 중심으로 결정된다.

HTD에 대해 보다 효율적인 데이터집합 구성을 위한 기간 이력 데이터를 정의한다. 기간 이력 데이터는 이동 객체의 이동궤적 이력 데이터를 특정 주기(e.g. 0시 ~ 24시까지의 하루 주기)를 바탕으로 데이터를 구성하는 형식이며 그 정의는 다음과 같다.

[정의 3] 기간 이동궤적 이력 데이터(Period Historical Trajectory Data, PHTD): 기간 이력 데이터는 이동 객체의 특정 주기 내의 이동궤적 데이터의 집합으로 다음과 같이 정의한다.

$$PHTD_{i,j}^{oid} = (\lambda_i^{oid}, \lambda_{i+1}^{oid} \cdots, \lambda_j^{oid})$$

여기서, oid는 이동 객체의 ID, i와 j는 타임시퀀스를 나타낸다.

3.2 스토리지 모듈 설계

BigTraDB는 이동궤적 이력 데이터의 확장 모델인 기간 이동궤적 이력 데이터 집합을 효율적으로 분할하여 저장한다. 또한 초 대용량 이동궤적 이력 데이터 삽입 및 이동궤적 질의처리 시 효율적인 맵리듀스를 이용한 병렬처리가 가능하게 하며 색인 및 질의처리 모듈의 이동궤적 인덱스 생성 및 질의처리에서 유클리드 거리 및 궤적 척도(metric)를 기반으로 하는 효율적인 공간 필터링(spatial filtering)을 지원한다.

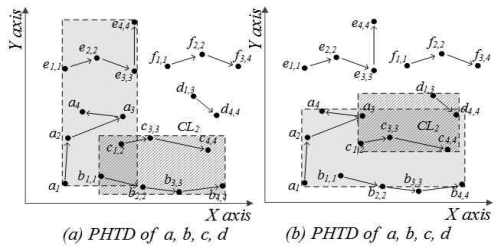
기존의 이동궤적 데이터에 대한 정적(static) 분할 방식의 전체 데이터 집합 재분할 부하 균형(load balancing)에 문제점 해결하기 위해 [정의 4]와 같이 새로운 척도를 제안한다.

[정의 4] 유클리드 거리 및 궤적 척도(λdist): 유클리드 거리 및 궤적 척도는 각 라인 세그먼트 단위로 계산하며 라인 세그먼트의 거리척도는 다음과 같이 정의한다.

$$\lambda dist(\lambda^i, \lambda^j) = \Gamma \bullet dist(\lambda^i.c, \lambda^j.c)$$

여기서, Γ는 같은 이동 객체의 라인 세그먼트에 가중치 상수를 $\lambda^i.c$는 객체 i의 라인 세그먼트의 중점을 나타낸다.

(그림 3) 최소 영역 겹침 클러스터링



(Figure 3) Minimal Overlap Area Clustering

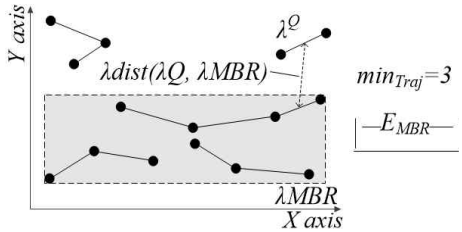
(그림 3)에서와 같이 유클리드 거리 및 궤적 척도만으로 클러스터 집합을 구하였을 때 다양한 형태의 클러스터 집합이 구성될 수 있다. 클러스터 집합 구성 시 오버랩 영역이 커지면 데이터 분할에 대한 효율성이 저하되며 색인과 질의처리에서 큰 비용이 발생한다. 따라서 BigTraDB는 클러스터 구성 시 영역 간 최소화된 오버랩을 가지는 클러스터링 기법을 통하여 보다 효율적인 데이터 분할을 수행한다. 최소 영역 겹침 클러스터링 분한기법은 이동 객체의 위치 포인트를 클러스터링하기 위한 알고리즘은 DBSCAN[9]과 유사하게 일정한 밀도(density)를 유지하며 접근 가능한 거리(reachable)를 기준으로 하여 확장하도록 설계한다. 이와 더불어, 최소 영역 겹침을 만족하기 위해 겹침(degeneracy) 정도를 확장의 필요조건으로 사용한다. 즉, 이동궤적의 밀도-접근 가능-겹침을 기준으로 하는 클러스터링을 진행하게 되며, 그 정의는 다음과 같다.

