

빅데이터 환경에서 연속 질의 처리를 위한 리버스 k-최근접 질의 처리 기법

Reverse k-Nearest Neighbor Query Processing Method for Continuous Query Processing in Bigdata Environments

임종태*, 박선용*, 서기원*, 이민호**, 복경수*, 유재수*
충북대학교 정보통신공학과*, 한국과학기술정보연구원**

Jongtae Lim(jtlim@chungbuk.ac.kr)*, Sunyong Park(sypark1992@chungbuk.ac.kr)*,
Kiwon Seo(seokyle@chungbuk.ac.kr)*, Minho Lee(cokeman@kisti.re.kr)**,
Kyoungsoo Bok(ksbok@chungbuk.ac.kr)*, Jaesoo Yoo(yjs@chungbuk.ac.kr)*

요약

최근 위치 측정 기술과 모바일 기기들의 발달과 함께 위치 기반 서비스가 중요하게 연구되고 있다. 위치 기반서비스를 제공하기 위해 많은 연구자들이 맵리듀스를 활용한 다양한 질의 처리 기법을 제안하였다. 그 중에 하나가 맵리듀스를 활용한 리버스 k-최근접 질의 처리 기법이다. 하지만 기존 기법들은 연속 리버스 k-최근접 질의 처리를 수행하기 위해 많은 처리 비용이 요구된다. 본 논문에서는 맵리듀스를 활용한 효율적인 연속 리버스 k-최근접 질의 처리 기법을 제안한다. 제안하는 기법은 60도 가지치기 기법을 이용한다. 제안하는 기법은 60도 가지치기 기법을 활용하여 모니터링 영역을 생성하고 모니터링을 수행하여 결과 업데이트를 수행하기 때문에 효율적으로 결과 업데이트를 수행한다. 또한, 본 논문에서는 제안하는 기법의 우수성을 보이기 위해 기존 기법과의 성능평가를 수행한다.

■ 중심어 : | 맵리듀스 | 리버스 k-최근접 질의 | 연속 질의처리 | 위치기반서비스 | 빅데이터 |

Abstract

With the development of location aware technologies and mobile devices, location-based services have been studied. To provide location-based services, many researchers proposed methods for processing various query types with Mapreduce(MR). One of the proposed methods, is a Reverse k-nearest neighbor(RkNN) query processing method with MR. However, the existing methods spend too much cost to process the continuous RkNN query. In this paper, we propose an efficient continuous RkNN query processing method with MR to resolve the problems of the existing methods. The proposed method uses the 60-degree-pruning method. The proposed method does not need to reprocess the query for continuous query processing because the proposed method draws and monitors the monitoring area including the candidate objects of a RkNN query. In order to show the superiority of the proposed method, we compare it with the query processing performance of the existing method.

■ keyword : | MapReduce | Reverse k-NN Query | Continuous Query | Location-based Services | Bigdata |

* 본 논문은 한국콘텐츠학회 ICC2014 국제학술대회 우수논문입니다.

* 본 연구는 미래창조과학부 및 정보통신산업진흥원의 대학IT연구센터육성지원사업/IT융합고급인력과정지원사업(NIPA-2014-H0301-14-1022), 2012년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업(2012R1A1A2A10042015, 2012R1A1A2041898)과 한국과학기술정보연구원의 「고성능 컴퓨팅 기반 빅데이터 기술 개발(K-14-L06_C04-S01)」 사업으로부터 지원받아 수행된 연구임

접수일자 : 2014년 09월 01일

심사완료일 : 2014년 09월 19일

수정일자 : 2014년 09월 18일

교신저자 : 유재수, e-mail : yjs@chungbuk.ac.kr

I. 서론

최근 위치 측정 기술과 모바일 기기들의 발달과 함께 위치 기반 서비스(Location-based Services : LBS)가 중요하게 연구되고 있다[1][2]. 위치기반서비스를 제공하기 위해 점 질의, 범위 질의[3][4], k-최근접(k-Nearest Neighbor : kNN) 질의[5][6], 스카이라인 질의[7][8], Top-k 질의[9][10]와 같은 다양한 질의 처리 기법들이 연구되었다. 그 중에서도 k-최근접 질의는 위치기반서비스에서 가장 중요하게 연구되는 질의이다. k-최근접 질의는 질의로부터 가장 가까운 k개의 대상 객체들 검색하는 질의이다.

