
LBSNS에서 연속 질의 빈도 감소를 위한 가상그리드 기법의 설계 및 구현

이은식* · 조대수**

A Design and Implementation of Virtual Grid
for Reducing Frequency of Continuous Query on LBSNS

Eun-Sik Lee* · Dae-Soo Cho**

이 논문은 2011년도 동서대학교 "Dongseo Frontier Project" 지원에 의하여 이루어진 것임

요 약

웹 상에서 사용자들 간의 관계(Relation)를 통해 인적 네트워크를 형성할 수 있게 해주는 온라인 서비스를 SNS(Social Networking Services)라고 한다. 최근에는 GPS가 내장된 디지털장치(스마트 폰, 태블릿 PC 등)를 통해 획득한 위치 정보를 SNS에 추가한 LBSNS(Location-Based SNS) 서비스에 대한 관심이 증가하고 있다. 사용자가 원하는 관심지역의 정보만을 구독하는 LBSNS 서비스를 구현하기 위해서는 공간필터링 기법이 요구된다. 공간필터링을 위해서는 트윗에 위치정보를 포함해야 한다. 위치정보는 사용자의 이동에 따라 함께 변하는 동적정보로 표현되거나 최초위치에서 고정되는 정적정보로 표현될 수 있다. 동적 위치정보를 사용할 경우 사용자의 이동에 따라 연속적으로 공간 필터링 질의가 발생하므로 서버에 많은 부하를 줄 수 있다. 본 논문에서는 동적 위치정보로 인해 대량으로 발생하는 연속질의 문제를 해결하기 위해서 가상그리드 기법을 이용한 공간필터링 알고리즘을 제안한다. 성능평가 결과 가상그리드 기법을 활용한 경우 질의 발생빈도 측면에서 최고 93%성능이 개선되었다.

ABSTRACT

SNS(Social Networking Services) is online service that enable users to construct human network through their relation on web, such as following relation, friend relation, and etc. Recently, owing to the advent of digital devices (smart phone, tablet PC) which embedded GPS some applications which provide services with spatial relevance and social relevance have been released. Such an online service is called LBSNS. It is required to use spatial filtering so as to build the LBSNS system that enable users to subscribe information of interesting area. For spatial filtering, user and tweet attaches location information which divide into static property presenting fixed area and dynamic property presenting user's area changed along the moving user. In the case of using a location information including dynamic property, Continuous query occurred from the moving user causes the problem in server. In this paper, we propose spatial filtering algorithm using Virtual Grid for reducing frequency of query, and conclude that frequency of query on using Virtual Grid is 93% decreased than frequency of query on not using Virtual Grid.

키워드

LBSNS, 공간조인, 공간필터링, PBSM, 실체화 된 뷰, 가상 그리드

Key word

LBSNS, Spatial Join, Spatial Filtering, PBSM, Materialized View, Virtual Grid

* 준회원 : 연세대학교 컴퓨터과학과 석사과정

접수일자 : 2012. 01. 30

** 정회원 : 동서대학교 컴퓨터정보공학부 부교수 (교신저자, dscho@dongseo.ac.kr)

심사완료일자 : 2012. 02. 16

Open Access <http://dx.doi.org/10.6109/jkiice.2012.16.4.752>

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

I. 서 론

웹 상에서 사용자들 간의 관계(팔로잉, 멘토링, 친구 등)를 통해 인적 네트워크를 형성할 수 있게 해주는 온라인 서비스를 SNS(Social Networking Service)라고 한다. 무선통신이 가능한 디지털장치들이 등장하면서 SNS 애플리케이션들이 등장하였고 최근에는 GPS가 함께 탑재된 디지털장치(스마트 폰, Tablet PC, 내비게이션 등)가 등장함에 따라 LBSNS(Location-Based SNS)라고 불리는 위치정보 기반의 애플리케이션들이 인기를 얻고 있다. 이 논문에서는 가장 대표적인 SNS인 트위터 서비스를 기반으로 위치정보를 추가하여 자신이 관심있는 지역의 트윗들을 구독하는 위치기반 트위터 서비스를 제안하고자 한다.