[정의 5] 밀도-접근 가능 (density-reachable) : 라인 세그먼트 λ^Q와 서로 다른 이동궤적의 라인 세그먼트 집합 λ를 포함하는 λMBR이 주어질 때, 아래의 조건을 만족하면 λ는 밀도-접근 가능이다.

1) $|\{oid \in \lambda\}| > \min_{Traj}$

2) $(\exists \lambda | \lambda dist(\lambda^Q, \lambda) \leqq E_{MBR})$

(그림 4) 밀도-접근의 예



(FIgure 4) an example of density-reachable

[정의 6] 밀도-접근 가능-겹침 (density-reachable-degeneracy): 밀도-접근 가능을 만족하는 라인 세그먼트의 집합 λ_{DR}과 그들을 담는 $\lambda_{DR}MBR$이 주어질 때, 아래의 조건을 만족하면 λ_{DR}은 밀도-접근 가능-겹침을 만족한다.

$$\frac{\{oid \in \lambda_{DR}\} \cap \{oid \in \lambda_{DR}MBR\}}{\{oid \in \lambda_{DR}\} \cup \{oid \in \lambda_{DR}MBR\}} \geqq \epsilon$$

ϵ는 *degeneracy ratio*을 나타낸다.

클러스터링 분할 기법은 최초에 최소 포함 이동궤적 수를 의미하는 $\min_{Traj}$ 이상의 라인 세그먼트로 클러스터를 구성하며, 밀도-접근 가능을 만족하는 라인 세그먼트를 추가하여 클러스터를 확장해 나간다. 클러스터에 새로운 라인 세그먼트 λ^Q가 들어오면, 이를 담는 λMBR을 재계산한다. λMBR에는 기존 클러스터의 밀도 보장 거리 E_{MBR} 바깥의 라인 세그먼트도 포함될 수 있으므로 λ_{DR}과 $\lambda_{DR}MBR$은 서로 다른 *oid*를 가질 수 있다. 만약에 새로이 구성되는 λMBR에 다수의 새로 클러스터에 포함되는 이동궤적이 들어올 경우, 최소 겹침 영역이 위배될 가능성이 높아지므로, 겹침 비중(degeneracy ratio) 값으로 이를 제어한다.

3.3 색인 모듈 설계

기존의 R-tree [6]는 중앙 집중식 환경에서 적합한 구조로 연구되었기 때문에 트리 구성 시 하향식(top-down) 구조로 노드를 확장해 나가는 구조를 갖는다. 그러나 BigTraDB는 일정 기간 또는 주기로 대용량의 기간 이력 데이터 집합을 입력 받는(bulk loading) 방식을 사용하기 때문에 서버의 인덱스 유지에 필요한 부하를 최소화하고 질의의 탐색 시간을 크게 향상시킬 수 있는 상향식 구조로 트리를 구성하는 색인 구조가 요구된다.

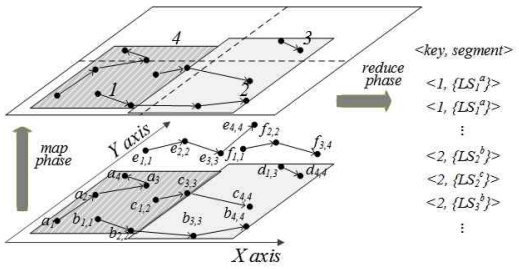
R-tree기반 상향식 인덱스는 초 대용량 이동궤적 데이터에 대해 공간 기반의 질의에서 매우 효율적인 데이터 검색을 지원하지만 시간 기반의 질의에서는 비효율적인 탐색을 초래한다. 따라서 색인 모듈 설계 시 시간에 대한 데이터의 색인은 별도의 B 트리 기반의 자료구조 [8]로 구축하며 R-tree 노드와 매핑을 하여 관리한다.

