

# 맵리듀스를 이용한 효율적인 k-NN 조인 질의처리 알고리즘

윤들녁\*, 장미영\*, 장재우\*\*

\*전북대학교 컴퓨터공학과

\*\*전북대학교 IT정보공학과

e-mail:[emfsur89, brilliant, jwchang]@jbnu.ac.kr

## Efficient k-Nearest Neighbor Join Query Processing Algorithm using MapReduce

Deulnyeok Yun\*, Miyoung Jang\*, Jaewoo Chang\*\*

\*Dept of Computer Science, Chonbuk National University

\*\*Dept of Information Technology, Chonbuk National University

### 요약

대용량 데이터를 분석하기 위한 맵리듀스 기반 k-NN 조인 질의처리 알고리즘은 최근 데이터 마이닝 및 분석을 기반으로 하는 응용 분야에서 매우 중요하게 활용되고 있다. 그러나, 대표적인 연구인 보로노이 기반 k-NN 조인 질의처리 알고리즘은 보로노이 인덱스 구축 비용이 매우 크기 때문에 대용량 데이터에 적합하지 못하다. 아울러 보로노이 셀 정보를 저장하기 위해 사용하는 R-트리는 맵리듀스 환경의 분산 병렬 처리에 적합하지 않다. 따라서, 본 논문에서는 새로운 그리드 인덱스 기반의 k-NN 조인 질의 처리 알고리즘을 제안한다. 첫째, 높은 인덱스 구축 비용 문제를 해결하기 위해, 데이터 분포를 고려한 동적 그리드 인덱스 생성 기법을 제안한다. 둘째, 맵리듀스 환경에서 효율적으로 k-NN 조인 질의를 수행하기 위해, 인접셀 정보를 시그니처로 활용하는 후보영역 탐색 및 필터링 알고리즘을 제안한다. 마지막으로 성능 평가를 통해 제안하는 기법이 질의 처리 시간 측면에서 기존 기법에 비해 최대 3 배 높은 질의 처리 성능을 나타낸다.

### 1. 서론

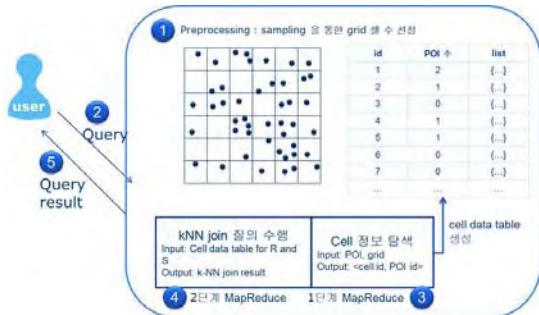
최근 IT 분야에서 대용량 데이터를 처리하기 위한 클라우드 컴퓨팅에 대한 연구가 활발히 진행되고 있다[1]. 이러한 대용량 데이터에 대한 관리를 지원하는 대표적인 기술로는 하둡(Hadoop)[2]이 존재한다. 하둡이란 대용량 데이터 저장을 위한 분산 파일 시스템 및 병렬 데이터 처리를 지원하는 맵리듀스[6] 기술을 구현한 대표적인 빅데이터 관리 시스템으로, 아마존, 야후 등 글로벌 기업의 주요 프로젝트에 적극 활용되고 있다<sup>1)</sup>.

한편, 최근 관심을 받고 있는 대용량 과학 데이터베이스 분석을 위한 대표적인 기법으로는 k-NN 조인 질의가 존재한다[3,4]. K-NN

조인 질의처리 알고리즘은 서로 다른 두 개의 데이터베이스 R과 S가 존재할 때, R의 모든 데이터 r에 대해 S의 데이터 중 가장 거리가 가까운 상위 k개의 데이터를 탐색하는 알고리즘이다. 기존의 k-NN 조인 기법은 대부분 중앙 집중적인 인덱스 구조에 의존적이며, 기본적으로 k-NN 연산 비용이 계속적으로 증가하는 문제점을 지닌다. 이를 해결하기 위해, Wei Lu et al.[5]은 맵리듀스 환경에서 보로노이 인덱스 구조를 사용한 k-NN 조인 질의 처리 기법을 제안하였다. 그러나 해당 기법은 데이터 업데이트에 따른 보로노이 인덱스 구축 비용이 매우 크며, 보로노이 셀 정보를 저장하기 위해 사용하는 R-트리는 맵리듀스 환경에서의 분산 병렬 처리에 적합한 구조를 제공하지 못한다.

본 논문에서는 대용량 데이터 처리에 효율적인 k-NN 조인 질의 처리 알고리즘을 제안한

1) B. F. Cooper et al. "Benchmarking cloud serving systems with YCSB", ACM Symposium on Cloud Computing.