k-최근접 질의를 응용한 질의의 하나로 리버스 k-최근접(Reverse kNN : RkNN) 질의 처리가 연구되었다[11][12]. 리버스 k-최근접 질의 처리는 질의를 내리는 객체를 k-최근접한 결과로 인식하는 대상 객체를 검색하는 질의이다. 예를 들어 사용자가 택시를 검색하기 위해 위치기반서비스를 이용할 때, 사용자는 다양한 질의를 활용할 수 있다. k-최근접 질의를 이용할 경우, 사용자는 “나의 위치로부터 가깝게 위치한 택시”를 검색 결과로 수신할 수 있다. 하지만 사용자가 이용하고자 하는 택시를 가깝게 생각하는 다른 사용자가 있어서, 먼저 그 택시를 이용하게 되면 사용자는 그 택시를 이용할 수 없게 된다. 하지만 리버스 k-최근접 질의를 이용하여 “나를 가깝게 위치한다고 생각하는 택시”를 선택하면 그 택시는 다른 사용자보다 나를 우선적으로 고려하고 있기 때문에 다른 사용자들로 인해 택시를 이용하지 못하는 상황이 발생하지 않을 것이다. 따라서 이러한 서비스를 제공하기 위하여 리버스 k-최근접 질의 처리가 필요하다.

최근 빅 데이터 처리가 중요하게 연구되면서 대규모의 데이터의 병렬 처리를 지원하기 위한 오픈 소스 프로젝트인 하둡(Hadoop)이 많이 연구되고 있다[13][14]. 하둡은 대규모의 데이터를 분산 저장하고 관리하기 위한 플랫폼인 하둡파일시스템(Hadoop Distribute File System : HDFS)[13]와 대규모의 데이터를 분산 병렬 처리하기 위한 플랫폼인 맵리듀스(MapReduce : MR)[14]로 이루어져있다. 최근 빅 데이터 환경에서 맵

리듀스를 활용한 많은 질의 처리 기법들이 연구되었다[15][16]. 그 중의 하나로 빅 데이터 환경에서 리버스 k-최근접 질의 처리 기법이 연구되었다[17-20]. 하지만 기존 기법들은 연속 질의 처리를 고려하지 않았기 때문에 질의 결과 업데이트를 수행하는데 많은 비용이 발생한다.

본 논문에서는 맵리듀스를 활용한 효율적인 연속 리버스 k-최근접 질의 처리 기법을 제안한다. 제안하는 기법은 전통적인 리버스 k-최근접 질의 처리 기법 중의 하나인 60도 가지치기 기법[21]을 이용한다. 제안하는 기법은 60도 가지치기 기법을 활용하여 모니터링 영역을 생성하고 모니터링을 수행하여 결과 업데이트를 수행하기 때문에 효율적으로 결과 업데이트를 수행한다. 제안하는 처리 기법은 총 3단계로 이루어져있다. 1단계는 지역적 처리 단계(Local processing phase)이고, 2단계는 전역적 처리 단계(Global processing phase)이다. 마지막 3단계는 연속 질의 처리 단계(Continuous query processing phase)이다. 1단계와 2단계를 통해 제안하는 기법은 최초의 리버스 k-최근접 질의 처리 결과를 도출하며, 3단계를 통해 리버스 k-최근접 질의 처리 결과의 업데이트를 수행하여 결과를 항상 최신으로 유지한다.

본 논문의 구성은 다음과 같다. 2장에서는 이전에 연구된 맵리듀스를 활용한 리버스 k-최근접 질의 처리 기법을 분석한다. 3장에서는 본 논문에서 제안하는 맵리듀스를 활용한 효율적인 연속 리버스 k-최근접 질의 처리 기법을 소개한다. 4장에서는 제안하는 기법의 우수성을 확인하기 위해, 기존 기법과의 성능평가를 수행한다. 5장에서는 결론과 향후연구를 기술하고 논문을 마친다.

II. 관련 연구

최근 맵리듀스를 활용한 리버스 k-최근접 질의를 처리하기 위한 방법이 연구되었다. 대표적인 연구들로는 보로노이 기반 알고리즘[17]과 이등분선 기반 알고리즘(Lazy-MRRkNN)[18]이 있다. [그림 1]은 기존 기법들

의 특징을 보여준다.

