블룸 필터를 이용한 그리드 기반 공간-텍스트 색인 방법

[1] 박상덕, [2] Nyamdavaa Ariunerdene, [3] 고대식, [4*]송석일
[1] 한국교통대학교, 컴퓨터정보기술공학부, 석사과정, deok31513@a.ut.ac.kr
[2] 한국교통대학교, 컴퓨터정보기술공학부, 박사과정, ariunkk@gmail.com
[3] 목원대학교, 전자공학과, 교수, kds@mokwon.ac.kr
[4*] 한국교통대학교, 컴퓨터정보기술공학부, 교수, sisong@ut.ac.kr

Grid based Spatio-Textual Indexing Method using Bloom Filter

[1] Sang-Deok Park, [2]Nyamdavaa Ariunerdene, [3]Daesik Ko, [4*]Seok-Il Song
[1]School of Computer Engineering & Information Technology , Korea National University of Transportation, master's course, deok31513@a.ut.ac.kr
[2]School of Computer Engineering & Information Technology , Korea National University of Transportation, Ph.d Candidate, ariunkk@gmail.com
[3]Department of Electronic Engineering, Mokwon University, Professor, kds@mokwon.ac.kr
[4*]School of Computer Engineering & Information Technology , Korea National University of Transportation, Professor, sisong@ut.ac.kr

요 약

이 논문에서는 효과적인 부울(Booolean) 범위 공간-텍스트 질의 처리를 위해 그리드 기법을 이용하여 공간 색인을 수행하고 각 그리드 셀에 포함된 객체의 텍스트 속성에 대해 블룸 필터(Bloom Filter)를 이용해 가지치기(Prunning)을 수행하는 방법을 제안한다. 그리드 공간 색인 및 블룸 필터는 색인을 유지하기 위한 공간의 크기가 매우 작은 특징을 갖기 때문에 인-메모리 색인 처리에 매우 적합하다. 이 논문에서 제안하는 색인 구조는 힐버트 커브를 기반으로 공간-텍스트 질의에 대한 그리드 공간 색인의 셀을 선택하고 선택된 각 셀들에 대해 질의의 텍스트를 포함하는지 여부를 블룸 필터를 이용하여 판별하여 질의 처리를 수행한다. 마지막으로, 실험을 통해서 기존 그리드 공간 색인 기법에 이 논문에서 제안하는 방법을 적용하여 부울 범위 공간-텍스트 질의 처리의 효율성을 얼마나 높일 수 있는지 보인다.

Abstract

In this paper, we propose a spatio-textual using grid techniques for spatial indexing and Bloom Filter for filtering text for efficient boolean range spatio-textual query processing. Grid space indexing and bloom filters are well suited for in-memory indexing because of the very small size of space to maintain the index. The index structure proposed in this paper selects the cells of the grid index for the spatial range queries based on the Hilbert curve, and determines whether the text of the query is included for each of the selected cells using a bloom filter to perform the query processing. Our experiments show how to improve the efficiency of boolean range spatio-textual query processing by applying the method proposed in this paper to the existing grid indexing technique.

Keywords: Spatio-textual indexing, Bloom Filter, Grid Index, Boolean Range Query, Database

* Corresponding Author
Received: Dec. 21, 2019, Revised: Dec. 26, 2019, Accepted: Dec. 31, 2019

I. 서론

GPS 가 장착된 모바일 기기의 사용이 확산되면서 모바일기기가 생산하는 위치 데이터를 이용한 위치기반 서비스가 점차 확산되어 왔다. 현재는 모바일 기기 사용자가 주변의 상점, 관광지 등을 검색하고 해당 목적지까지 경로를 파악하는 등 우리의 일상에서 다양한 종류의 위치기반 서비스가 상용화 되었고 사용자가 점차 늘어나고 있다. 위치 데이터 및 이를 이용한 위치기반 서비스가 확산되면서 어떤 콘텐츠의 지리적 공간상의 위치와 그에 대한 텍스트 설명이 결합된 공간-텍스트(Spatio-Textual) 데이터가 증가하였다[1].

공간-텍스트 데이터는 다양한 형태로 존재한다. 예를 들면 POI(Points of Interest), 대중교통, SNS(Social Network Service) 메시지 등에서 공간-텍스트 데이터가 존재한다. POI 의 경우 주유소는 위치와 함께 유가, 주소 등의 텍스트 데이터가 같이 존재한다. SNS 의 경우에도 SNS 메시지는 모바일 기기의 현재 위치가 같이 결합될 수 있다. 이러한 데이터를 지오태그(Geo-tagged) 데이터라고 한다. SNS 의 지오태그 데이터는 추가로 이미지를 같이 포함할 수도 있다.