현재 트위터 서비스에서도 위치정보를 활용하여 다음과 같은 세 가지 서비스가 제공되고 있으나, 관심 메시지에 대한 구독서비스가 아니라 일회성의 검색 서비스 또는 위치정보 전달 서비스로 제공되고 있다. 첫 번째는 #태그를 이용한 지역검색이다. 예를 들어, “#Busan” 조건을 통해서 ‘Busan’이라는 단어를 포함하는 모든 트윗들을 검색 할 수 있다. 그런데, 이 경우에 “#Busan”이라는 키워드의 결과로 나온 트윗들이 실제로 부산지역의 위치와 관련된 것인지는 확실하지가 않다. 왜냐하면 단지 텍스트 매칭으로 인해 “Busan”이라는 글자가 들어간 트윗들의 검색 결과이기 때문이다. 두 번째는 공간 연산자(Spatial Operator)를 이용한 검색 방법이다. 예를 들어 “near:Pusan within:15mi”조건을 통해 부산 인근 15마일 내에서 작성한 사용자들의 트윗들을 검색 할 수 있다. 이 경우에는 트윗을 작성한 시점의 작성자의 위치를 기반으로 트윗들을 검색하기 때문에 트윗의 내용자체가 해당지역과 관련되었는지 알 수가 없다. 세 번째는 자신의 관심 지역 정보를 트윗에 포함하여 팔로워들에게 전달하는 기능이다. 팔로워는 자신이 팔로잉 한 사용자가 보낸 트윗과 함께 그 사용자의 위치정보를 지도상에서 볼 수 있다.

본 논문에서 제안하는 위치기반 트위터 서비스는 사용자가 위치기반 구독조건을 등록해 놓으면 해당 조건에 만족하는 메시지를 구독할 수 있는 서비스를 의미한다. 기존 트위터의 구독조건은 본인이 관심이 있는 사용자를 지정하는 것이다. 구독조건으로 지정된 사용자

는 자신의 팔로잉(following)이 되고, 자신은 그 사용자의 팔로워(follower)가 된다. 하지만 위치기반 트위터에서는 자신이 팔로잉한 사람의 트윗들 중에서 자신이 관심 있어 하는 지역과 관련된 것들을 구독 할 수 있어야 하므로 구독조건으로 사람 뿐 아니라 위치정보도 필요하다.

본 논문에서는 사용자가 원하는 관심지역정보만을 구독하기 위해서 그림 1과 같은 공간필터링 기법을 활용한다. 공간 필터링 기법은 기존 트위터 서비스에 비해 다음과 같은 차이점이 존재하는 위치기반 트위터 서비스가 가능하게 한다.

- ① LBSNS에서 사용되는 위치 트윗은 텍스트 메시지 뿐 아니라 언급하는 지역에 대한 위치정보가 포함된다.
- ② 위치기반 구독자는 자신이 구독하려는 위치 트윗들을 포함하는 관심지역을 구독조건에 명시 하여야 한다.
- ③ 다수의 위치 트윗들과 위치기반 구독자를 서로 매핑시키기 위해서는 연속적인 공간필터링 연산이 요구된다.

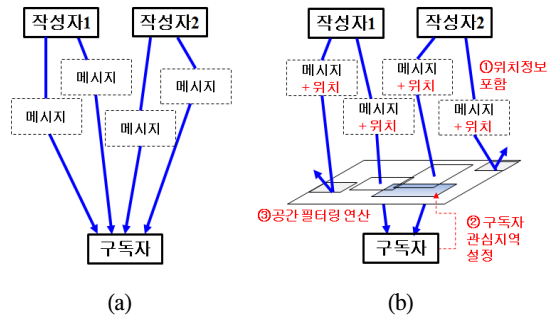


그림 1. 공간필터링 기법
 (a) 기존 트위터 서비스 (b) 위치기반 트위터 서비스
 Fig. 1 Spatial Filtering Technique
 (a) twitter service (b) location-based twitter service

위치정보는 특정한 지점을 가리키는 것으로써 Point로 표현될 수도 있으며, 보다 넓은 범위의 영역을 가리키는 것으로써 MBR(Minimum Bounding Rectangle)로 표현될 수도 있다. 본 논문에서 다루려는 위치정보는 후자에 더 가까우므로, 위치정보를 MBR로 표현한다. 구독자와

트윗은 각각 위치정보를 가질 수 있다. 여기서 위치정보는 다시 정적 속성과 동적 속성으로 나눌 수 있는데 정적 속성은 초기 설정 이후 변하지 않는 것을 뜻하며 동적 속성은 위치정보가 지속적으로 변경하는 것을 뜻한다. 동적 속성의 위치정보의 경우 연속적으로 공간필터링 연산을 대량으로 실행해야 하므로 사용자가 많은 SNS의 특성상 서버에 큰 부담을 줄 수 있다. 본 논문에서는 가상그리드(Virtual Grid)를 생성하여 클라이언트에서의 연속 질의 발생 빈도를 줄였다.