비교	보로노이 기반	이등분선 기반
전 처리	존재	존재하지 않음
질의처리	상대적으로 빠름	상대적으로 느림
업데이트	고려하지 않음 (다이어그램 재구축)	고려하지 않음 (이등분선 재구축)

그림 1. 기존 기법 비교

A. Akdogan et al.(2010)이 제안한 보로노이 기반 알고리즘[17]은 리버스 k-최근접 질의 처리를 위하여 보로노이 다이어그램(Voronoi Diagram)을 이용한다. [그림 2]는 보로노이 기반 알고리즘을 보여준다. 보로노이 다이어그램이란 [그림 2]처럼 각 객체들 사이에 위치 관계를 이등분선으로 정리하여 연결한 것이다. 보로노이 다이어그램을 통해 q_i 과 같은 질의가 내려졌을 때 q_i 에서 가장 가까이 위치한 객체는 p_i 이라는 것을 손쉽게 확인할 수 있다. 보로노이 기반 알고리즘은 맵리듀스에서 데이터를 각 노드에 매핑할 때, 보로노이 다이어그램을 이용하여 데이터를 나누어 저장하여 k-최근접 질의와 리버스 k-최근접 질의를 효율적으로 처리할 수 있다. 보로노이 기반 알고리즘은 데이터가 수정되지 않는 환경에서는 보로노이 다이어그램을 지속적으로 활용할 수 있기 때문에 훌륭한 질의 처리 성능을 보인다. 하지만 데이터 추가 등의 변경이 발생했을 때, 보로노이 다이어그램을 다시 구축하여야하기 때문에 연속 질의 처리에 효율적이지 못하다는 단점을 가진다.

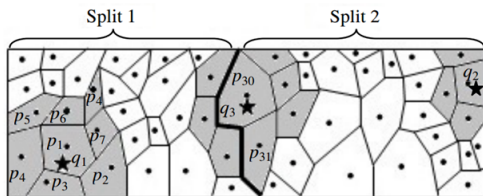


그림 2. 보로노이 기반 알고리즘

C. Ji et al.(2013)이 제안한 이등분선 기반 알고리즘 [18]은 리버스 k-최근접 질의 처리를 위하여 이등분선

의 성질을 이용한다. [그림 3]은 이등분선 기반 알고리즘을 보여준다. 보로노이 기반 알고리즘은 보로노이 다이어그램을 구축하기 위하여 모든 객체들의 위치 관계를 정리하여 연결하는데 비해, 이등분선 기반 알고리즘은 질의가 내려졌을 때, 질의와 p_1, p_3, p_4 와 같은 주변 객체들과의 관계만을 [그림 3]과 같이 구축한다. 이등분선 기반 알고리즘은 맵리듀스에서 데이터를 각 노드에 매핑할 때, 이등분선에 의해 만들어진 영역들을 기준으로 데이터를 나누어 저장하여 리버스 k-최근접 질의를 효율적으로 처리할 수 있다. 이등분선 기반 알고리즘은 보로노이 기반 알고리즘과는 달리 전처리를 위하여 소비하는 비용이 존재하지 않는다는 장점이 존재한다. 하지만 이등분선 기반 알고리즘 역시 데이터 추가 등의 변경이 발생했을 때, 이등분선을 재사용하기 힘들기 때문에 연속 질의 처리에 효율적이지 못하다는 단점을 가진다.

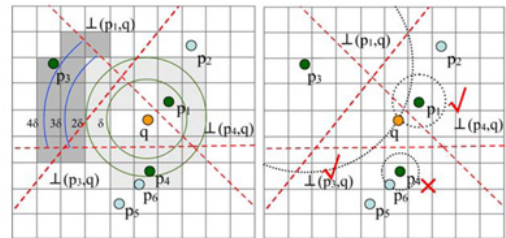


그림 3. 이등분선 기반 알고리즘

III. 제안하는 기법

1. 전체적인 처리 절차

최근 빅 데이터 환경에서 리버스 k-최근접 질의 처리 기법이 연구되었다. 하지만 기존 기법들은 연속 질의 처리를 고려하지 않았기 때문에 연속 리버스 k-최근접 질의 처리를 수행하기 위해 보로노이 다이어그램이나 이등분선을 재구성하기 때문에 많은 처리 비용이 요구된다. 따라서 본 논문에서는 맵리듀스를 활용한 효율적인 연속 리버스 k-최근접 질의 처리 기법을 제안한다. 제안하는 기법은 전통적인 리버스 k-최근접 질의 처리

기법 중의 하나인 60도 가지치기 기법을 이용한다. [그림 4]는 60도 가지치기 기법을 보여준다. 60도 가지치기 기법은 질의를 기준으로 주변 영역을 60도씩 6개의 영역으로 분할하여 각 영역마다 가장 질의에 가깝게 위치한 객체를 후보 객체로 선정한다. 후보 객체를 제외한 나머지 객체들은 가지치기한다. 그 이유는 후보 객체를 제외한 나머지 객체들의 k-최근접 질의 처리를 수행하면 나머지 객체들은 반드시 질의 객체보다 각 영역의 후보 객체를 가깝게 위치하는 것으로 인식하기 때문이다. 따라서 이렇게 남은 후보 객체들에 대하여 최종 검증을 수행하여 최종적인 k-최근접 질의 결과를 도출한다. 제안하는 기법은 60도 가지치기 기법을 이용하기 때문에 항상 후보 객체를 일정하게 유지할 수 있고 이를 이용하여 효율적으로 연속 질의 처리를 수행할 수 있다.

