한국정보전자통신기술학회논문지(jkiiect)'17-02, Vol.10 No.1
논문 17-10-01-1 http://dx.doi.org/10.17661/jkiiect.2017.10.1.1

Hadoop에서 SQL 기반 질의언어를 지원하는 공간 빅데이터 질의처리 시스템

주인학*

Spatial Big Data Query Processing System Supporting SQL-based Query Language in Hadoop

In-Hak Joo*

요 약 본 논문에서는 Hadoop에 공간 데이터를 저장하고 SQL 기반 질의언어에 의하여 공간 데이터를 질의할 수 있는 공간 빅데이터 질의처리 시스템을 제시한다. 제안한 시스템은 대용량의 공간 빅데이터를 HDFS 기반의 저장 시스템에 저장하고 공간 데이터 처리기능이 추가확장된 SQL 기반 질의언어로 질의를 할 수 있도록 지원하며 OGC 심플 피처 모델 기반의 공간 데이터 표준 데이터타입과 함수를 지원한다. 본 논문에서는 질의언어 파싱, 질의언어 검증, 질의계획 생성, 저장시스템 연동 등 질의처리의 주요 기능 개발을 제시하였다. 제안한 시스템의 성능을 기존 시스템과 비교하였으며, 실험에서는 Hadoop에 저장된 공간 데이터에 대한 영역질의의 질의실행시간에 있어서 비교 시스템 대비 약 58%의 성능향상을 나타냄을 보였다.

Abstract In this paper we present a spatial big data query processing system that can store spatial data in Hadoop and query the data with SQL-based query language. The system stores large-scale spatial data in HDFS-based storage system, and supports spatial queries expressed in SQL-based query language extended for spatial data processing. It supports standard spatial data types and functions defined in OGC simple feature model in the query language. This paper presents the development of core functions of the system including query language parsing, query validation, query planning, and connection with storage system. We compares the performance of the suggested system with an existing system, and our experiments show that the system shows about 58% performance improvement of query execution time over the existing system when executing region query for spatial data stored in Hadoop.

Key Words : Big data, Hadoop, Query language, Query processing, Spatial data, SQL

1. 서론

SNS(Social Network Service), 모바일 어플리케이션, IoT(Internet of Things) 서비스 등의 발전과 이에 따른 수많은 데이터의 활용으로 인하여 다양한 형태와 내용을 가진 빅데이터가 폭발적으로 증가하고 있다. 이러한 방대한 용량의 데이터 처리 요구사항을 만족시키기 위해서 최근 빅데이터 처리에 대한 연구 및 기술개발이 활발하게 진행되어 왔으며, 분산 컴퓨팅 프레임워크인 Hadoop을 기반으로 많은 빅데이터 분석 기술과 사업화가 이루어지고 있다.

빅데이터의 80% 이상은 직·간접적으로 공간정보를 포함하고 있는 것을 고려하면, 빅데이터 중에

This research was supported by the MOLIT(Ministry of Land, Infrastructure and Transport), Korea, under the national spatial information research program 'Geospatial Big Data Management, Analysis and Service Platform Technology Development(16NSIP-B081011-03)', supervised by the KAIA(Korea Agency for Infrastructure Technology Advancement).
* Corresponding Author : Electronics and Telecommunications Research Institute (ihjoo@etri.re.kr)
Received November 25, 2016 Revised December 15, 2016 Accepted February 22, 2017

서도 특히 공간 빅데이터 처리 기술에 대한 관심과 요구가 급증할 것으로 예상된다. 공간 빅데이터는 2차원 이상의 좌표를 가지는 공간정보와 융합된 다양한 속성정보로 이루어진 데이터이며. 일반적인 빅데이터 와 마찬가지로 기존 방식으로 저장, 관리, 분석할 수 있는 범위를 초과하는 대규모 공간 데이터를 말한다.