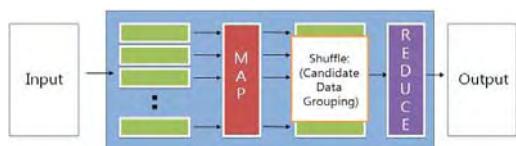
(그림 2) 제안하는 시스템 구조

다. 첫째, 높은 인덱스 구축 비용 문제를 해결하기 위해, 데이터 분포를 고려한 동적 인덱스를 제안한다. 둘째, 맵리듀스 환경에서 효율적으로 k-NN 조인 질의를 수행하기 위해, 인접 셀 정보를 시그니처로 활용하는 후보영역 탐색 및 필터링 알고리즘을 제안한다. 이를 통해, 데이터 입출력 및 연산 시간이 크게 감소하는 장점을 지닌다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 최신 연구인 PBJ 기법을 설명하고 문제점을 분석한다. 3장에서는 제안하는 동적 인덱스 기반 k-NN 질의 처리 알고리즘을 설명한다. 4장에서는 제안하는 알고리즘의 효율성을 검증하기 위한 성능 평가를 수행하며, 마지막으로 5장에서는 결론을 도출하고 향후 연구를 제시한다.

## 2. 관련 연구

맵리듀스[10]는 데이터를 key와 value로 구성한 한 쌍의 데이터로 변환하여 사용자가 정의한 함수를 병렬 처리하는 모델로서, (그림 1)과 같이 실행한다. 입력 데이터는 일정한 크기의 split으로 분할되어 각 맵에 병렬로 입력되고 중간 결과를 생산한다. 생산된 중간 결과는 다시 리듀스 단계에 입력되어 태스크를 수행하고 최종결과를 산출한다.



(그림1) 맵리듀스 실행 모델

Wei Lu et al.[5]이 제안한 맵리듀스를 이용한 보로노이 기반 k-NN 조인 질의처리 알고리즘은 총 3단계를 통해 수행된다. 첫째, 전처리 단계에서 k-NN 조인에 대한 연산 비용을 줄이기 위해 R과 S에 대한 pivot을 랜덤으로 선정한다. 둘째, 1단계 맵리듀스 단계에서는 선정한 pivot을 이용하여 각 데이터 셋 S와 R에 대한 보로노이 인덱스를 구축한다. 아울러, 각 보로노이 파티션에 할당된 데이터 셋 개수, 파티션 간의 최대/최소거리, 가장 가까운 k개의 거리에 대한 파티션 통계정보(Partition statics)정보를 수집한다. 마지막으로, 2단계 리듀스 함수에서는 데이터 셋 R의 파티션에 대해 k-NN 후보를 포함하는  $S_i$  서브셋을 탐색하여 최종 k-NN 후보를 탐색하여 결과를 반환한다.

그러나 이 기법은 보로노이 인덱스를 사용하기 때문에, 데이터 업데이트 시 인덱스 재구축 비용 문제를 야기하며, pivot을 실제데이터로 선정하여 보로노이 다이어그램을 생성하기 때문에 데이터 밀집도에 따라서 성능의 차이가 큰 문제가 존재한다.

## 3. 맵리듀스를 이용한 그리드 기반 k-NN 조인 질의처리 알고리즘

제안하는 시스템 구조는 전처리 단계, 1단계, 2단계 맵리듀스로 구성된다, (그림 2).

### 전처리(preprocessing)단계 : 샘플 추출을 통한 그리드 분할 값 설정

전처리 단계에서는, 데이터베이스 R, S로부터 데이터 분할을 위한 초기 그리드 셀 영역을 생성한다. 또한 R, S의 부분 데이터를 추출하고 이를 그리드 인덱스에 저장한다. 이때, 각 그리드 셀은 식 (1)을 이용하여 데이터 삽입된 경우 1, 데이터가 없는 경우 0을 나타내는 비트를 생성한다.

$$BS_{p_i} = \begin{cases} 0, & \text{if } p_i \text{ is a non-empty cell} \\ 1, & \text{otherwise} \end{cases} \quad \text{식 (1)}$$

따라서, 최종적으로 그리드 셀 수만큼의 비트열이 생성된다. 이때, 모든 맵 함수에서 병렬적으로 비트 정보를 생성하기 때문에, 리듀스 함

수에서 이를 집계하여 최종 그리드 셀에 대한 비트열을 생성한다. 또한 셔플(suffle) 단계를 통해 동일한 그리드 셀 아이디를 지니는 데이터를 그룹화 함으로써, 같은 리듀스 함수에서 처리할 수 있도록 한다. 리듀스 함수는 생성된 비트열 정보 및 그리드 셀의 밀도를 계산함으로써, 적절한 그리드 분할 값을 계산한다. 아울러, 각 그리드 셀의 인접 셀의 id를 저장하여 k-NN 질의 수행 시 영역 확장을 지원한다.