공간-텍스트 데이터는 두가지로 구분될 수 있다[1]. 첫번째는 데이터의 위치와 텍스트가 자주 변경되지 않는 정적 공간-텍스트 데이터이다. 예를 들면, 상점이나 식당과 같이 정적인 공간 객체와 그에 대한 텍스트가 결합된 데이터는 변경이 빈번하지 않다. 두번째는 데이터의 위치나 텍스트가 빈번하게 변경되는 동적 공간-텍스트 데이터이다. 대표적으로 이동객체의 경우 위치가 지속적으로 변경하며 이러한 동적 공간-텍스트 데이터는 스트림 형태로 존재한다.

일반적으로 공간데이터에 대한 질의는 특정 영역의 객체를 검색하는 범위 질의(Range Query), 특정 객체에서 가장 가까운 k 개의 객체를 찾는 k 최근접 질의(k Nearest Neighbor Query, kNN Query), 질의 객체를 k 최근접 객체로 하는 객체들을 검색하는 역 k 최근접 질의(Reverse k Nearest Neighbor Query, RkNN Query) 가 있다. 텍스트 질의의 경우 특정 키워드 집합을 모두 포함하는 객체를 찾는 부울 질의(Boolean Query)와 키워드 집합과 유사도를 측정하여 순위에 따라 객체를 검색하는 순위 질의(Ranked Query)로 구분될 수 있다. 공간-텍스트 질의는 3 가지의 공간질의와 2 가지의 텍스트질의의 조합으로 구분된다. 예를 들어서 부울 범위 질의(Boolean Range Query)는 범위, 질의 위치, 키워드 집합으로 구성되며 질의 위치로부터 범위 안에 있는 객체들 중 키워드집합을 모두 만족하는 객체들을 검색하게 된다.

이와 같은 공간-텍스트 질의를 처리하기 위해서는 기존에 제안된 공간 색인이나 텍스트 색인이 아닌 공간 속성과 텍스트 속성 모두를 이용하여 가지치기(Pruning)이 가능한 새로운 형태의 색인이 필요하다. 기존에 다양한 공간-텍스트 질의를 위한 색인 기법들이 제안된 바 있다 [2-9]. 기존에 제안된 방법들은 크게 공간 우선 색인과 텍스트 우선 색인, 공간과 텍스트를 동시에 고려하는 색인 방법들로 구분할 수 있다. 각 방법들은 색인 구축시 어떤 속성을 우선으로 하는지에 따라 결정된다. 다음 표 1 은 기존에 제안된 공간-텍스트 색인 방법들을 공간우선, 텍스트우선, 공간 및 텍스트 동시 고려 그룹으로 구분한 것이다.

Table 1. Classification of existing spatio-textual indexing methods [1]

구분	색인 방법
공간 우선	IR-트리[2], IR^2-트리[4]
텍스트 우선	S2I[3], GeoWAND[5]
공간 및 텍스트 동시 고려	DIR-트리[3], CDIR-트리[3], BR-트리[6], AP-트리[7]

공간 우선 색인 기법의 IR-트리[2]는 기본적으로 R-트리의 구조를 가지며 각 노드는 서브 트리에 포함된 객체들의 키워드들로 구성되는 역파일(Inverted File)에 대한 참조를 부가적으로 가진다. IR-트리를 기반으로 몇몇 변형들이 제안되었다. DIR-트리[3]는 인덱스 구축시 공간 및 텍스트 속성을 모두 이용하여 IR-트리의 MBR 을 최소화하고 노드에 포함된 키워드 들의 유사성을 최대화하는 방법을 제안하고 있다. CDIR-트리[3] 은 DIR-트리의 각 노드의 객체들을 텍스트를 기반으로 클러스터링 하여 텍스트 유사성을 높이는 방법을 제안하고 있다. IR^2-트리[4]는 R-트리와 시그니처 파일(Signature File)을 통합하는 방법을 제안하고 있다. R-트리의

각 노드에 포함된 객체들의 키워드들의 시그니처를 모아서 노드에 같이 저장한다. R-트리의 각 레벨에 최적의 시그니처 길이를 계산하여 거짓 양성(False Positive)을 줄일 수 있는 방법을 포함하고 있다.

