
LBSNS를 위한 Virtual Grid 및 필터링기법의 설계 및 구현

이은식 · 조대수

동서대학교

Design and Implementation of Virtual Grid and Filtering Technique for LBSNS

Eun-Sik Lee · Dae-Soo Cho

Dongseo University

E-mail : crubeido@nate.com , dscho@dongseo.ac.kr

요 약

기존의 SNS(Social Networking Service)서비스에 LBS(Location-Based Service)서비스가 추가된 LBSNS(Location-Based Social Networking Service)서비스들이 상용화되면서 큰 인기를 얻고 있다. 트위터는 그러한 서비스의 대표적인 예라고 볼 수 있다. 트위터의 현재 위치기반서비스는 자신이 원하는 지역정보와 상관없는 정보를 구독하게 하는 구조로 되어 있다. 팔로잉한 사용자는 단순히 개인적인 선호도에 의해 지역정보가 추가된 메시지를 트윗하지만 구독하는 입장의 팔로워는 자신이 원하지 않는 지역정보를 받아 볼 수도 있다. 이러한 상황을 개선하기 위해 공간조인을 이용한 필터링 기법이 제안되었다.

필터링 기법을 위한 우선적인 작업은 바로 각각의 사용자와 트윗들에 위치정보가 추가되어야 한다. 여기서 위치정보는 MBR(Minimum Bounding Rectangle)로 표현된다. 위치정보는 동적속성 또는 정적속성으로 나누어진다. 동적인 경우를 예를 들어보면 사용자가 지속적으로 움직이는 상황을 들 수 있다. 이 때 발생하는 대량의 연속질의는 사용자가 많은 SNS의 특성상 서버에 많은 부하를 줄 수 있다. 본 논문에서는 구글 맵 상에서 Virtual Grid를 생성하여 문제를 해결 하였고 성능 평가 결과 Virtual Grid를 사용하지 않았을 때 보다 질의 발생 빈도수가 줄어들었다.

ABSTRACT

The LBSNS(Location-Based Social Networking Service) service has been well-received by researchers and end-users, such as Twitter. Location-Based service of Twitter is now structured that users could not subscribe the information of their interesting local area. Those who being following from someone tweet message included information of local area to them just for their own interesting. However, follower may receive that kind of tweet. In order to handle the problem, we propose filtering technique using spatial join.

The first work for filtering technique is to add a location information to tweets and users. In this paper, location information is represented by MBR(Minimum Bounding Rectangle). Location information is divided into dynamic property and static property. Suppose that users are continuously moving, that means one of the dynamic property's example. At this time, a massive continuous query could cause the problem in server. In this paper, we create Virtual Grid on Google Map for reducing frequency of query, and conclude that it is useful for server.

키워드

필터링 기법, LBSNS(Location-Based Social Networking Service), 공간조인, Virtual Grid

I. 서론

최근 무선통신, GPS, 스마트 폰의 기술발전과 함께 LBSNS(Social Networking Service)라고 불리는 위치기반서비스가 인기를 얻고 있다. 이와 함께 수많은 LBSNS서비스를 제공하는 업체들이 나오고 있으며 기존 SNS서비스를 제공하는 업체들 또한 그 뒤를 따라 자신들의 서비스에 LBS서비스를 추가하고 있다. 트위터가 그러한 서비스의 대표적인 예라고 할 수 있다. 일반적인 트위터에서의 구독조건은 본인이 다른 사용자에게 관심이 있다면 그 사용자의 팔로워가 되어 그 사람의 트윗을 구독 하는 시스템이었다. 하지만 LBS라는 서비스가 추가 되면서 구독조건을 위한 관심의 대상이 사람에서 위치로 바뀌어 질 수 있다. 즉, 내가 관심 있는 지역을 관심 있어 하는 사람을 구독조건으로 볼 수 있다.