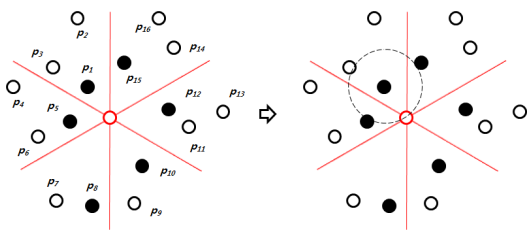


그림 4. 60도 가지치기 기법

제안하는 처리 기법은 총 3단계로 이루어져 있다. 1단계는 지역적 처리 단계(Local processing phase)이고, 2단계는 전역적 처리 단계(Global processing phase)이다. 마지막 3단계는 연속 질의 처리 단계(Continuous query processing phase)이다. 1단계와 2단계를 통해 제안하는 기법은 최초의 리버스 k-최근접 질의 처리 결과를 도출하며, 3단계를 통해 리버스 k-최근접 질의 처리 결과의 업데이트를 수행하여 결과를 항상 최신으로 유지한다.

2. 지역적 처리 단계(Local processing phase)

지역적 처리 단계는 각 노드가 자신에게 매핑된 정보들을 가지고 영역을 나누고 가지치기를 수행하여 각 노드가 리버스 k-최근접 질의를 수행하는 단계이다. [그림 5]는 본 논문에서 제안하는 기법의 지역적 처리 단계를 보여준다. 최초 지역적 처리 단계에서 각 객체는 무작위로 각 노드에 매핑된다. 맵 작업을 수행하는 매퍼(Mapper)는 자신에게 입력된 객체들의 위치 정보를 이용하여, 새로운 key와 value를 계산한다. key는 각 객체가 소속될 영역(Section)을 의미한다. value는 각 객체와 질의 객체 사이의 거리를 의미한다. Sort/Shuffle 단계에서 객체들은 key에 의해 정렬 및 분류된다. 리듀스 작업을 수행하는 리듀서(Reducer)는 정렬 및 분류된 데이터들을 대상으로 최종 후보 객체 선정을 위한 병합(Merge)을 수행한다. 모든 단계에서 각 노드는 60도 가지치기 기법을 이용하여 각 영역에서 질의 객체에 가장 가깝게 위치한 후보 객체를 제외한 나머지 객체들의 가지치기를 수행한다.

지역적 처리 단계는 총 3단계로 이루어져 있다. 1단계는 지역적 처리 단계(Local processing phase)이고, 2단계는 전역적 처리 단계(Global processing phase)이다. 마지막 3단계는 연속 질의 처리 단계(Continuous query processing phase)이다. 1단계와 2단계를 통해 제안하는 기법은 최초의 리버스 k-최근접 질의 처리 결과를 도출하며, 3단계를 통해 리버스 k-최근접 질의 처리 결과의 업데이트를 수행하여 결과를 항상 최신으로 유지한다.

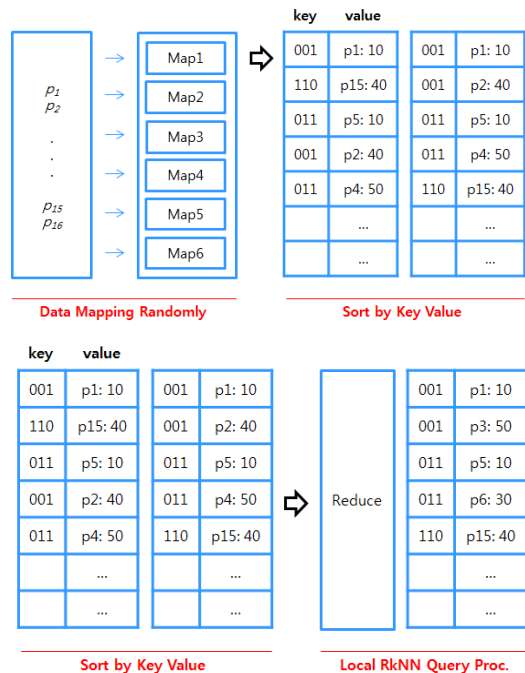


그림 5. 지역적 처리 단계

3. 전역적 처리 단계(Global processing phase)

전역적 처리 단계는 지역적 처리 단계에서 도출된 각 노드의 리버스 k-최근접 질의 결과를 병합하여 최초의 리버스 k-최근접 질의 결과를 도출하는 단계이다. [그림 6]는 본 논문에서 제안하는 기법의 전역적 처리 단계를 보여준다. 전역적 처리 단계는 지역적 처리 단계에서 도출된 각 노드의 리버스 k-최근접 질의 결과를 병합하여 최초의 리버스 k-최근접 질의 결과를 도출하는 단계이다. [그림 6]는 본 논문에서 제안하는 기법의 전역적 처리 단계를 보여준다.