텍스 우선 색인 기법으로 S2I(Spatial Inverted Index)[3]가 제안된 바 있다. S2I 는 빈번한(Frequent) 키워드는 aR-트리(augmented R-트리)에 저장하고 빈번하지 않은(Infrquent) 키워드는 역파일에 저장한다. GeoWAND[5]는 인-메모리(In-memory) 색인으로 역인덱스(Inverted Index)에 공간 정보를 결합한 색인 방법이다.

BR-트리[6]는 B-트리, R-트리, 역리스트(Inverted List)를 결합한 색인구조로 공간 및 텍스트 동시에 고려하는 색인 방법이다. BR-트리에서는 SBR 알고리즘과 KBR 알고리즘을 제안하고 있다. SBR 알고리즘은 B-트리와 역리스트를 이용하여 텍스트 우선 질의처리를 수행하는 알고리즘이며 KBR 알고리즘은 R-트리를 이용하여 공간 우선 질의처리를 수행하는 알고리즘이다. AP-트리[7]역시 공간 및 텍스트를 동시에 고려하는 색인 방법이다. 이 방법에서는 트리의 각 노드에서 노드에 적합한 분할 방법(텍스트 또는 공간)을 선택하여 객체들을 분할한다.

기존에 제안된 대부분의 방법들의 공통적인 문제는 공간뿐 아니라 텍스트를 고려하면 색인의 크기가 커진다는 것이다. 색인의 크기가 커지면 이에 따라 입출력 시간이 증가하게 되어 질의처리 시간이 길어지게 된다. GeoWAND 에서는 이러한 문제를 완화하기 위한 방법을 제안하고 있지만, 기본적으로 역인덱스를 사용하고 있고 여기에 공간정보가 더해지므로 색인의 크기를 줄이는데 한계가 있다.

이 논문에서는 힐버트 커브(Hilbert Curve)[8] 기반의 그리드 기법을 이용하여 공간 색인을 수행하고 각 그리드 셀에 포함된 객체의 텍스트 속성에 대해 블룸 필터(Bloom Filter)[9]를 이용해 가지치기(Prunning)을 수행하는 방법을 제안한다. 이 논문에서 지원하는 질의는 부울 범위 공간-텍스트 질의이다. 힐버트 커브기반 그리드 기법은 그리드 크기만 있으면 색인이 가능하다. 블룸 필터는 해시(Hash) 기법을 이용하여 특정 집합에 어떤 객체가 포함되는지 여부를 빠르게 판별하는 확률적 데이터 구조이다. 블룸 필터는 작은 크기의 비트맵(Bitmap)을 필터로 유지한다. 블룸 필터의 비트맵의 크기는 거짓 양성(False Positive) 발생 비율, 최대 객체 수에 따라 결정되며 거짓 양성 발생 비율이 작아지고 객체 수가 커질수록 비트맵의 크기는 커질 수 있다.

이 논문에서 제안하는 색인 구조는 힐버트 커브를 기반으로 공간-텍스트 질의에 대한 그리드 공간 색인의 셀을 선택하고 선택된 각 셀들에 대해 질의의 텍스트를 포함하는지 여부를 블룸 필터를 이용하여 판별하여 가지치기를 수행한다. 또한, 제안하는 색인구조의 크기는 매우 작으며 항상 인-메모리에 유지하는 것이 가능하여 빠른 질의처리가 가능한 장점이 있다. 이 논문의 구성은 다음과 같다. 2 장에서는 제안하는 블룸 필터 기반의 그리드 공간-텍스트 색인 방법을 자세하게 설명하고 3 장에서는 실험을 통해 블룸 필터가 공간-텍스트 질의의 효율을 얼마나 향상시킬 수 있는지 보인다. 마지막으로 4 장에서 결론을 맺는다.

II. 제안하는 블룸 필터 및 그리드 기반 공간-텍스트 색인 방법

이 논문에서 제안하는 공간-텍스트 색인구조는 힐버트 커브를 이용한 그리드 기법과 블룸 필터를 이용한다. 그리드 기법을 이용하여 객체들에 대한 공간 색인을 수행하고 각 그리드 셀에 포함된 객체들의 텍스트 들에 대해 블룸 필터를 생성한다. 그리드 셀 별로 생성된 블룸 필터들은 블룸 필터 테이블에 저장된다.