기존의 공간 데이터는 전통적으로 GIS (Geographic Information System) 또는 공간 DBMS를 통해 관리되고 분석되어 왔다. 그러나 공간정보의 활용분야가 다양해짐에 따라 센서 데이터, 이동객체 데이터, u-City 데이터, 위치기반 SNS 데이터 등 새로운 형태의 공간 빅데이터가 급속하게 증가되고 방대한 양의 공간 데이터를 신속하게 분석 및 처리할 필요가 있는 응용과 서비스가 증가하게 되자, 기존 GIS 또는 공간 DBMS로는 관리할 수 없는 대규모 데이터의 처리 방법이 필요하게 되었다.

그러나 기존 빅데이터 연구 분야의 결과는 대부분 공간 데이터를 단순하게 텍스트 혹은 숫자 타입의 비공간 데이터와 같은 방법으로 다루기 때문에 공간 데이터의 검색이나 분석에 있어 효율 및 성능 문제가 발생한다.

이러한 공간 빅데이터의 효율적 처리, 분석, 활용에 대한 요구에 따라 Hadoop 환경에서 공간 데이터를 처리할 수 있는 공간 빅데이터 처리 기술들이 연구개발되기 시작하였다. Minnesota Univ.에서 2013년 8월에 발표한 'Spatial Hadoop'은 Hadoop을 기반으로 공간 데이터를 지원한다[1]. Hadoop에서 MapReduce를 이용하여 프로그램을 작성하는 것과 유사한 방식으로 동작하며 공간 데이터를 다룰 때 Hadoop보다 우수한 성능을 보인다. Pig 를 확장한 Pigeon 언어를 제공하며 영역질의, kNN, 공간 조인, 최단경로 등의 연산을 사용할 수 있다. 미국의 Emory Univ.와 Ohio State Univ.는 공동으로 대규모의 공간 질의를 Hadoop에서 실행하기 위한 확장성있고 고성능의 공간 데이터 웨어하우징 시스템인 'Hadoop-GIS'를 2012년 10월에 발표하였다[2]. Hadoop-GIS는 공간 분할 기법에 의하여 MapReduce 상에서 공간질의를 병렬처리한다. 그러나 이들은 on-top 방식으로 공간 데이터를 지원하도록 확장하는 방법이며 SQL(Structured Query Language)과 같은 고급 질의 기능이 부족하다.

한편 Hadoop 환경에서 응용 개발자들이 복잡한 MapReduce 프로그래밍을 익히지 않고도 기존 시스템과 같은 방식으로 친숙한 SQL을 사용할 수 있게 하기 위해 SQL 유사 언어를 지원하는 Hive, Impala, Presto, Tajo 등의 SQL-On-Hadoop 솔루션들이 개발되었다[3][4]. SQL-on-Hadoop은 Hadoop에 저장된 데이터 파일을 개발자 또는 분석가에게 친근한 인터페이스인 SQL을 이용하여 분산 환경에서 질의를 실행함으로써 DBMS에서 SQL에 의해 질의하는 것과 같은 방법으로 질의할 수 있는 솔루션들을 총칭하는 말이다. 이러한 솔루션들은 SQL과 거의 유사한 언어를 사용하여 Hadoop 환경에서 데이터를 저장하고 분석할 수 있게 해주고 있으나, 공간 데이터를 직접 지원하지 않는다.

본 논문에서는 이러한 문제를 해결하기 위하여, 기존의 공간 데이터 처리 기술과 빅데이터 처리 기술을 접목하여 Hadoop 환경에서 대규모의 공간 빅데이터를 효율적으로 저장관리할 수 있고 SQL 기반 질의언어로 공간 질의를 할 수 있는 공간 빅데이터 질의처리 시스템을 제시하고자 한다.

2. 공간 빅데이터 질의처리 시스템

2.1 개요

본 논문에서 제시하는 공간 빅데이터 질의처리 시스템은 공간 RDBMS에서 SQL에 공간 데이터 기능을 확장하는 것과 유사하게 확장된 SQL 기반 질의언어에 의하여 Hadoop에 저장된 공간 데이터를 질의하는 시스템이다. 본 시스템의 목표는 다음과 같이 크게 두 가지로 요약된다.

