

병렬 분산 처리 시스템에서 공간 연산을 위한 데이터 접근 방안

김진덕*

*동의대학교

A Method to Access Data for Spatial Operation in Parallel Distributed Processing System

Jindeog Kim*

*Donggeui University

E-mail : jdk@deu.ac.kr

요 약

과거에 비해 비약적으로 생산되는 공간 데이터에 대한 처리를 위한 공간 연산은 빠른 처리 응답성을 요구하는 경우가 많다. 그래서 최근 하둡(Hadoop)과 같은 빅데이터 처리 시스템을 이용하여 처리하고자 하는 시도가 많다. 한편, 공간 조인은 데이터 분할(Partitioning)과 공간 색인의 이용 여부, 여과 단계와 정제 단계를 거치는 등 그 복잡도가 강한 공간 연산이다. 그래서 빅데이터 처리 시스템을 이용한 공간 조인의 처리 방식은 매우 다양하다. 그러나 지금까지 이러한 공간 조인의 처리 방식에 다른 리소스 활용에 대한 비교는 거의 없다.

이 논문에서는 다양한 공간 연산의 수행 방법에 따른 빅데이터 시스템 클러스터에서 데이터 전송 방식을 고찰하고 데이터 전송에 따른 네트워크 리소스의 효율적인 사용 방안을 제안하고자 한다. 구체적으로 단일할당과 다중할당 색인 기법의 비교, 파티셔닝 방법의 비교, 맵리듀스 시스템의 태스크 할당 방법에 따른 비교를 통해 다양한 연산 유형에 따른 공간 조인의 처리 방안 선정에 고려 요소를 제시하고자 한다.

키워드

공간정보, 병렬처리, 맵리듀스, 데이터처리

1. 서 론

공간 정보 시스템은 최근 다양한 데이터 생성 모듈(모바일 기기, 센서 등)로 인해 그 양과 횟수가 크게 증가하고 있다. 또한 이를 처리하는 공간 연산은 빠른 응답시간을 요구하고 있다.

그 중 대용량 고밀도 자료의 OpenStreetMap[4] 지도 데이터 처리 분야와 병리학(Pathology) 영상 분석[6] 분야에서도 공간 조인의 필요성이 대두되고 있다.

따라서 최근 많이 이용되고 있는 빅데이터 처리 시스템을 활용하여 공간 조인 연산을 처리하고자 하는 연구가 진행되고 있다[1,2,5,6,7,8].

이러한 데이터들은 2차원 공간상의 영역을 갖는 데이터로서 빠른 처리를 위해 공간 색인을 이용하여 이에 따른 공간 분할 정책이 필요하고, 속도 향상을 위해 여과 단계와 정제 단계가 필요하다. 일반적으로 많이 활용되고 있는 하둡 시스템에서의 맵리듀스 처리 방식은 그림 1과 같다. 기

본적으로 하둡의 태스크 할당은 FIFO와 지역성(locality)을 근본원칙으로 한다. 그러나 그림 1에서 각 노드의 맵 단계와 리듀스 단계 사이에서 지역성을 보장하기 어려워 클러스터 내의 각 노드 간에 많은 데이터 전송이 이루어진다. 또한 공간 조인의 복잡성으로 인해 여러 번의 맵리듀스 단계를 거치게 된다. 이와 같은 빅데이터 시스템에서의 다양한 공간 조인의 처리 방식에 따른 데이터 접근 방안에 대한 연구는 거의 없다.

따라서 이 논문에서는 다양한 공간 연산의 수행 방법을 제시하고, 빅데이터 시스템 클러스터에서 데이터 전송 방식을 살펴본다. 세부적으로 상황별 색인기법, 태스크 할당 방법에 따른 데이터 접근 방안을 제안하고자 한다.

제 2장에는 빅데이터 시스템에서의 공간 조인 처리 연구와 태스크의 스케줄링 방안에 대해 살펴보고, 제 3장에서는 다양한 공간 조인 유형에 따른 데이터 전송 방안을 제안하고, 제 4장에서 결론을 맺는다.

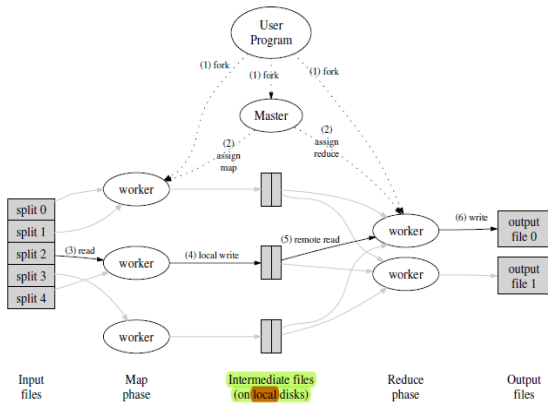


그림 1. 맵리듀스 처리 방식[14]

II. 관련 연구

하둡 맵 리듀스 시스템[14]을 확장한 공간 빅데이터 시스템 다양한 연구[5-10]가 있다. 또한 다양한 처리속도 향상방안에 대한 연구[11-17]가 있다.