림 6은 본 논문에서 제안하는 기법의 전역적 처리 단계를 보여준다. 최초 전역적 처리 단계에서 각 객체는 key에 따라 각 노드에 매핑된다. 매핑은 자신에게 입력된 객체들에 대해 최종적으로 가지치기를 수행하고 같은 영역에 존재하는 객체들과 후보 객체사이의 검증을 수행한다. 마지막으로 리듀서는 다른 영역에 존재하는 객체들과 남은 후보 객체들 사이의 최종 검증 과정을 수행하여 전역적인 리버스 k-최근접 질의 처리 결과를 도출한다.

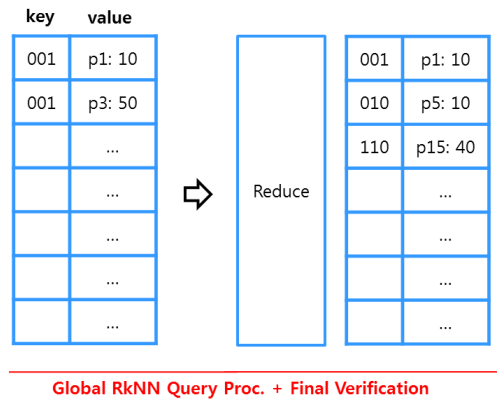
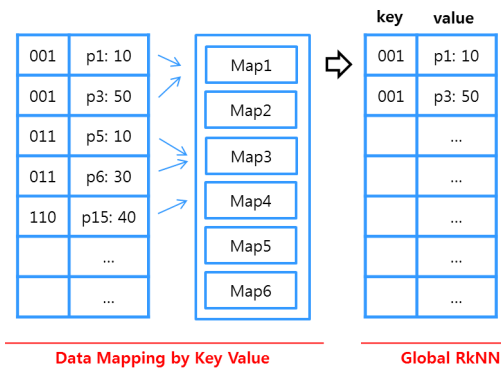


그림 6. 전역적 처리 단계

4. 연속 질의 처리 단계(Continuous query processing phase)

연속 질의 처리 단계는 최초의 리버스 k-최근접 질의 처리가 수행되고 난 뒤에 데이터의 추가 등 결과를 업데이트 할 필요가 있을 때, 업데이트를 수행하여 결과

를 최신으로 유지하는 단계이다. [그림 7]은 본 논문에서 제안하는 기법의 연속 질의 처리 단계를 보여준다. 연속 k-최근접 질의 처리 수행을 위해 제안하는 기법은 [그림 7]과 같이 모니터링 영역(Monitoring Region)을 생성한다. 모니터링 영역은 p_i, p_j 와 같은 각 영역의 후보 객체들을 이용하여 생성한다. 각 노드는 모니터링 영역을 기준으로 모니터링을 수행하고 데이터 추가 등으로 인해 결과에 변화가 생겼을 때 결과를 업데이트하여 항상 최신의 결과를 유지한다. 기존 기법은 연속 질의 처리를 위해서 질의 처리의 모든 과정을 재수행하는 것과 달리, 제안하는 기법은 전역적 처리 단계만을 수행하여 효율적으로 연속 리버스 k-최근접 질의를 처리할 수 있다.

한 번의 질의 처리를 수행하는데 소모되는 비용을 ($Cost_{local} + Cost_{global}$)로 표현한다면, 기존 기법은 n 시간만큼 연속 질의 처리를 수행했을 경우, $n * (Cost_{local} + Cost_{global})$ 의 비용이 소모된다. 하지만 제안하는 기법은 $Cost_{local} + (n * Cost_{global})$ 의 비용이 소모된다. 즉, 제안하는 기법이 기존 기법보다 $(n-1) * Cost_{local}$ 만큼 우수한 성능을 나타낸다.