현재 트위터에서 제공하는 위치기반서비스에는 관심지역에 대한 검색 서비스가 있다. #태그를 이용한 지역검색의 경우 "#Busan"이라고 검색을 했을 때 검색된 트윗들이 실제로 Busan지역과 관련이 있는지 없는지 알 수 없는 한계가 있다. 연산자(operator)를 통한 고급 검색도 제공하는데 이 또한 일반 사용자가 생소한 공간 연산자를 쓰기에는 불편한 문제점이 있다. 검색 서비스 외 실제로 트윗에 자신의 위치정보를 추가하여 트윗할 수 있는 기능이 있다. 팔로워들은 트윗을 구독하여 위치정보를 받아 볼 수 있음과 동시에 자신이 원하는 지역정보와 무관한 정보를 받을 수도 있다. 트위터의 기본적인 구조는 일방적인 구독 형태이기 때문에 상대방을 팔로우 했다면 무조건 해당 정보를 구독해야 한다. 이러한 점을 개선하기 위해 내가 원하는 지역의 정보만을 구독할 수 있는 공간조인[2][3]을 이용한 필터링 기법이 제안되어졌다.[1]

필터링 기법을 위한 최우선 작업은 각각의 사용자와 트윗들에 위치정보가 추가되어야 한다. 여기서 위치정보는 MBR(Minimum Bounding Rectangle)로 표현되고 동적속성과 정적속성으로 나누어진다. 동적속성인 경우의 예를 들어보면 지속적으로 움직이는 사용자를 예로 들 수 있다. 그와 동시에 발생하는 대량의 연속질되는 사용자가 많은 SNS의 특성상 서버에 큰 부담을 줄 수 있다.

이 논문에서는 질의 발생 빈도수를 낮추기 위한 구글 맵 상에서 Virtual Grid[1]를 생성하는 방법에 대해 상세히 기술한다. 가상으로 이동객체(Moving Object)들(사람, 자전거, 자동차)을 생성하여 각각의 속력을 다르게 한 뒤 Virtual Grid 상에서 일정한 방향으로 이동시켜 지정된 시간 내 질의 발생 빈도수가 어떻게 달라지는지 성능 평가하고 비교 분석 한다.

II. 관련 연구

2.1 필터링 기법(Filtering Technique)

필터링 기법[1]은 그림 1와 같이 자신이 원하지 않는 지역정보의 구독을 필터링 하여 받아 볼 수 있게 해주는 방법이다.

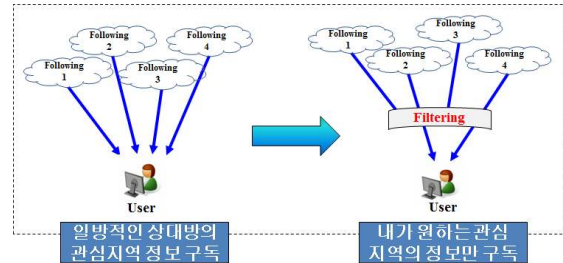


그림 1. 필터링 기법

필터링 기법을 위한 전체적인 시스템 구조는 그림 2와 같다.

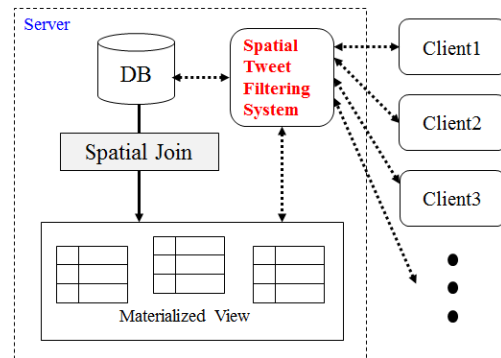


그림 2. 시스템 구조

각 클라이언트들로부터 발생되는 질의들은 서버내의 STFS(Spatial Tweet Filtering System)에 의해 모두 처리가 된다. 트윗들과 사용자들의 위치정보를 이용하여 주기적인 공간조인을 수행하여 각각 DB에 저장하고 타임라인과 같은 복잡한 질의에 관한 것은 실체화 된 뷰에 저장하고 이를 사용자의 요구가 있을 때 마다 보내준다. 이때 공간조인은 인덱스가 없는 PBSM(Partition-Based Spatial Merge Join)[3]을 사용한다.