Hadoop에 공간 데이터를 저장. 공간 데이터는 2차원 이상의 좌표들로 표현되며 텍스트나 숫자 데이터와는 다른 데이터 처리 기법이 필요하다. 이러한 기법들은 DBMS에서 공간 DBMS로 확장이 이루어지며 많이 연구되어 왔으나 Hadoop 환경에서는 적

용되지 못하였으며, 따라서 공간 데이터를 Hadoop에 저장하여도 효율적인 공간 데이터 처리가 이루어지지 못하였다. 따라서 공간 데이터를 효과적으로 Hadoop에 저장 및 처리하기 위한 방법이 필요하다.

사용자에 친숙한 질의언어 제공. Hadoop 환경에서 데이터를 처리하기 위하여 사용하는 MapReduce는 분산환경에서 대용량 데이터를 병렬적으로 처리하기에 적합하지만 개발자가 배우기에 어렵다는 단점이 있다. 따라서 복잡한 MapReduce 프로그래밍 방법을 배우지 않고도 익숙한 방법으로 Hadoop 환경에서 데이터를 처리하기 위한 방법이 요구된다.

이러한 목적을 달성하기 위하여, Hadoop 환경에서 공간 데이터에 대한 SQL 기반 언어에 의하여 공간 데이터를 처리하는 공간 빅데이터 질의처리 시스템을 개발하였으며, 공간 빅데이터를 위한 질의언어를 정의하고 질의언어 파싱, 질의 검증, 질의 계획 생성 등 질의처리의 핵심기능을 개발하였다.

2.2 구조

그림 1은 공간 빅데이터 질의처리 시스템의 구조이다. 본 시스템은 공간 빅데이터 저장 시스템에서 제공하는 API를 통하여 동작하고, 외부 프로그램에는 API와 인터페이스를 제공하는 구조를 가진다.

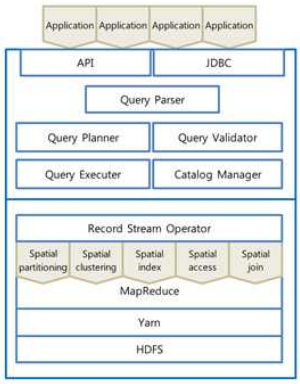


그림 1. 공간 빅데이터 질의처리 시스템의 구조
Fig. 1. Architecture of spatial big data query processing system

그림 2는 공간 빅데이터 질의처리 시스템이 주어진 질의언어로부터 질의처리를 수행하는 흐름을 나타내고 있다.

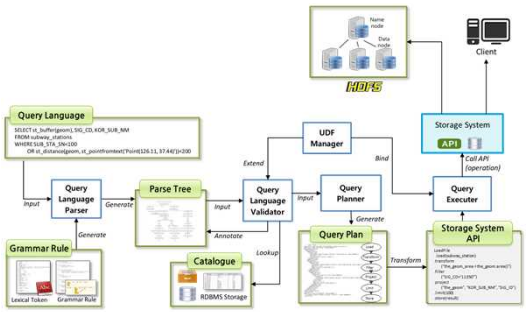


그림 2. 공간 빅데이터 질의처리 시스템의 처리 흐름
Fig. 2. Processing flow of spatial big data query processing system

2.3 데이터 모델

공간 빅데이터 질의처리 시스템은 RDBMS와 유사한 데이터 모델을 채택하며, 테이블, 칼럼, 레코드 등과 같은 RDBMS에서 사용하는 용어들을 사용한다. 공간 데이터를 처리하기 위한 데이터 타입, 함수, 입출력 함수 등은 [5] 및 [6] 에서 정의된 OGC(Open Geospatial Consortium)의 심플 피처(Simple Feature) 모델 및 SQL 프로파일에 따라서 정의하였다. 또한 일반적인 RDBMS와 유사한 기본 데이터 타입을 지원하며, 공간 데이터에 대한 표준 데이터 타입 및 함수를 추가적으로 지원한다. 질의언어에서 지원하는 데이터 타입은 다음과 같다.