본 논문의 구성은 다음과 같다. 2장에서는 실체화된 뷰 기법과 공간조인 기법에 대해 설명한다. 3장에서는 공간필터링 기법을 이용해 설계한 위치기반 트위터 서비스에 대해 설명한다. 4장에서는 가상그리드를 직접 구현하고 성능 평가한 결과를 설명한다. 마지막으로 5장에서는 결론에 대해 설명한다.

II. 관련 연구

2.1. 실체화된 뷰(Materialized View)

실체화된 뷰는 DB에서 뷰에 대한 질의의 결과로 생성되는 임시 데이터를 실제 데이터로 저장함으로써 동일한 질의가 재 수행 되었을 때 빠르게 질의 결과를 활용할 수 있는 기법이다. 본 논문에서는 타임라인의 업데이트와 같은 복잡한 질의를 처리할 때 마다 DB로 접근하는 문제를 보완하기 위해 실체화 된 뷰를 사용한다. 하지만 실체화 된 뷰는 업데이트가 자주 일어나지 않는 환경에서 최적의 성능을 보인다.

2.2. 공간조인(Spatial Join)

공간조인 연산은 두 가지 이상의 공간 데이터 집합 중에서 주어진 공간술어(Spatial Predicate)를 만족하는 모든 튜플들의 쌍들을 구하는 것이다. 예를 들어 그림 2에서 "모든 공원 내(inside)에 있는 건물들을 찾아주세요" 라는 질의가 수행 되었을 때 최종 튜플들의 쌍은 {(A1, B1), (A2, B1), (A3, B2)}로 나타낼 수 있다.

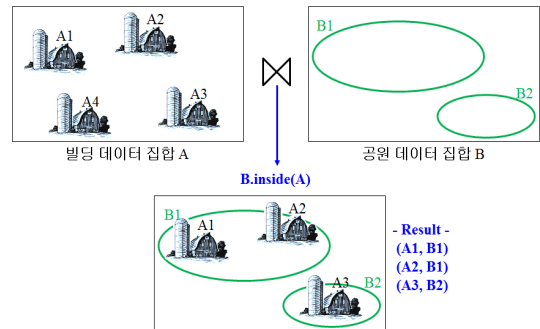


그림 2. 공간조인의 예
Fig. 2 Example of Spatial Join

공간조인은 필터 단계(Filter Step)와 정제 단계(Refinement Step)로 나누어 수행된다. 실제 공간 객체간의 겹침 연산을 수행하는 데는 많은 연산비용이 발생하므로 공간객체의 MBR만을 이용하여 겹침 가능성이 있는 공간객체 쌍을 빠르게 검색하는 것을 필터 단계라고 한다. 필터 단계의 결과물인 공간객체 쌍들이 실제로 겹치는 확인하는 것을 정제 단계라고 한다. 그림 3에서는 두 데이터 셋 간의 겹침 공간조인을 처리하기 위한 필터 단계와 정제단계의 예를 보이고 있다.

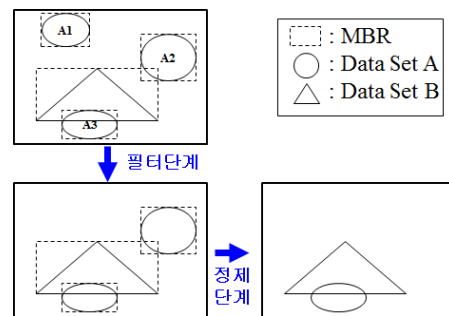


그림 3. 필터단계 및 정제단계
Fig. 3 Filter Step and Refinement Step

필터단계에서 A1은 MBR이 겹치지 않으므로 제거되었고 정제단계에서 A2는 MBR은 겹치지만 실제객체는 겹치지 않으므로 제거되었다. 본 논문에서는 위치 정보를 모두 MBR로 표현하였기 때문에 필터단계만이 사용된다.