필터링 기법과 시스템 구조에 관한 자세한 사항은 [1]에 자세하게 나와 있으니 참조하기 바란다.

III. Virtual Grid

3.1 위치정보의 추가

시스템 설계를 위해 가장 먼저 할 작업은 사용자와 트윗들에 각각 위치정보(MBR)를 추가하

는 것 이다. 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. 본 논문에서는 동적 속성을 그림 4와 같이 Sensitive Dynamic과 Non-Sensitive Dynamic 2가지로 구분하였다.

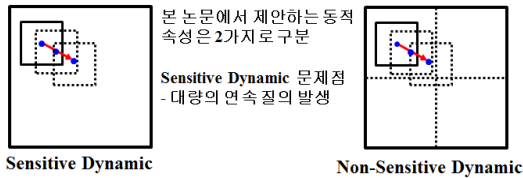


그림 3. 동적속성의 2가지 구분

먼저 Sensitive Dynamic의 경우 사용자의 위치 정보가 조금이라도 움직이게 될 때 마다 그와 함께 질의가 발생된다. 일정한 주기(예를 들어, 10초, 20초)에 한번씩 질의가 발생되더라도 마찬가지 일 것이다. 이러한 점을 개선하기 위해 Non-Sensitive Dynamic을 제안하였다. Sensitive Dynamic과 달리 Non-Sensitive Dynamic에서는 VG(Virtual Grid)를 사용하여 전체 MBR을 Grid 형태로 나누게 된다. VG는 레벨을 가지게 된다. 그림 5와 같이 기본 MBR을 레벨 0이라고 정의하고 레벨이 증가 할수록 셀(Cell)의 개수가 증가한다. 레벨 0에서 사용자가 움직인 것, 즉, 한 개의 셀 안에서 움직인 것은 위치이동으로 보지 않고 질의가 발생하지 않지만 VG의 레벨이 1일 경우 셀을 벗어났을 때 위치이동으로 간주 하여 질의를 발생시킨다. Non-Sensitive Dynamic의 경우 VG를 사용함으로써 질의 발생 빈도를 낮출 수 있기 때문에 본 논문에서는 Non-Sensitive Dynamic을 사용하였다.

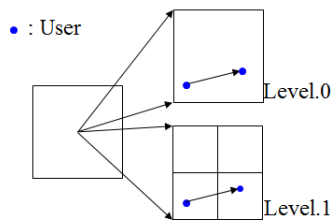


그림 4. 각 레벨에서 위치 이동에 따른 차이점

3.2 Virtual Grid 생성 절차

Virtual Grid 생성을 위한 첫 번째 작업은 맵의 크기와 레벨을 선택하는 것이다. 본 논문의 Virtual Grid는 구글 맵[4][5]을 대상으로 생성되어졌다. 생성에 필요한 매개변수는 맵 크기(right, left, top, bottom)와 레벨이다. 두 번째 작업은 측면 셀들의 개수, 해당 셀들의 높이와 길이, 맵 전체의 높이를 계산한다. 세 번째 작업은 원본 맵의

left, top 좌표 값을 (0, 0)으로 전체를 이동시켜 변환 시킨다. 네 번째 작업은 이동 된 새로운 맵을 그림 5와 같이 부호화 시킨다.

3.3 Virtual Grid를 사용한 위치이동 검사

Virtual Grid를 사용하는 주 목적은 가장 최근 좌표를 포함하는 셀에서 다음검사에서 현재 좌표가 벗어났는지를 검사 하는 것이다. 단순한 좌표들의 비교로는 셀에서의 벗어남 유무를 확인할 수 없다. 따라서 이것을 위해 3.2절에서 생성한 Virtual Grid를 이용한다. 현재좌표를 (0, 0)좌표를 기준으로 변환 된 맵에 사상시켜 새로운 좌표를 얻는다. 새로운 좌표를 다시 부호화 된 맵에 사상시켜 최종적으로 부호화 된 값을 얻을 수 있다.