Hadoop-GIS[6]는 글로벌 색인을 그리드로 사전에 생성하고, 다중할당 방식의 런타임 R*-tree 색인을 이용하여 공간 조인을 수행하고, 그 결과의 중복을 제거하는 방안을 제시하였다.

Spatial Hadoop[5]은 3단계의 MR 기반 인덱싱을 사전에 수행하고, 질의 단계는 사전에 생성된 색인을 이용하여 수행된다.

공간 조인의 수행 속도 향상을 위한 연구로서 기존 FIFO방식[14]의 스케줄링을 개선하여 Fair와 Capacity 방식[15]을 도입하는 연구와 공정성을 보존하면서 지역성을 높이기 위한 연구[16]와 이질적인 클러스터 환경의 수행속도 향상을 위한 연구[17]등이 제시되었으며, 네트워크 사용량 감소를 위한 질의 적응형 블록 재배치 알고리즘을 이용하여 지역성을 높이고자 하는 연구[12]도 있으며, 대용량 스펙 파일의 성능향상을 위해 마스터의 부하 감소를 위한 연구[13]도 수행되어 왔다. 그러나 지금까지 다양한 공간 조인의 처리 기법에 관한 고찰은 거의 없다.

III. 공간 조인 병렬 분산 처리 기법

이 장에서는 Hadoop-GIS와 SpatialHadoop에서의 공간 조인의 병렬 분산 처리 기법에 대해 살펴보고 HBase[18]를 이용한 공간 조인 병렬 분산 처리 기법을 제안하고자 한다.

HadoopGIS는 2단계의 색인 구성 방식이며, 그 중 첫 번째 맵리듀스에서 글로벌 색인은 다중 할당 과정을 거쳐 그리드 방식으로 색인한다. 색인 파일을 마스터는 분산 캐쉬(Distributed Cache) 파일로 제공하며, 맵 단계에서 각 레코드에 타일 ID를 부여하여 분할(Partition)되며, 리듀스단계에

서는 동일한 타일 ID를 가진 레코드들을 실시간으로 R*-tree를 이용하여 각 노드에서 색인한 뒤 조인 연산을 수행한다. 다중 할당 색인이므로 각 타일의 크기는 전체 데이터의 크기 및 각 공간 다각형의 면적과 밀접한 관련이 있어 적절한 휴리스틱(Heuristic)이 요구되며, 그에 따른 다중 할당 비율 만큼 마스터에서 각 노드로의 데이터 전송이 필요하다. 이 때 리듀서에서 공간 조인을 위한 여과와 정제 단계가 연속 수행된다. 그리고 두 번째 맵 리듀스 수행에서는 결과의 중복을 제거하는 연산과정이 필요하다.

SpatialHadoop은 HadoopGIS와는 달리 Grid, R-Tree, R+-tree를 이용하여 색인을 사전에 생성한 뒤 중간 파일이 아닌 생성된 HDFS 파일을 이용하여 공간 조인 질의를 수행한다. 따라서 각 Split을 읽기 위해 SpatialFileSplitter와 SpatialRecordReader가 필요하다. SpatialHadoop이 색인을 생성할 때 위치 식별 파티셔너(location aware partitioner)를 이용하며, 글로벌 색인은 그리드 또는 R-tree를 이용하며, 로컬 색인은 R+-tree를 이용한다. 단, 각자 생성된 색인 R과 S에 대해 공간 조인을 수행할 때 글로벌 레벨에서 발생하는 불일치를 해소하기 위해 실시간 재구성 과정이 필요하다.

상기의 두 가지 방법은 매우 상이한 처리 구조를 가지고 있는데 둘 다 휴리스틱과 색인 재구성이라는 과정이 필요하며, 이는 그림 2와 같이 많은 데이터 전송 과정이 요구된다.