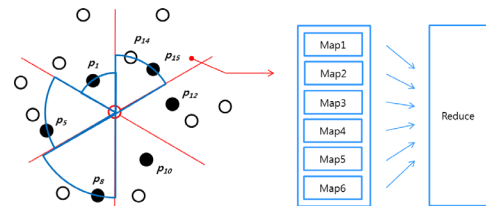


그림 7. 연속 질의 처리 단계

IV. 성능평가

1. 성능평가 환경

본 논문에서는 제안하는 기법의 우수성을 보이기 위해 기존 기법[16]과의 성능평가를 수행했다. [그림 8]은 성능평가 환경을 보여준다. 성능평가는 CentOS 운영체제, 3.0GHz CPU, 그리고 2GB 메인 메모리 환경에서 실시했다. 제안하는 기법의 성능평가를 위해, 본 논문에서

는 각 기법을 이용하여 리버스 k-최근접 질의 처리를 수행하는 과정에서 발생한 객체 접근 횟수를 측정했다. 객체의 개수는 1K개, 위치는 모두 무작위 값으로 생성하였으며 공간은 1,024 x 1,024으로 세팅했다.

Parameters	Values
Processor	Pentium 3.0GHz
Memory	2GB
OS	CentOS
Program Language	Java
Size of space	1,024 x 1,024
Number of Objects	1K

그림 8. 성능평가 환경

2. 성능평가 결과

[그림 9]는 리버스 k-최근접 질의 수행에 따른 객체 접근 횟수를 보여준다. 성능 평가 결과, 기존 기법이 제안하는 기법보다 우수한 성능을 보이는 것을 확인할 수 있다. 객체 접근 횟수는 가지치기 성능과 후보 객체들의 최종 검증과정에 가장 영향을 많이 받는다. 제안하는 기법은 후보 객체 개수가 항상 6개로 동일하게 나타나는 반면 기존 기법은 제안하는 기법보다 후보 객체의 개수가 적게 나타난다. [그림 10]은 리버스 k-최근접 질의 수행에 따른 기존 기법의 후보 객체 개수를 보여준다. 기존 기법은 상대적으로 적은 후보 객체 개수를 보이기 때문에 상대적으로 최종 검증을 위한 범위 질의가 감소하여 보다 좋은 성능을 보인다. 그럼에도 불구하고 기존 기법이 제안하는 기법보다 항상 좋은 성능만을 보이지 않고 성능이 비슷하게 나타나는 이유는 이등분선을 그리기 위하여 k-최근접 질의를 수행하며, 이때 객체 접근이 발생하기 때문이다.

[그림 11]은 연속 리버스 k-최근접 질의 수행에 따른 누적 객체 접근 횟수를 보여준다. 성능 평가 결과, 제안하는 기법이 기존 기법보다 우수한 성능을 보이는 것을 확인할 수 있다. [그림 9]와 같이 최초의 리버스 k-최근접 질의 처리에서 기존 기법이 우수한 성능을 보이는 것으로 나타나지만, 연속 질의 처리에서 기존 기법은 연속 질의 처리를 위해 질의를 재수행하여 많은 수의

객체 접근 횟수를 보이는 것을 확인할 수 있다. 하지만 제안하는 기법은 최초의 리버스 k-최근접 질의 처리에서는 기존 기법보다 다소 성능이 떨어지지만 연속 질의 처리에서는 보다 적은 객체 접근으로 리버스 k-최근접 질의를 처리하는 것을 확인할 수 있다. 제안하는 기법이 연속 질의 처리 성능에서 약 34% 우수한 성능을 보이는 것으로 나타났다.