- 기본 데이터 타입 : `Byte, Short, Int, Long, Double, String, DateTime, Binary`
- 공간 데이터 타입 : `Point, LineString, Polygon, MultiPoint, MultiLineString, MultiPolygon`

2.4 질의언어

SQL 기반 언어에 공간 데이터 타입, 공간 함수, 공간객체 표준 텍스트 표현인 WKT(Well-Known Text) 명세 등 공간 데이터 관련 요소들을 확장하여 공간 질의언어를 정의하였다.

데이터 선택. `SELECT/FROM/ WHERE` 절에 의한 기본적인 데이터 검색 및 선택을 지원한다. `ORDER BY` 문에 의하여 공간 데이터 타입을 제외한 주어진

칼럼에 대하여 레코드를 정렬하는 기능을 지원한다. GROUP BY에 의하여 레코드의 그룹핑이 가능하며, 각 그룹에 대한 속성값의 집계함수(COUNT, SUM, MIN, MAX, AVG)을 사용할 수 있다.

공간함수 지원. 질의언어에 내장되어 사용할 수 있는 공간 함수는 표 1과 같으며, SELECT 절의 함수 및 WHERE 절의 조건절에 포함될 수 있다. SELECT 절에 포함된 공간함수는 반환타입이 공간 데이터 타입인 경우 ST_AsText 또는 ST_AsBinary 의 export 함수와 같이 사용되며 WHERE 절에는 조건절에 다른 함수 표현식과 동일하게 공간함수가 사용될 수 있다.

표 1. 질의언어에서 지원되는 공간 함수
Table 1. Spatial functions supported in query language

Category	Spatial Functions
Import Functions	ST_PointFromText, ST_LineStringFromText, ST_PolygonFromText, ST_MultiPointFromText, ST_MultiLineStringFromText, ST_MultiPolygonFromText
Export Functions	ST_AsText, ST_AsBinary
Basic Spatial Functions	ST_Area, ST_Length, ST_Distance,ST_Envelop, ST_Boundary, ST_Centroid, ST_Buffer, ST_Convexhull
Relational Functions	ST_Intersect, ST_Contains, ST_Overlaps, ST_Within, ST_Disjoint, ST_Crosses, ST_Touches, ST_Equals
Set Functions	ST_Union, ST_Intersection, ST_Difference

공간 데이터 타입의 객체를 생성하는 ST_PointFromText 와 같은 import 함수에서는 WKT 표현을 사용한다. 공간 함수가 포함된 질의문의 예시는 다음과 같다.

```
SELECT                                ST_AREA(geom),
ST_ASTEXT(ST_BUFFER(geom))
FROM cadastral
WHERE ST_CONTAINS(geom, ST_POINTFROMTEXT
       ('Point(408655.51 331492.12)'))=1
```

데이터 삽입. 저장소로 사용하는 HDFS의 특성상 기존 존재하는 테이블에 대한 레코드의 insertion, deletion 및 update는 직접 지원하지 않는다. 질의언어에서는 기존 테이블의 데이터로부터 SELECT 절에 의하여 전체선택하거나 WHERE 절의 조건절에 의해 일부선택한 데이터를 새로운 데이터 셋으로 저장하는 기능을 다음 예와 같이 지원한다.

```
INSERT INTO new_table
SELECT *
FROM org_table
WHERE col1 > 200 AND ST_AREA(geom)>50000
```

공간 빅데이터 질의처리 시스템에서 사용되는 질의언어에서 지원하는 질의문장 및 예는 표 2와 같다.