BigTraDB는 이동궤적 데이터 입력 유형을 두 가지로 분류한다. 첫째는 실시간으로 다수의 이동 객체들에 대해 현재 위치를 입력받는 형태이다. 실시간 디스크 I/O 비용이 매우 크기 때문에 그리드 기반 논리적 공간을 사용한다. 질의처리 모듈은 분산된 다수의 데이터 노드들 중에 데이터 수용능력의 한계로 데이터를 새로운 노드로 분할하는 연산 모듈이 필요하다. 둘째는 특정 기간 또는 주기 단위로 입력되는 형태로 BigTraDB의 인덱스 모듈은 초 대용량 데이터에 대해 맵리듀스 환경에서 효율적인 병렬처리를 가능하게하기 위해 일정 기간 또는 주기 단위의 기간 이력 데이터에 대해 효율적인 분할(partitioning) 및 색인을 수행한다. 그러나 실시간 위치기반 서비스를 지원하기 위해서는 이동객체들의 현재 위치 정보를 입력을 받아서 서비스를 제공해야하기 때문에 이러한 유형의 데이터 입력에 대해서는 계산 비용이 작은 논리적 그리드기반 맵을 이용한다. (그림 5)와 같이 클러스터 단위의 기간 이력 데이터집합에 대해 맵리듀스 기반의 병렬 삽입 알고리즘을 이용하여 데이터를 삽입한다. 각 클러스터는 그리드기반 논리적 공간에 매핑되어 클러스터의 키(key)를 부여받고 각 데이터 노드들은 맵 단계에서 자신의 부여받은 해당 키의 클러스터 데이터들을 읽는다.

3.4 질의처리 모듈 설계

이동궤적에 대한 질의는 이동 객체의 데이터 상의 질의로 정의되며 크게 질의 확장을 위해 두 가지 유형으로 추상화하여 작은 지연시간이

(그림 5) 맵리듀스 기반 데이터 삽입



(Figure 5) MapReduce-based Data Insert

요구되는 위치기반 서비스와 같은 실시간 서비스를 지원한다. 이를 위해 BigTraDB는 이동궤적 질의를 크게 두 유형으로 구분한다.

첫 번째 유형으로 다양한 이동궤적 선택 질의들에 대한 분석 및 공통 특징을 추출하여 확장에 기본 규격이 되는 추상화 단계를 거쳐 설계를 진행한다. 마찬가지로 질의처리 모듈에서도 이러한 구조로 선택 유형의 질의들을 분석 및 상위 선택 오퍼레이터(selection operator)로 정의함으로써 선택 유형의 질의에 유연하고 효율적인 플랫폼 시스템이 가능하게 한다. BigTraDB는 기본적으로 선택 유형의 질의들 중 가장 대표적인 두 개의 질의에 대한 효율적인 처리 모듈을 가지도록 설계한다. 포인트 질의(point query)은 사용자가 지정한, 포인트로 표현되는 정확한 위치 및 원하는 시간에 존재하는 객체를 검색을 요청하는 질의를 의미하며 범위 질의(range query)로 사용자가 지정한, 영역으로 표현되는 지역 및 원하는 시간에 존재하는 객체를 검색을 요청하는 질의를 의미한다.