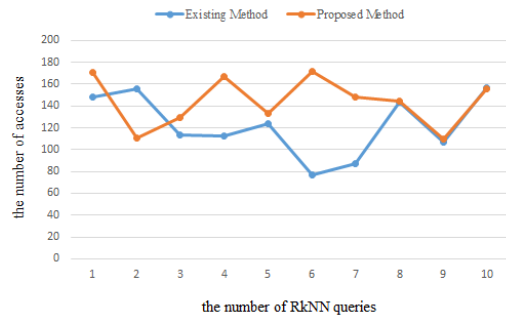


그림 9. RkNN 질의 수행에 따른 객체 접근 횟수

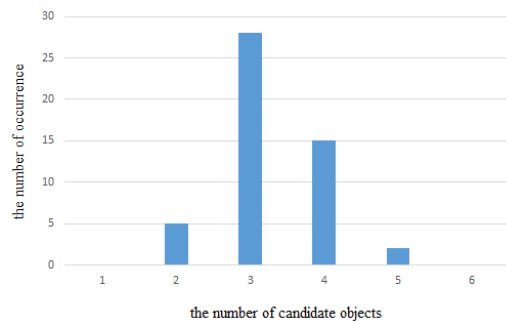


그림 10. RkNN 질의 수행에 따른 후보 객체 개수

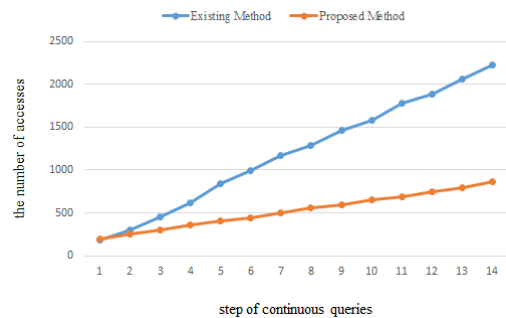


그림 11. 연속 RkNN 질의 수행에 따른 객체 접근 횟수

V. 결론

본 논문에서는 맵리듀스를 활용한 효율적인 연속 리버스 k-최근접 질의 처리 기법을 제안했다. 제안하는 기법은 전통적인 리버스 k-최근접 질의 처리 기법 중의 하나인 60도 가지치기 기법을 이용한다. 제안하는 기법은 60도 가지치기 기법을 활용하여 모니터링 영역을 생성하고 모니터링을 수행하여 결과 업데이트를 수행하기 때문에 효율적으로 결과 업데이트를 수행한다. 제안하는 기법을 통해 기존의 기법으로는 제한적이었던 연속 질의 처리가 요구되는 위치기반서비스의 제공이 가능하다. 성능평가 결과 제안하는 기법이 연속 질의 처리 성능에서 약 34% 우수한 성능을 보이는 것으로 나타났다. 추후 연구로는 질의 객체가 움직이는 환경에서 맵리듀스를 활용한 리버스 k-최근접 질의 처리 기법과 맵리듀스 환경에서 다양한 질의 처리 기법을 연구하겠다.

참고 문헌

- [1] J. Schiller and A. Voisard, "Location-based Services," Morgan Kaufmann, San Francisco, California, 2004.
- [2] S. Brakatsoulas, D. Pfoser, and N. Tryfona, "Modeling, storing, and mining moving object databases," International Database Engineering and Applications Symposium, 2004.
- [3] D. Papadias, J. Zhang, N. Mamoulis, and Y. Tao, "Query Processing in Spatial Network Databases," Proc. International conference on Very Large Data Bases, 2003.
- [4] K. L. Wu, S. K. Chen, and P. S. Yu, "On Incremental Processing of Continual Range Queries for Location-Aware Services and Applications," Proc. International Conference on Mobile and Ubiquitous Systems, pp.261-269, 2005.
- [5] N. Roussopoulos and S. F. Kelley, "Nearest neighbor queries," Proc. ACM SIGMOD international conference on Management of data, 1995.
- [6] K. Mouratidis, M. Hadjieleftheriou, and D. Papadias, "Conceptual Partitioning: An Efficient Method for Continuous Nearest Neighbor Monitoring," Proc. ACM Conference on Management of Data, pp.634-645, 2005.
- [7] S. Borzsonyi, D. Kossmann, and K. Stocker, "The Skyline Operator," Proc. IEEE International Conference on Data Engineering, pp.421-430, 2001.
- [8] C. Li, B. B. Ooi, A. K. H. Tung, and S. Wang, "DADA: a Data Cube for Dominant Relationship Analysis," Proc. ACM SIGMOD International Conference on Management of Data, pp.659-670, 2006.
- [9] I. Ilyas, G. Beskales, and M. Soliman, "A Survey of Top-k Query Processing Techniques in Relational Database Systems," ACM Computing Surveys, Vol.40, 2008.
- [10] I. F. Ilyas, W. G. Aref, and A. K. Elmagarmid, "Supporting top-k join queries in relational databases," Proc. International conference on Very Large Data Bases, 2003.
- [11] W. Wu, F. Yang, C. Y. Chan, and K. Tan, "Continuous Reverse k-Nearest-Neighbor Monitoring," Proc. International Conference on Mobile Data Management, pp.132-139, 2008.
- [12] B. Yao, F. Li, and P. Kumar, "Visible Reverse k-Nearest Neighbor Queries," Proc. IEEE International conference on Data Eng. (ICDE), 2009.
- [13] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, "The Hadoop Distributed File System," Proc. IEEE Symposium on Mass Storage Systems and Technologies (MSST),

pp.1-10, 2010.