그림 1 은 이 논문에서 제안하는 색인의 전체적인 구조를 보여준다. 그림에서 보는 바와 같이 각 객체들은 위치 속성에 따라 그리드 상에서 각 셀별로 분할 된다. 각 셀에 포함된 객체들의 텍스트들에 대해서 블룸 필터를 생성하고 셀별 블룸 필터를 블룸 필터 테이블에 저장 관리한다. 잘 알려진 것처럼 블룸 필터의 크기는 거짓 양성 확률과 최대 객체 수에 의해서 결정된다. 수식 1 은 블룸 필터를 위한 비트 수 m 을 계산하는 식이다. 여기에서 n 은 예상되는 최대 레코드 수이고, p 는 거짓 양성 확률이다.

$$m = \frac{n \cdot log_p}{\log(\frac{1}{2^{log_2}})} \quad (1)$$

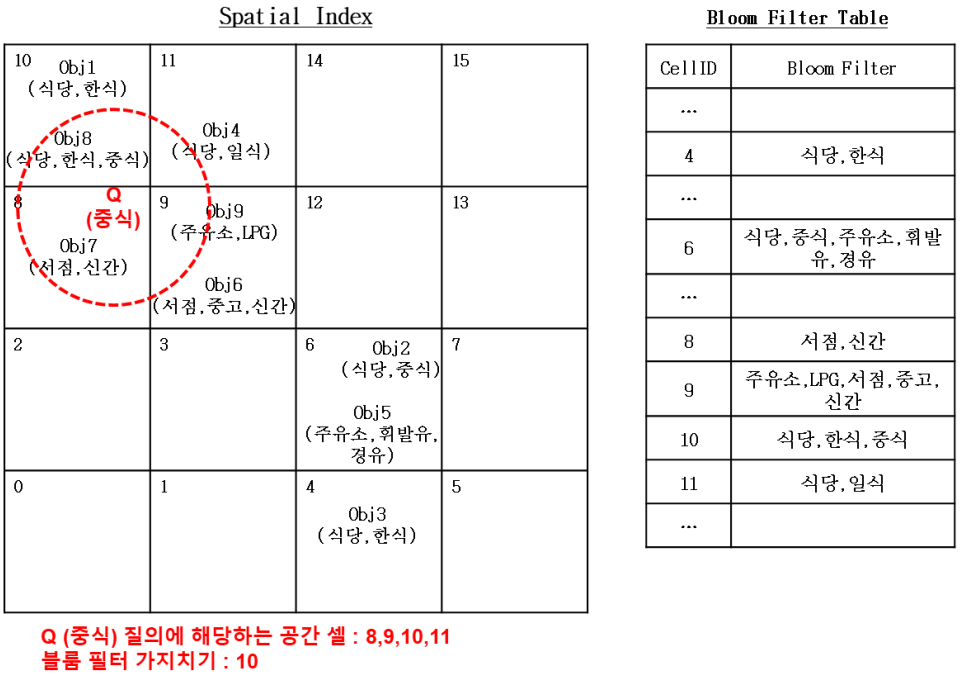

Figure 1. Proposed grid based spatio-textual index structure using Bloom Filter

제안하는 색인구조는 셀별로 블룸 필터를 생성하므로 블룸 필터 전체의 크기는 다음 수식 2 로 계산할 수 있다. 이 수식에서 c 는 셀의 수, cn 은 셀별 최대 객체수, k 는 객체가 편향되는 것을 고려하여 각 셀별 객체 수를 일정 하게 증가시키기 위한 1 보다 큰 값이다. $total\_m$ 은 셀별 블룸 필터 크기를 합한 값이다. 표 2 는 전체 레코드 수를 100,000,000, 거짓 양성 확률 p 를 0.001, 셀 수 c 를 1,024, k 를 1~5 로 증가시킬 때 전체 블룸 필터의 크기를 보여주는 것이다. 이 표에서 보는 것과 같이 전체 레코드 수가 1 억개, 셀별 객체의 수를 원래의 2 배로 할 때 전체 블룸 필터의 크기는 약 368MB 이다. 1 억개 레코드 수를 감안할 때 이는 충분히 작은 크기라고 판단되며 인-메모리 처리에 있어 문제가 없다고 보인다.