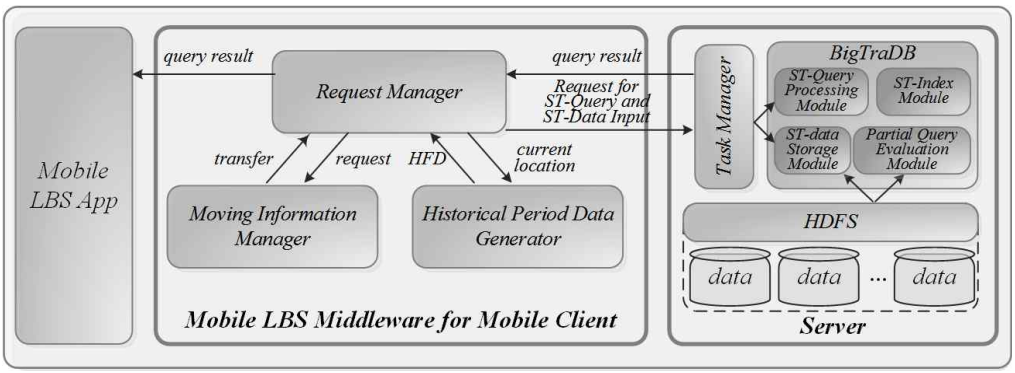
BigTraDB에서 질의처리의 실시간적 특성을 극대화시키기 위해 질의처리 모듈은 필터링 모듈을 통해 맵리듀스의 처리 비용을 최소화할 수 있다. 시공간 필터링은 이동궤적 데이터의 공간적 특성(유클리드 거리) 및 시간적 특성을 이용하여 질의처리 시 최소한의 인덱스 탐색을 가능하게 한다. 이는 맵리듀스를 이용한 병렬처리 시 최소한의 데이터 노드를 이용함을 의미하며 곧 클라우드 환경에서 자원 사용에 대한 비용 감소 및 전체적인 질의처리 시스템의 성능을 향상시킨다. 선택 유형의 질의들은 그리드기반 논리적 공간을 통해 보다 빠르게 처리할 수 있다.

두 번째 유형의 질의는 근접객체 질의에 대한 처리이다. 이동궤적 근접객체 질의는 기존의 이동 객체의 이동궤적 데이터 상에서 사용자가 원하는 위치에서 근접된 객체 정보를 요청한다. 선택 질의와 마찬가지로 질의처리 모듈은 다양한 이동궤적 근접객체 질의들에 대한 분석 및 공통 특징을 추출하여 확장에 기본 규격이 되는 추상화를 제공한다. 또한 근접객체 유형의 질의들을 분석 및 근접객체 오퍼레이터(NN operator)로 추상화하여 근접객체 검색 유형의 질의에 유연하고 효율적인 질의처리 시스템이 가능하게 한다. BigTraDB는 기본적으로 근접객체 검색 유형의 질의들 중 가장 대표적인 두 개의 질의를 지원한다. 첫 번째는 포인트 기반의 k-Nearest Neighbor(p-kNN) 질의로 사용자가 원하는 시간 및 포인트로 표현되는 정확한 위치를 기준으로 가장 근접하는 객체 k개를 검색한다. 두 번째는 이동궤적 기반의 k-Nearest Neighbor(t-kNN) 질의로 사용자가 원하는 시간 및 사용자의 이동궤적을 기준으로 가장 근접하는 객체 k개를 검색을 요청한다. 마찬가지로 근접객체 유형의 질의를 실시간적으로 처리하기 위해서는 필터링 모듈이 반드시 필요하며 선택 유형의 질의처리 필터와는 다르게 근접객체 질의처리에서는 물리적 클러스터기반의 계층적 분산 색인 모듈을 이용하여 효율적으로 처리한다.

실시간성을 위해 이동 객체 상에서 연속적 이동궤적 질의처리를 모듈을 도입한다. 이동 객체의 이동 정보를 재생성할 때, 부분 질의 재계산(partial query processing)을 통해 데이터 접근 및 인덱스 사용을 최소화 한다.