### 1단계 맵리듀스 : SummaryTable 생성

1단계 맵 함수에서는 생성된 그리드를 이용하여 실제 데이터 R과 S를 삽입하고, 각 데이터가 포함된 셀의 id를 키(key)로 하는 중간 결과물을 생성한다. 따라서, 맵 함수의 결과는 <셀 id, 데이터 정보(id, x, y)>의 형태로 저장된다. 1단계 리듀스 함수에서는 생성된 정보를 종합하여 각 데이터 R 및 S에 대한 SummaryTable을 생성한다. (그림 2)에서 데이터 R의 경우, <그리드셀 id, 데이터 id, x, y> 형태로 저장된다. 한편, S 데이터의 경우 2차 맵리듀스 단계에서 Ri의 인접 셀리스트에 포함되는 Si를 Ri와 동일한 노드로 전송해야 하기 때문에, 데이터가 저장된 S의 셀을 탐색하고, 해당 Si가 인접 셀로 저장된 Ri의 수만큼 반복하여 데이터를 생성한다.

### 2단계 맵리듀스 : k-NN 조인 질의 처리

2단계 맵리듀스에서는, 질의 데이터인 R을 읽고 질의와 동일한 셀에 저장된 S의 데이터를 SummaryTable을 이용하여 탐색한다. 질의 점( $r_q$ )과 탐색된 S 데이터 간의 거리를 측정하여 동일한 셀에 저장된 데이터 중에서 k-NN 데이터를 찾고, 거리를 계산한다. 아울러, 본 논문에서는 k-NN의 정확도를 보장하기 위해 외부 인접셀까지의 확장을 수행한다. 이때, 불필요한 확장으로 인한 k-NN 연산 오버헤드를 감소하기 위해, 질의 셀로부터 총 8방향(동, 서, 남, 북, 북서, 북동, 남서, 남동)으로의 거리를 검색하여, 확장 여부를 결정한다. 셀 확장 조건은 다음과 같다. 질의 점( $r_q$ )은 자신이

포함된 셀( $R_i$ )에서 가장 인접한 한 변까지의 거리  $dist(r_q, s_i)$ 와 꼭지점까지의 거리  $dist(r_q, v_i)$ 를 계산하고, k-NN 데이터의 거리  $dist(r_q, k-NN)$ 과 비교하여, 최소 값을 탐색한다. 이때, 최근접 변까지의 거리가 k-NN 데이터보다 크면, 해당 변을 이루는 꼭지점을 포함하는 모든 셀은 확장에서 제외된다. 만약 최근접 변까지의 거리가 k-NN 데이터까지의 거리보다 작다면, 해당 변의 방향으로 확장을 수행한다. 아울러, 해당 변을 이루는 꼭지점 중 질의와 가장 근접한 꼭지점의 방향을 고려하여 추가 확장 여부를 계산하고, 조건을 만족할 경우 추가 확장을 수행한다. 질의 셀의 외부 확장은 조건을 만족하는 셀이 없을 때까지 반복한다. 마지막으로 리듀스 함수에서는 최종 후보 영역에 포함된 모든 셀의 데이터와 질의 점과의 거리를 계산하고, 최종 k-NN 데이터를 반환한다.

## 4. 성능 평가

본 장에서는 제안하는 그리드 인덱스 기반 k-NN 조인 질의 처리 알고리즘의 우수성을 검증하기 위해, 기존 PBJ[5] 기법과 제안하는 기법의 질의 처리 성능을 비교한다. 성능평가의 실험 환경은 <표 1>과 같다. MapReduce 환경 구축을 위해 범용 pc 7대에서 하둡 2.2.0을 이용하여 기존 연구의 알고리즘 및 제안하는 알고리즘을 구현하고, 성능 평가를 수행하였다.