2.3. PBSM(Partition Based Spatial-Merge) 조인

공간조인은 조인에 참여하는 데이터 집합에 대해서 공간 색인의 유무에 따라서 여러 가지 알고리즘이 존재한다. 본 논문에서 제안하는 LBSNS에서는 대량의 데이터 삽입연산으로 인하여 색인의 유지비용(색인 재구축 비용)이 많이 발생하므로 공간조인에서 색인을 활용하는 방법은 어렵다. 공간색인을 필요로 하지 않는 대표적인 공간조인 알고리즘으로는 PBSM 조인 알고리즘[2]이 있다.

PBSM 조인에서 전체 데이터 영역을 일정 크기의 파티션으로 분할하고 각 파티션 별로 Plane Sweep 알고리즘을 통해 서로 겹치는 객체 쌍을 필터링 하게 된다. 공간 데이터가 균등하게 분포되지 않아 일정 파티션에 편중되는 경우를 방지하기 위해서 NT Tiles 기법을 사용한다.

III. 위치기반 트위터 서비스

3.1. 위치정보의 모델링

위치기반 트위터 서비스를 설계하기 위해 가장 먼저 할 작업은 구독자들의 관심영역과 위치트윗을 생성하기 위해 위치정보(MBR)를 모델링하는 것이다. 여기서 각 위치정보는 정적 속성과 동적 속성으로 나눌 수 있다. 정적 속성은 초기 설정 이후 변하지 않는 MBR을 뜻하며 동적 속성은 GPS와 같은 위치정보에 의해 지속적으로 변하는 가변적인 MBR을 의미한다.

표 1. 위치정보 모델링
Table. 1 Modeling of Location Information

		구독자 관심영역(A)	
		정적(S)	동적(D)
위치 트윗(T)	정적(S)	TS-AS	TS-AD
	동적(D)	TD-AS	TD-AD

위치기반 트위터 서비스는 구독자의 관심영역과 트윗의 위치정보를 지원하는 방식에 따라 표1과 같이 4가지 모델로 구분될 수 있다. TS(트윗-정적) 모델은 트윗이 특정지역에 대한 정보를 포함할 때 사용될 수 있다. 반면 TD(트윗-동적) 모델은 움직이는 대상(일반적으로는 트

윗 작성자)의 주변 지역에 대한 정보를 포함할 때 사용되므로,대상이 움직임에 따라 트윗이 언급하는 지역도 함께 변경된다. 예를 들면, 빈차로 운행하는 택시기사가 쓴 ‘여기 빈 택시 있어요!’라는 트윗의 경우에 택시가 이동함에 따라 트윗의 위치정보도 함께 변경될 수 있다. AS(구독자-정적)모델은 구독자의 관심영역이 고정되어 있을 때 사용할 수 있으며, AD(구독자-동적)모델은 관심영역이 변경 될 때 사용할 수 있다. 예를 들면, 관심영역을 ‘내 주변 11km’로 설정한 경우에 내 위치가 변경되면 나의 관심영역도 변경될 수 있다.

본 논문에서는 TD모델은 제외하고, TS-AS, TS-AD 모델만을 지원한다. 따라서 트윗은 정적인 속성만 사용 가능하며 사용자들의 경우 정적, 동적 속성 모두 사용이 가능하다. 자신의 관심지역을 정적으로 설정하고 그 지역만의 정보만을 구독 할 수도 있으며 동적으로 설정하여 자신의 위치가 이동함에 따라 달라지는 정보를 지속적으로 구독 할 수도 있다.

3.2. 시스템 설계

이 논문에서 제안하는 위치기반 트위터 서비스의 전체적인 시스템 구조는 그림 4와 같다.

사용자는 트윗 작성자와 트윗 구독자로 나뉘어 질 수 있고, 트윗 구독자는 3.1절에서 소개된 TS-AD모델과 TS-AS모델로 나누어진다. 트윗 구독자는 관심영역을 정적 또는 동적으로 설정 하여 트윗을 받아 볼 수 있다.