그림 5에서와 같이 새로운 좌표(220, 120)을 다시 부호화 된 맵에 사상시켜 [3, 1]이라는 부호화 값을 얻을 수 있다.

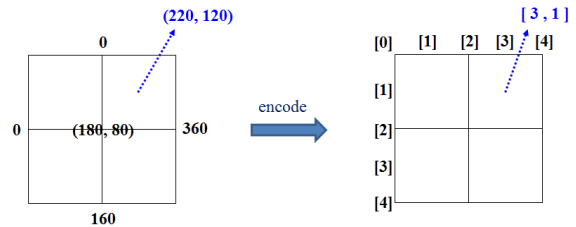


그림 5. 맵과 좌표의 부호화

IV. 성능 평가

이번 장에서는 Virtual Grid를 사용함으로써 감소하는 질의 발생 빈도수에 대한 실험과 이를 통한 성능 평가내용을 다룬다.

실험은 고정된 시간 30분간 가상의 이동 객체(Moving Objects)들(사람, 자전거, 자동차)를 생성하여 각각 객체들의 속력을 다르게 주고 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 보기위해 각각 16 ~ 19 총 4번의 실험을 하였다.

표 1. 30분간 발생된 질의 발생 빈도수

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

표 1에서 Virtual Grid를 사용하지 않았을 때의

질의 발생 빈도수는 60회다. 즉, 30분간 어떤 객체를 어떤 속력으로 이동시키는 것에 관계없이 최대 질의 발생 빈도수는 60이라는 것이다. 이 수치를 기준으로 사람의 경우(1m/s) 질의 발생 빈도수가 최대 93% ~ 최소 50% 까지 감소하였다. 자전거의 경우(5m/s)는 질의 발생 빈도수가 최대 68% ~ 최소 0% 까지 감소하였다. 자동차의 경우(27m/s)는 모든 레벨에서 감소율이 없었다.

그림 6을 통해 사람의 경우 VG를 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 VG를 사용하지 않았을 때와 차이가 없음을 보인다.

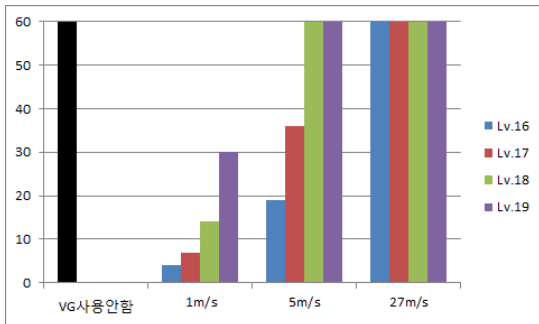


그림 6. 30분간 다른 속력으로 이동하는 객체들의 레벨 별 질의 발생 빈도수

V. 결 론

본 논문에서는 필터링 기법을 사용하기 위해 필요한 작업들에 대해 설명하였고 작업에서 발생 되는 대량의 연속 질의에 대한 문제점을 지적하였다. 문제점을 해결하기 위해 Virtual Grid를 소개 하였고 구현을 통해 Virtual Grid를 사용하지 않았을 때와의 차이를 성능평가 하였다.

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. 하지만 레벨을 낮추게 되면 질의 발생 빈도수는 줄어들지만 그만큼 정확도는 떨어지게 된다.

따라서 자동차와 같은 빠른 이동수단을 이용할 때는 Virtual Grid를 사용하는 것이 좋다고 할 수 없을 것이다.

참고문헌

- [1] 이은식, 조대수, "LBSNS에서의 공간조인을 이용한 필터링 기법의 설계", 해양정보학회 춘계 학술 대회, 2011.05

- [2] Edwin H. Jacox , Hanan Samet "Spatial Join technique" ACM Transactions on Database Systems, 2007
- [3] Jignesh M. Patel , David J. Dewitt "Partition Based Spatial-Merge Join" ACM SIGMOD Record, 1996
- [4] <http://developer.android.com/index>
- [5] <http://code.google.com/intl/ko-KR/android/add-ons/google-apiis/reference/index.html>