$$total\_m = \sum_{i=1}^{c} \frac{cn \cdot log_p}{\log(\frac{1}{2^{log_2}})} \quad , cn = \frac{n \cdot k}{c} \quad (2)$$

Table 2. Total size of Bloom Filter

k	$total\_m$ (MB)
1	184.033152
2	368.066304
3	552.099456
4	736.132608
5	920.165632

질의 처리는 다음과 같은 순서로 진행된다. 이 논문에서는 부울 범위 공간-텍스트 질의만을 고려한다. 부울 범위 공간-텍스트 질의는 먼저 공간 색인을 통해 질의에 부합하는 셀을 선택한다. 공간 색인을 통해서 선택된 셀들의 블룸 필터들에 대해서 텍스트가 포함되어 있는지 확인하여 최종 셀들을 선택한다. 최종적으로 선택된 셀들에 대해서 저장소에서 읽어낸 후 질의의 범위 및 텍스트를 비교한 후 최종 질의 결과를 반환한다.

예를 들어서 질의 처리 과정을 설명한다. 그림 1 에서 부울 공간-텍스트 범위 질의 Q 의 범위는 점선으로 된 원이며 텍스트는 “중식”이다. 질의의 범위에 겹치는 셀들은 8, 9, 10, 11 이다. 이어서 셀 8, 9, 10, 11 의 각 블룸 필터에 대해서 “중식”이 포함되어 있는지 확인한다. 거짓 양성이 발생하지 않는다는 가정하에 최종 선택된 셀은 10 번이 된다. 셀 10 에 해당하는 모든 데이터를 저장소로부터 읽어 들인 후 최종 비교를 통해서 결과를 반환한다.

III. 성능 평가

이 논문에서는 제안하는 색인 방법의 효율성을 측정하기 위해서 실험을 진행하였다. 실험의 주요 내용은 기존 그리드 기반 공간색인 방법[10]에 블룸 필터 기반의 텍스트 가지치기 기법을 결합할 때 어느 정도의 성능이 개선되는지 확인하는 것이다. 기존 제안된 그리드 기반 공간 색인 방법은 분산 병렬환경에서 Apache Accumulo[11]를 기반으로 구현되었다. 이 방법은 Apache Accumulo 의 테이블 분할 특성을 이용하여 데이터 삽입 및 질의처리에 대한 병렬성을 최대화 한다. 이 논문에서는 기존 제안된 그리드 색인 방법에서 셀별로 블룸 필터를 생성하고 관리하는 기능을 추가하여 공간-텍스트 질의 처리가 가능하도록 구현하였다.

이 논문에서 사용한 실험 환경은 표 3 과 같다. 전체 레코드 수는 1 억개로 하고 실험에 사용한 질의의 수는 1 만개로 하였다. 각 질의는 선택도를 1% ~ 5% 로 하였다. 노드의 수는 3 개와 9 개로 하여 실험을 수행하였고 매 실험시 마다 질의의 평균 응답시간과 질의를 통해서 선택된 셀 수를 측정하여 비교하였다.

Table 3. Experiment environment

구분		내용
H/W	CPU	Intel(R) Xeon(R) CPU E5-2686 v4 @ 2.30GHz
	RAM/HDD	16G/50G
S/W	OS	Ubuntu 16.04.LTS
	Hadoop	2.8.4
	Zookeeper	3.4.10
	Apache Accumulo	1.9.1
Data	Number of records	100,000,000
	Number of Query	10,000 (selection rate : 1% ~ 5%)

이 논문의 실험결과는 표 4 와 같다. 이 표 에서처럼 노드 수와 관계 없이 기존 방법과 제안 방법의 평균 셀 선택 수는 각각 12, 16 으로 블룸 필터 기반의 텍스트 가지치기 기법을 적용했을 때 선택되는 셀의 수가 25% 감소하였다. 평균 질의 처리 시간을 측정했을 때 노드 3 개에서는 기존 방법이 6.8 초인 반면 제안 방법은 0.676 초의 성능을 보였다. 노드 수가 9 개일때는 제안 방법이 0.333 초, 기존방법은 4.126 초였다. 공통적으로 제안방법의 응답시간이 기존 방법의 응답시간에 비해 약 90% 정도 줄어드는 것을 확인할 수 있었다.