트윗 작성자에 의한 삽입, 삭제등의 업데이트는 먼저 최근 위치 트윗들 영역에 저장된다. 저장된 데이터는 일정 시간이 지나면 과거 위치 트윗들(TS모델) 영역으로 이동하게 된다. 과거 위치 트윗들 영역에 있는 데이터는 트윗 구독자가 저장 해놓은 관심영역들 영역에 있는 데이터와 조인하여 실제화된 뷰에 저장 되는데 관심영역들의 영역이 AS모델인지 AD모델인지에 따라 조인결과가 틀려진다.

과거 위치 트윗들과 관심영역들(AS모델)의 실제화된 뷰의 조인결과에 대해서는 주기적 갱신이 가능하다. 하지만 과거 위치 트윗들과 관심영역들(AD모델)의 실제화된 뷰의 조인결과에 대해서는 빈번한 갱신이 일어나게 된다. 그 이유는 동적으로 이동하는 구독자의 위치변화 때문이다. 업데이트가 자주 발생하지 않는 환경에서 최적인 실제화된 뷰의 특성을 고려했을 때 구독자의 위치변화에 의한 빈번한 갱신은 서버에 성능저하를 가져

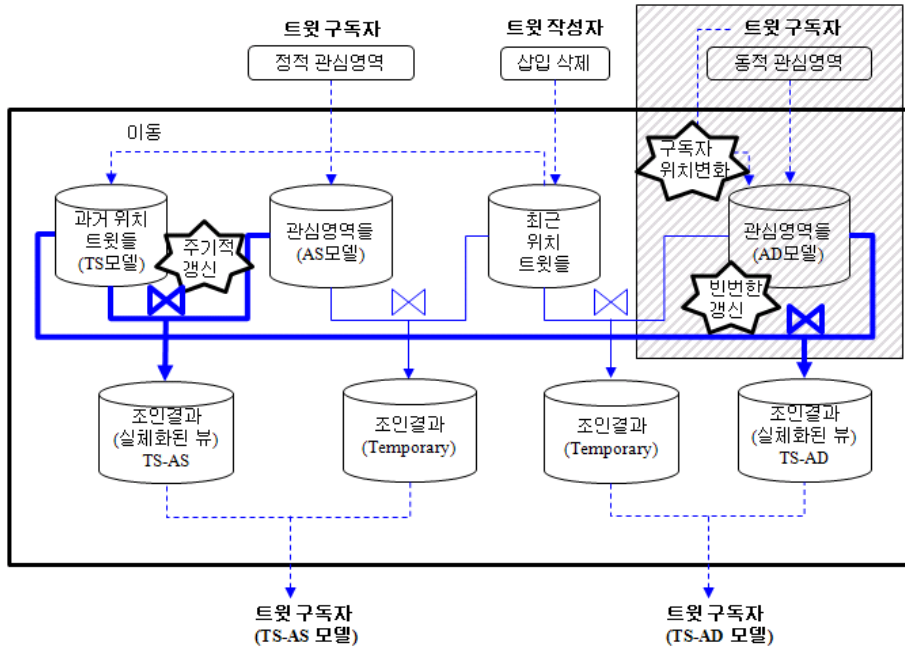


그림 4. 시스템 설계
Fig. 4 System Architecture

온다. 따라서 본 논문에서는 그림 4의 빗금쳐진 영역의 문제를 해결하기 위해 4장에서 가상그리드를 소개하고 성능 평가 한 결과를 보인다.

마지막으로 실체화된 뷰에 저장된 조인결과가 트윗 구독자에게 전달될 때 항상 그 데이터가 최신일수는 없다. 이를 위해 최신 위치 트윗들과 관심영역들의 임시적인 조인을 통한 결과와 실체화된 뷰에 있는 조인결과를 더하여 최종적으로 구독자에게 전달한다.

하기 위해 이 논문에서는 CS(Controlled Sensitivity) 동적 속성을 제안한다.