(그림 6)는 현재 사용자 즉 이동 객체가 이동 중에 지속적인 위치기반 서비스를 받는 상황을 나타낸다. p-NN 질의를 요청하였을 경우, b의 이동궤적 데이터가 가장 가깝게 존재하기 때문에 클러스터 2에 존재하는 b가 결과로 반환되는 예를 보여준다. 하지만 사용자는 이동을 하기 때문에 일정 시간이 지나면 질의의 결과가 클러스터 1에 존재하는 C가 질의의 결과로 업데이트 되어야한다. 따라서 질의처리 모듈에서는 이동 중의 지속적인 질의요청이 있을 때 사용자의 이동 정보를 통한 검색 공간을 축소하여 처리하는 부분(partial) 질의처리 모듈을 제공한다. 예를 들어 사용자의 이동 정보를 받아 클러스터 2에

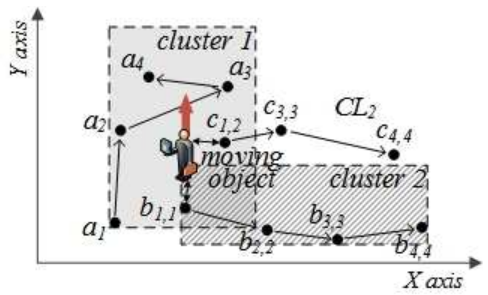
(그림 7) 모바일 클라이언트 모듈 설계



(Figure 7) Design of Mobile Client Modules

질의의 결과 후보가 존재할 수 없음을 계산하고 클러스터 1에서만 질의처리를 수행하도록 한다.

(그림 6) 연속적인 p-kNN 질의의 예



(Figure 6) An Example of Continuous p-kNN Query

3.5 클라이언트 모듈 설계

모바일 위치기반 서비스 지원을 위한 BigTraDB는 서버 측 모듈 구성만으로는 다양한 어플리케이션 지원에 한계가 있기 때문에 모바일 위치기반 어플리케이션의 중간 모듈이 필요하다. (그림 7)은 BigTraDB의 서버/클라이언트 아키텍처를 나타내고 있으며 클라이언트에 위치하는 세부 모듈과 그 관계를 나타낸다. 클라이언트 측의 질의 요청 모듈(Query Request Manager)는 서버에 이동궤적 질의처리 및 데이터의 입력을 수행한다. 그리고 이동하는 객체에 의한 질의처리 요청은 이동 정보를 이동 정보 관리 모듈(Moving Information Manager)에서 생성하여 서버 측에 함께 전송한다. 또한 데이터 구조화 모듈(Data Specification Manager)는 사용자의 GPS 로그를 바탕으로 기간 또는 주기별로 이동궤적 데이터를 새로운 데이터 모델로 변환 및 반환하는 모듈로 질의 요청 모듈로 전송하여 서버로 전송한다. 마지막으로 질의 결과 관리 모듈(Query Result Manager)는 서버로부터 전송받은 질의결과를 관리하여 모바일 위치기반 어플리케이션에 전달해주는 역할을 담당한다.

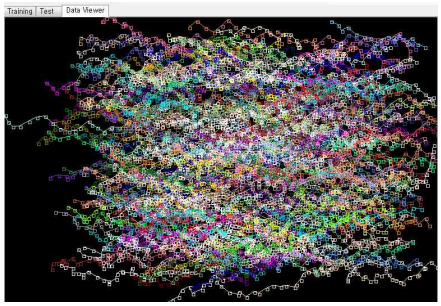
4. 실험 및 평가

제안된 시스템의 유효성을 위해 실험은 앞에서 살펴본 모듈들 중에서 핵심 기능인 스토리지 모듈의 클러스터 기반의 데이터 분할 기법에 대해 알고리즘 구현을 진행하고 검증 및 평가 하였다. 실험에 사용된 이동궤적 데이터는 GSTD[9]를 이용하여 생성하였고 사용된 이동객체 수는 500개이며 총 세크먼트 수는 9153개이다. (그림 8)는 실제 실험에 사용된 데이터 분포를 보여준다.

제안된 기법과 기존 연구 기법의 분할 알고리즘의 성능을 비교한 실험결과 제안된 클러스터 기반의 분할 기법(DegScan)이 22%, 기존 연구(DBScan)는 44%의 겹침 영역을 보여 제안된 기법이 약 50% 향상된 결과를 보였다. 이는 분산처리에 대해 데이터 지역성을 높이게 되어 보다 효율적인 데이터 관리 및 처리가 가능하게 된다.