<표 1> 실험 환경

항목	성능
CPU	Core i3-3240 CPU @ 3.40GHz
Memory(CPU)	8GB
OS	Ubuntu 12.04
Data Set	약 5만, 150만, 200만, 250만 개
K	20,40,60,80,100
N	30,50,70,80,100

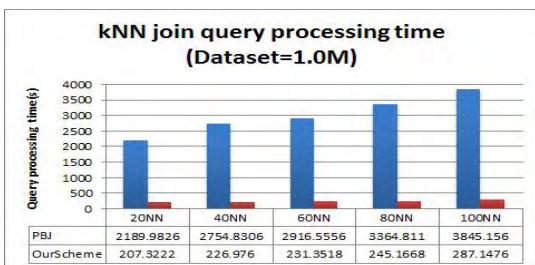
### 4.1 질의 처리 시간

<표 2>는 약 56만개의 데이터를 이용해 수행한 제안하는 알고리즘과 기존 PBJ의 전처리 수행 시간을 나타낸다. 기존 기법의 경우, 보로노이 다이어그램을 이용하여 데이터 분할 영역을 생성하고, 해당 정보를 R-tree에 저장한다. 따라서 대용량 데이터 및 업데이트가 빈번하게

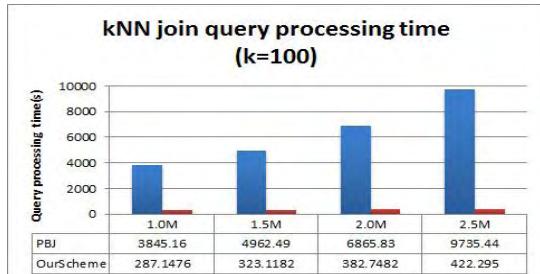
발생하는 위치데이터 등에 적합하지 않은 특징을 나타낸다. 반면, 제안하는 기법은 그리드를 사용하기 때문에 분할 영역 연산이 빠르고 한번 인덱스를 구축하면 데이터 업데이트가 용이하다는 특성을 가지고 있으므로, PBJ에 비해 대용량 데이터 처리 및 인덱스 구축에 적합하다.

&lt;표 2&gt; 전처리 시간

항목	PBJ	제안 기법
시간 (초)	24.84	12.57



(a) k의 변화에 따른 질의 처리 시간

(b) 데이터 크기 별 질의 처리 시간  
(그림 9) 질의 처리 시간 측정 결과

(그림 9 (a))는 데이터 개수가 100만, 그리드 분할 수가 100일 때, k 변화에 따른 질의 처리 시간의 변화 추이를 나타낸다. k가 20일 때, 제안하는 기법은 207.3222초가 요구되는 반면, PBJ 기법은 2189.983초가 요구된다. 이는 제안하는 기법이 질의 셀과 인접한 셀 정보를 그리드 인덱스를 이용하여 빠르게 탐색하는 반면, 기존 연구는 인접한 보로노이 영역 탐색 및 거리 계산 비용이 크기 때문이다. 또한 (그림 9(b))는 그리드 분할 수가 100, k가 100일 때, 데이터 크기의 변화에 따른 질의 처리 시간을 나타낸다. 데이터 크기가 증가될수록 셀에 포함되는 데이터 수가 증가하여 질의 처리 시간이 증가하나, 기존 연구와 비교했을 때 평균 17배 빠른 것을 확인할 수 있다.

## 5. 결론 및 향후 연구

본 논문에서는 맵리듀스 환경에서 효율적으로 k-NN 조인 질의를 수행하기 위한 그리드 인덱스 기반 k-NN 조인 질의처리 알고리즘을 제안한다. 제안하는 기법은 기존 k-NN 조인 질의처리 연구인 PBJ[5]와 성능비교를 통해 질의 처리 시간 측면에서 평균 17배 우수한 성능을 제시함을 입증하였다.

향후 연구는 다양한 분석 질의를 지원하기 위해 제안하는 기법을 top-k 질의를 지원하는 알고리즘으로 확장하는 것이다.

## 6. Acknowledgement

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(grant number 2013R1A1A4A01010099).

## 참고 문헌

- [1] 민영수, 김홍연, 김영균, 2009, "클라우드 컴퓨팅을 위한 분산 파일 시스템 기술", 한국정보과학회지, 제27권 제5호, pp. 86-94.
- [2] Apache Software Foundation, Hadoop MapReduce: <http://hadoop.apache.org/> -mapreduce
- [3] C. Yu, B. Cui, S. Wang, and J. Su. Efficient index-based knn join processing for high-dimensional data. Information and Software Technology, 49(4):332 - 344, 2007
- [4] B. Yao, F. Li, and P. Kumar. K nearest neighbor queries and knn-joins in large relational databases (almost) for free. In ICDE, pages 4 - 15, 2010
- [5] Lu, Wei, et al. "Efficient processing of k nearest neighbor joins using mapreduce." Proceedings of the VLDB Endowment 5.10 (2012): 1016-1027.
- [6] J. Dean, S. Ghemawat, 2004, "MapReduce : Simplified Data Processing on Large Clusters," Operating System Design and Implementation, pp. 10-10