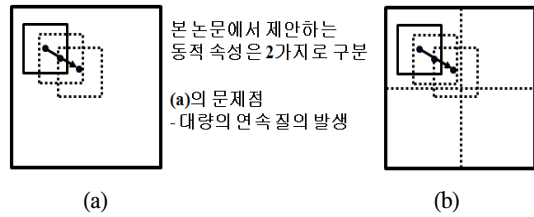


그림 5. 동적 속성의 두 가지 구분
(a) 동적속성 (b) CS 동적속성

Fig. 5 Two classification of Dynamic property
(a) dynamic property (b)CS dynamic property

IV. 가상그리드를 활용한 성능개선

4.1. 민감도 제어 동적 속성

동적 속성은 그림 5의 (a)와 (b)처럼 두 가지로 구분 할 수 있다. 동적 속성의 위치정보를 사용할 경우 사용자의 위치정보가 조금이라도 움직이게 될 때 마다 그와 함께 연속적으로 발생하는 대량의 질의가 발생하므로 SNS 환경에서 서버에 큰 부담을 줄 수 있다. 이러한 점을 개선

CS 동적 속성에서는 가상그리드를 사용하여 전체 MBR을 그리드형태로 나누게 된다. 동일한 그리드 내에서 발생하는 위치변화는 시스템에서 무시함으로써 연속질의 발생을 줄일 수 있다.

가상그리드는 레벨을 가지게 되는데 그림 6과 같이

전체 맵(Universe)상태를 레벨 0이라고 정의하고 레벨이 증가 할수록 셀(Cell)의 개수가 증가한다. 레벨 0에서 사용자가 움직인 것, 즉, 한 개의 셀 안에서의 이동은 위치 이동으로 보지 않고 질의가 발생하지 않지만 가상그리드의 레벨이 1일 경우 현재 셀을 벗어났을 때 위치이동으로 간주하여 질의를 발생시킨다. CS 동적 속성의 경우 가상그리드를 사용함으로써 질의 발생 빈도를 낮출 수 있기 때문에 본 논문에서는 가상그리드를 사용하는 CS 동적 속성만을 사용하였다.

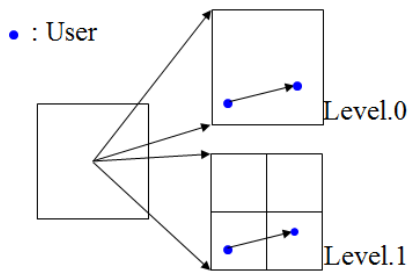


그림 6. 각 레벨에서 위치 이동에 따른 차이점
Fig. 6 Difference of location transition at each level

그림 7은 3.2절의 시스템 설계에서 언급된 문제점이 가상그리드를 통해 해결된 모습을 보인다.

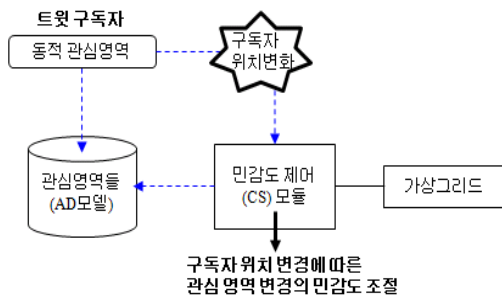


그림 7. 가상그리드를 통한 구독자 위치변화에 따른 질의 빈도수 감소
Fig. 7 Reducing of frequency of query using Virtual Grid

4.2. 가상그리드의 성능 평가

이번 장에서는 가상그리드를 사용함으로써 감소하는 질의 발생 빈도수에 대한 실험결과 이를 통한 성능개선에 대한 내용을 다룬다.

실험은 고정된 시간 30분간 가상의 이동 객체들(사람, 자전거, 자동차)을 생성 하여 각각 객체들의 속력을 다르게 정의하고 일정한 방향으로 이동시켜 질의 발생 빈도수가 어떻게 달라지는지 성능 평가 하였다.

사람의 속력은 1m/s, 자전거의 속력은 5m/s 그리고 자동차의 속력은 27m/s로 가정하였고 동쪽방향을 향해 일직선으로 30분간 이동시켰다. 레벨에 따른 성능차이도 고려하기 위해 각각 레벨16, 17, 18, 19 별로 총 4번씩의 실험을 하였다.

표 2. 질의 발생 빈도수
Table. 2 Frequency of query

Level	속력(m/s)			VG 비사용
	1	5	27	
16	4	19	60	60 (MAX)
17	7	36	60	
18	14	60	60	
19	30	60	60	

표 2에서 가상그리드를 사용하지 않았을 때의 질의 발생 빈도수가 60회를 나타낸다. 즉, 30분간 어떤 객체를 어떤 속력으로 이동시키는 것과는 관계없이 최대 질의 발생 빈도수는 60이라는 것이다. 이 수치를 기준으로 사람의 경우(1m/s)에 질의 발생 빈도수가 최대 93% ~ 최소 50%까지 감소하였다. 자전거의 경우(5m/s)는 질의 발생 빈도수가 최대 68% ~ 최소 0%까지 감소하였다. 마지막으로 자동차의 경우(27m/s)는 모든 레벨에서 감소율이 없었다.