(그림 9)는 제안된 클러스터기법이 기존 연구 기법보다 데이터 분포를 고르게 갖는 것을 나타내고 있다. (그림 9) a에서 기존 기법에는 특정

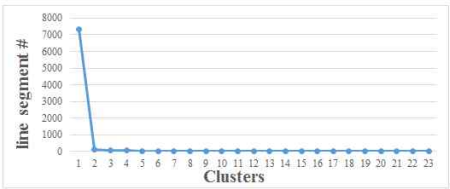
(그림 8) 실험에 사용된 이동궤적 데이터



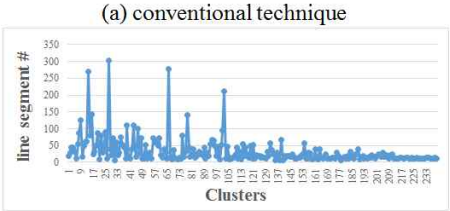
(Figure 8) trajectory data for experiments

클러스터에 편중되는 경향(표준편차 1526.01)을 보이는 반면, (그림 9) b에서 제안된 기법은 기존 기법과 대비하여 상대적으로 균형적으로 분포된 결과(표준편차 39.32)를 나타낸다. 기존 기법으로 클러스터링을 하게 되었을 경우 하나의 노드에 부하집중현상이 발생할 수 있으며 이것은 전체 시스템의 성능을 저하시킬 수 있다.

(그림 9) 클러스터별 데이터 분포



(a) conventional technique



(b) proposed technique

(Figure 9) Data distribution in each cluster

5. 결론

본 논문은 초 대용량 이동궤적 데이터 환경에서 분산된 스토리지에 저장된 데이터를 효율적으로 병행적 색인 및 질의처리를 할 수 있는 실시간적 시스템 설계를 제안하였다. 맵리듀스의 일괄처리적 특성을 실시간적으로 향상할 수 있는 본 연구는 시스템을 크게 스토리지, 색인, 질의처리 모듈로 나누어 시스템 설계를 진행하였다. 또한 부가적으로 클라이언트에서의 모듈들을 설계함으로써 전체 시스템에 실시간적인 성능 향상을 할 수 있게 설계를 진행하였다. 향후 본 시스템 설계를 바탕으로 우리는 하둡 기반의 시스템을 구현하고 그것을 바탕으로 다양한 데이터 환경에서의 비교 실험을 통해 본 시스템의 우수성을 증명하고자 한다.

References

[1] Eun-Jee Song, “A Case of the Mobile Application System Development using Location Based Service“, Journal of Digital Contents Society Vol. 13 No. 1 pp.53-60 Mar. 2012

[2] Dieter et al, “Novel Approaches to the Indexing of Moving Object Trajectories”, VLDB, 2009

[3] V. Prasad et al, “Indexing large trajectory data sets with seti”, CIDR, 2003

[4] Shubin et al, “Spatial queries evaluation with mapred uce”, GCC, 2009

[5] Qiang et al, “Query processing of massive trajectory data based on mapreduce”, CloudDB, 2009

[6] Yunqin et al, “Towards parallel spatial query process ing for big spatial data”, HPDIC, 2011

[7] Guttman, A ″R-trees : A dynamic index structure for spatial searching″ Proc. ACM SIGMOD, 1984

[8] Comer, Douglas, ″The Ubiquitous B-Tree″, Computi ng Surveys, June, 1979

[9] Martin et al. “A Density-Based Algorithm for Discov ering Clusters in Large Spatial Databases with Nois e”, KDD, 1996

정 재 화

1999년 : 고려대학교 대학원 (이학석사)

2011년 : 고려대학교 대학원 (이학박사-공간질의처리)

2011년～2012년: 고려대학교 정보창의연구소 연구교수

2012년～현 재: 한국방송통신대학교 컴퓨터과학과 조교수

관심분야 : 공간질의처리 및 인덱싱(Spatial Query Processing and Indexing), 모바일 데이터 관리(MDM), 무선 센서 네트워크(WSN)