Table 4. Experimental result

노드 수 / 성능 척도	3		9	
	제안방법	기존방법	제안방법	기존방법
Responce Time(sec)	0.676	6.86	0.333	4.126
number of Cell	12	16	12	16

IV. 결론

이 논문에서는 효과적인 부울 범위 공간-텍스트 질의처리를 위한 인-메모리 블룸 필터를 이용한 그리드 기반 공간-텍스트 색인 방법을 제안하였다. 그리드 기법을 이용하여 공간 색인을 수행하고 각 그리드 셀에 포함된 객체의 텍스트 속성에 대해 블룸 필터를 생성하여 질의의 텍스트에 대한 가지치기을 수행한다. 그리드 공간 색인 및 블룸 필터의 전체 크기는 1 억개 객체, 그리드 크기를 1024 로 할 때 약 184MB 이면 되므로 공간 효율성이 매우 높다. 이 논문에서는 마지막으로, 실험을 통해서 기존 그리드 공간 색인 기법에 이 논문에서 제안하는 방법을 적용하여 부울 범위 공간-텍스트 질의처리의 효율성을 확인하였다. 기존 그리드 공간 색인 기법에 비해서 질의 결과로 선택되는 셀의 수는 25% 줄이고 질의 응답시간은 90%까지 줄일 수 있음을 확인하였다.

V. 감사의 글

본 연구는 국토교통기술촉진사업 “세계시장 진출을 위한 구글맵(Google Maps) 기반의 증강현실(AR) 적용 실내공간용 보행자 내비게이션 플랫폼 개발” 과제번호 19CTAPC15272801000000) 지원으로 수행하였습니다. 또한, 본 연구는 산림청(한국임업진흥원) 산림과학기술 연구개발 사업 ´(2017063A00_1919-AB01) ´ 의 지원에 의하여 이루어진 것입니다.

VI. 참고문헌

[1] F. Choudhury, “Efficient query processing on spatial and textual data: beyond individual queries,” Ph.D. dissertation, School of Science, RMIT University, Melbourne, VIC, Australia, 2017

[2] Z. Li, K. C. Lee, B. Zheng, W. C. Lee, D. Lee, and X. Wang, “Ir-tree: An efficient index for geographic document search,” IEEE Transactions on Knowledge and Data Engineering, Vol. 23, No. 4, pp. 585-599, April 2011.

[3] J. B. Rocha-Junior, O. Gkorgkas, S. Jonassen, and K. Nørvåg, “Efficient processing of top-k spatial keyword queries,” in Proc. of the International Symposium on Spatial and Temporal Databases, Minneapolis, MN, USA, 2011, pp. 205-222.

[4] I. De Felipe, V. Hristidis, and N. Rishe, “Keyword search on spatial databases,” in Proc. of the 2008 IEEE 24th International Conference on Data Engineering, Cancun, Mexico, 2008, pp. 656-665.

[5] J. Mackenzie, F. M. Choudhury, and J. S. Culpepper, “Efficient location-aware web search,” in Proc. of the 20th Australasian Document Computing Symposium, New York, NY, USA, Article No. 4

[6] G. Li, J. Xu, and J. Feng, “Keyword-based k-nearest neighbor search in spatial databases,” in Proc. of the 21st ACM international conference on Information and knowledge management, New York, NY, USA, 2012, pp. 2144-2148.

[7] X. Wang, Y. Zhang, W. Zhang, X. Lin, and W. Wang, “Ap-tree: Efficiently support continuous

spatial-keyword queries over stream," in Proc. of the 2015 IEEE 31st International Conference on Data Engineering, 2015, pp. 1107–1118.
[8] T. Bozkaya, M. Ozsoyoglu, "Indexing large metric spaces for similarity search queries," ACM Transactions on Database Systems (TODS), Vol. 24, no. 3, pp. 361-404, Mar. 1999.
[9] P. S. Almeida, C. Baquero, N. Preguiça, and D. Hutchison, "Scalable bloom filters," Information Processing Letters, Vol. 101, No. 6, pp. 255-261, Mar. 2007.
[10] S. Park, D. Ko, and S. Song, "Parallel insertion and indexing method for large amount of spatiotemporal data using dynamic multilevel grid technique," Applied Sciences, Vol. 9, No. 20, 4261, Oct. 2019.
[11] R. Sen, A. Farris, and P. Guerra, "Benchmarking apache accumulo bigdata distributed table store using its continuous test suite," in Proc. of the 2013 IEEE International Congress on Big Data, Santa Clara, CA, USA, 2013, pp. 334-341.