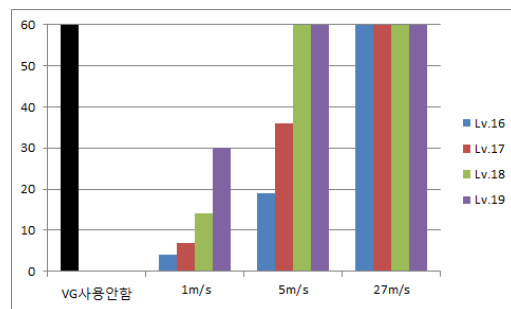


그림 8. 30분간 다른 속력으로 이동하는 객체들의 레벨 별 질의 발생 빈도수
Fig. 8 Frequency of query over three moving objects at each level

그림 8을 통해 사람의 경우 가상그리드를 사용하지 않았을 때와 비교하여 질의 발생 빈도수가 확연히 감소하였음을 볼 수 있고 자전거의 경우 각 레벨 별로 조금의 차이를 보인다. 마지막으로 자동차의 경우 모든 레벨에서 모두 가상그리드를 사용하지 않았을 때와 차이가 없음을 보인다.

V. 결 론

본 논문에서는 위치기반서비스를 사용하는 SNS에 대해 설명하고 그 중 대표적인 애플리케이션인 트위터가 가지는 위치기반서비스의 문제점을 언급하였고 그것을 해결하기 위해서 공간필터링 기법을 제안하였다.

본 시스템에서는 업데이트와 같은 복잡한 질의를 처리하기 위해 실체화 된 뷰를 사용하였으며 실체화 된 뷰가 업데이트가 자주 일어나지 않는 환경에서 최적인 것을 고려하여 인덱스를 사용하지 않는 공간조인인 PBSM을 사용하였다. 시스템의 전체적인 설계에 대해 설명하고 LBSNS에서 발생하는 대량의 연속 질의에 대한 문제점, 예를 들어, TS-AD모델에서 구독자의 계속적인 위치 이동에 따른 문제점을 해결하기 위해 가상그리드를 소개하였다. 성능평가를 통해 사람의 경우 레벨에 관계없이 질의 발생 빈도수가 감소하였으나 자전거와 자동차의 경우는 레벨 별로 질의 발생 빈도수가 차이가 있음을 보였다. 특히 자동차의 경우는 빠른 속력의 원인으로 질의 발생 빈도수를 감소시키기 위해서는 레벨을 더욱 낮게 해야 함을 알 수 있다. 그러나 레벨을 낮추게 되는 만큼 정확도가 떨어지게 된다. 따라서 자동차와 같은 빠른 이동수단을 이용할 때는 가상그리드를 사용하는 것이 좋다고 할 수 없다.

참고문헌

[1] Edwin H. Jacox, Hanan Samet "Spatial Join technique" ACM Transactions on Database Systems, 2007

[2] Jignesh M. Patel, David J. Dewitt "Partition Based Spatial-Merge Join" ACM SIGMOD 1996

[3] Ming-Ling Lo, China V. Ravishankar "Spatial hash-joins" ACM SIGMOD 1996

[4] http://ihelpers.x2soft.co.kr/programming/reference/01spring_materialized_view.pdf

[5] Jose A Blakeley, Per-Ake Larson, Frank Wm Tompa "Efficiently Updating Materialized Views" ACM SIGMOD 1986

[6] Ming-Ling Lo, China V. Ravishankar "Spatial joins using Seeded Trees" ACM SIGMOD 1994

[7] <http://develpoper.android.com/index>

[8] <http://code.google.com/intl/ko-KR/android/add-ons/google-appiis/reference/index.html>

저자소개



이은식(Eun-Sik Lee)

동서대학교
컴퓨터정보공학부 학사
연세대학교
컴퓨터과학과 석사과정

※ 관심분야: 공간조인, LBSNS, 간격 질의처리 및 색인, Stabbing 질의처리

조대수(Dae-Soo Cho)

한국정보통신학회논문지
제14권 제11호 참조