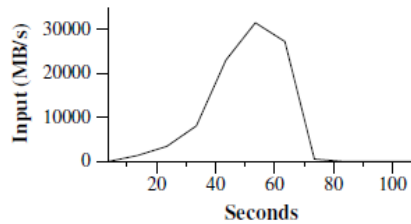


그림 2. 맵리듀스 데이터 전송 예[14]

이 논문에서는 Column Family 기반 NoSQL인 HBase[18]를 이용한 병렬 분산 공간 조인 방안을 제시한다. 공간 조인은 2항 연산자이며, 피연산자는 정해져 있지 않으므로 피연산자의 데이터 분포를 고려한 GeoHash 코드를 RowID에 반영하고자 한다. 이 때 데이터 분포의 효과적인 파악을 위해 아래와 같이 블룸 필터를 2차원으로 확대하여 맵리듀스 플랫폼을 통해 사전에 분포 정보를 파악하고 HBase에서 활용하고자 한다.

HBase는 Column Family 방식의 확장이 가능하므로 하나의 레코드에 타일 정보를 표현할 수 있으므로 레코드 단위의 조인이 가능하므로 RowID를 기반으로 한 대응 레코드 확인이 손쉬운 측면이 있으며, 태스크를 형성하기 위한 데이터의 Volume 제약이 없이 실시간 DB 검색으로 필요한 데이터만 검색하여 공간 조인을 수행할 수 있는 장점이 있다.

IV. 결 론

이 논문에서는 다양한 공간 조인의 처리 방식을 살펴보고, 기존 공간조인 병렬 처리 방법이 색인 재생성, 휴리스틱 등이 요구됨을 알아보았다. 그리고 이를 해결하기 위한 방법으로 NoSQL을 이용한 공간 조인 방법을 제안하였다.

제안한 방법은 Column Family 방식의 확장이 가능하다는 점을 활용하였고, 이에 대응 레코드 결정이 쉬우며, 실시간 DB 검색으로 Split의 양적 제약을 극복할 수 있다.

그리고 데이터 분포를 사전에 검색하여 레코드의 RowID로 표현함으로써 데이터의 불균일 분포를 해결하고자 하였다.

참고문헌

[1] Hadoop-GIS : <http://hadoopgis.org>
 [2] MongoDB : <http://www.mongodb.org>
 [3] OGC : <http://opengeospatial.org>
 [4] OpenStreetMap: <http://openstreetmap.org>
 [5] A. Eldawy, M. F. Mokbel, "A Demonstration of SpatialHadoop: An Efficient MapReduce Framework for Spatial Data", Proc. of VLDB Endowment, Vol.6, No. 12, 2013
 [6] A Aji, F. Wang, H. Vo, R. Lee, Q. Liu, X. Zhang, J. Saltz, "Hadoop-GIS: A High Performance Spatial Data Warehousing System over MapReduce, Proc. of VLDB Endowment, Vol.6, No.11, 2013
 [7] 조현구, 유기현, 양평우, 이연식, 남광우, "Spatial Tajo : 공간 빅 데이터 웨어하우스 시스템의 설계 및 구현", 한국정보과학회 동계 학술대회, 2015
 [8] 최용권, "위치 정보 데이터가 포함된 빅 데이터 환경에서 시공간 인덱스 갱신 효율을 위한 MapReduce 확장", 인하대학교 대학원 석사, 2012
 [9] A Eldawy, L Alarabi, M. F. Mokbel, "Spatial Partitioning Techniques in SpatialHadoop", Proc. of VLDB Endowment, Vol. 8, No. 12, 2015
 [10] K. Lee, R.K. Ganti, M Srivatsa, L. Liu, "Efficient Spatial Query Processing for Big Data", SIGPATIAL '14, 2014
 [11] 정대영, 남윤성, 이권용, 박성용, "가상 클러스터 환경에서 하둡 맵리듀스의 성능 향상을 위한 부하분산 기법", 한국정보과학회 컴퓨팅의 실제 및 레터, 제 19권, 제 12호, 2013
 [12] 김준상, 김창현, 이원주, 전창호, "하둡 클러스터의 네트워크 사용량 감소를 위한 블록 재

배치 알고리즘", 한국컴퓨터정보학회 논문지, 제 19권 제 11호, 2014
 [13] 최철웅, "하둡기반 대량의 스펙파일 처리 성능 향상을 위한 효율적인 데이터 처리방법", 조선대학교 대학원, 석사학위논문, 2016
 [14] J. Dean, S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters", Comm. of ACM, Vol. 51, No. 1, 2008
 [15] M Zaharia, "Job Scheduling with the Fair and Capacity Schedulers", Hadoop Summit, 2009
 [16] M Zaharia, D. Borthakur, J.S. Sarma, "Delay Scheduling: A Simple Technique for Achieving Locality and Fairness in Cluster Scheduling", EuroSys, 2010
 [17] M. Zaharia, A Konwinski, A.D. Joseph, R.H. Katz, I. Stoica, "Improving MapReduce Performance in Heterogeneous Environments", Tech. Report UCB, 2008
 [18] <http://hbase.apache.org/>