- [14] J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters," Proc. conference on Symposium on Operating Systems Design & Implementation, p.10, 2004.
- [15] 류은경, 손인국, 박준호, 복경수, 유재수, "비-전용 분산 컴퓨팅 환경에서 맵-리듀스 처리 성능 최적화를 위한 효율적인 데이터 재배치 알고리즘", 한국콘텐츠학회논문지, 제13권, 제9호, pp.28-35, 2013.
- [16] 손인국, 류은경, 박준호, 복경수, 유재수, "맵-리듀스 처리 속도 향상을 위한 데이터 접근 패턴에 따른 핫-데이터 복제 기법", 한국콘텐츠학회논문지, 제13권, 제11호, pp.21-27, 2013.
- [17] A. Akdogan, U. Demiryurek, F. Banaei-Kashani, and C. Shahabi, "Voronoi-based Geospatial Query Processing with MapReduce," Proc. of the IEEE Cloud Computing, 2010.
- [18] C. Ji, H. Hu, Y. Xu, and Y. Li, "Efficient Multi-dimensional Spatial RkNN Query Processing with MapReduce," Proc. ChinaGrid Annual Conference, 2013.
- [19] W. Lu, Y. Shen, S. Chen, and B. C. Ooi, "Efficient processing of k nearest neighbor joins using MapReduce," Proc. VLDB Endow (PVLDB), Vol.5, No.10, pp.1016-1027, 2012.
- [20] C. Ji, T. Dong, Y. Li, Y. Shen, K. Li, and W. Qiu, "Inverted grid-based knn query processing with mapreduce," Proc. ChinaGrid annual conference, pp.25-32, 2012.
- [21] I. Stanoi, D. Agrawal, and A. E. Abbadi, "Reverse Nearest Neighbor Queries for Dynamic Databases," Proc. of the ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, 2000.

저 자 소 개

임 중 태(Jongtae Lim)

정회원



- 2009년 2월 : 충북대학교 정보통신공학과(공학사)
- 2011년 2월 : 충북대학교 정보통신공학과(공학석사)
- 2011년 3월 ~ 현재 : 충북대학교 정보통신공학과 박사과정

<관심분야> : 데이터베이스 시스템, 시공간 데이터베이스, 위치기반 서비스, 이동 P2P 네트워크, 소셜 네트워크 서비스, 빅 데이터 등

박 선 용(Sunyong Park)

준회원



- 2013년 8월 : 배재대학교 전자상거래학과(경영학사)
- 2011년 2월 ~ 현재 : 충북대학교 정보통신공학과 석사과정

<관심분야> : 이동 P2P 네트워크, 소셜 네트워크 서비스, 빅 데이터 등

서 기 원(Kiwon Seo)

준회원



- 2013년 2월 : 충북대학교 정보통신공학과(공학사)
- 2013년 2월 ~ 현재 : 충북대학교 정보통신공학과(공학석사)

<관심분야> : 데이터베이스 시스템, 위치기반 소셜 네트워크 서비스, 빅 데이터 등

이 민 호(Minho Lee)

정회원



- 2000년 : 충남대학교 대학원 컴
퓨터공학과 졸업(석사)
- 2014년 : 충남대학교 대학원 컴
퓨터공학과(박사)
- 2001년 ~ 현재 : 한국과학기술
정보연구원 선임연구원

<관심분야> : 데이터마이닝, 바이오인포매틱스

북 경 수(Kyoungsoo Bok)

종신회원



- 1998년 2월 : 충북대학교 수학과
(이학사)
- 2000년 2월 : 충북대학교 정보통
신공학과 (공학석사)
- 2005년 3월 ~ 2008년 2월 : 한국
과학기술원 정보전자연구소

Postdoc

- 2008년 3월 ~ 2011년 2월 : 가인정보기술 연구소 연
구원
- 2011년 3월 ~ 현재 : 충북대학교 전자정보대학 정보
통신공학부 초빙부교수

<관심분야> : 데이터베이스 시스템, 자료저장시스템,
이동객체 데이터베이스, RFID 및 센서네트워크, 모
바일 P2P 네트워크

유 재 수(Jaesoo Yoo)

종신회원



- 1989년 2월 : 전북대학교 컴퓨터
공학과(공학사)
- 1991년 2월 : 한국과학기술원 전
산학과(공학석사)
- 1995년 2월 : 한국과학기술원 전
산학과(공학박사)

- 1995년 3월 ~ 1996년 8월 : 목포대학교 전산통계학
과 전임강사

- 1996년 8월 ~ 현재 : 충북대학교 전자정보대학 교수

<관심분야> : 데이터베이스 시스템, XML, 멀티미디
어 데이터베이스, 분산 객체 컴퓨팅, 빅 데이터 등