표 2. 질의언어에서 지원하는 질의 문장
Table 2. Query sentences supported by the query language

Category	Query statement(example)
Records selection	SELECT * FROM cadastral
Filter by predicate	SELECT * FROM buildings WHERE EMD_CD='11530' AND BSI_INT_SM<60
Filter by spatial predicate	SELECT * FROM cadastral WHERE ST_AREA(geom) <500 OR BCHK='1'
Spatial function	SELECT ST_AREA(geom), ST_ASTEXT(ST_CONVEXHULL(geom)) FROM cadastral
Sort	SELECT * FROM political_emd ORDER BY EMD_NM DESC
Aggregation function	SELECT CTY_NM, SUM(POP) FROM workers GROUP BY CTY_NM
Limit record count	SELECT * FROM cadastral LIMIT 100
Import spatial data	SELECT * FROM cadastral WHERE ST_DISTANCE (geom, ST_POINTFROMTEXT('Point(408623.3372 331002.4397)'))>500
Export spatial data	SELECT ST_ASTEXT(geom) FROM cadastral
Dataset insertion	INSERT INTO new_subway SELECT * FROM subway_stations WHERE sub_sta_sn<100

2.5 질의언어 파서(Parser)

질의언어 파서는 파서(parser) 생성기인 ANTLR를 사용하여 개발되었다. ANTLR의 문법 포맷인 Lexer.g4와 Parser.g4 두 개의 파일로 문법을 정의하고 이를 ANTLR 4.5 라이브러리에 입력함으로써 파서를 생성하였다.

질의언어 파싱 단계에서는 각 문법 규칙(rule)마다 tree visitor라고 하는 모듈이 대응되며 이 visitor들마다 해당 문법 규칙을 만날 때 수행해야 하는 action을 정의한다. 질의언어 파서는 질의언어로 작성된 입력 문장을 파싱하여 주어진 문장이 문법에 적합한지를 검사하는 syntax 분석을 수행하고 파스 트리(parse tree)를 생성한다.

2.6 질의언어 검증기(Query Validator)

질의언어를 파싱하여 생성된 파스 트리에 대하

여 문법상으로 오류가 없는 질의 문장이 의미적 오류를 포함하고 있는지 검사한다.

질의언어 각 요소들의 의미를 검증하기 위하여는 데이터셋의 목록, 테이블 내의 스키마 정보 등에 대한 정보가 필요하다. 이러한 저장소에 있는 데이터셋의 메타데이터는 카탈로그 관리자(Catalog Manager)가 관리한다. 메타데이터를 위한 저장소로는 RDBMS인 PostgreSQL을 사용한다. 카탈로그는 공간 데이터를 나타내는 칼럼의 이름, 좌표계 정보, MBR(Minimum Bounding Rectangle) 정보를 포함한다. 질의언어 검증기는 카탈로그를 통하여 테이블 이름, 칼럼 이름 및 칼럼 데이터 타입 등 파스 트리의 각 요소에 오류가 없는지 의미를 검증한다. 공간 객체(geometry) 칼럼을 포함하고 있는 `SELECT` 절 및 `WHERE` 절에 오는 공간객체에 대한 함수들에 대한 검증도 수행한다.

`SELECT` 절이나 `WHERE` 절에는 단일 칼럼 이름 외에 함수, 표현식(expression), 별명(alias) 및 이들이 복합된 요소가 올 수 있는데 이러한 요소들의 타입 검증 과정에서 각 요소의 타입을 파스 트리의 해당 요소에 annotate하여 이후 단계에서 오류 검사 및 데이터 처리에 활용한다.

2.7 질의계획 생성기(Query Planner)

질의계획 생성기는 파스 트리로부터 질의처리의 단위 연산의 논리적 실행순서를 질의계획(query plan)으로 수립하여 연산 트리(operation tree)를 생성한다. 질의계획 생성은 데이터 접근을 최소화하기 위한 최적의 실행순서를 찾는다.

확장성을 위하여 연산 트리는 저장소 독립적인 논리적 구조로 생성되며, *Operation*이라는 node와 node간의 연결관계로 나타나는 트리 구조로 표현된다. 연산 트리를 사용자 또는 개발자에게 리포트할 경우나 모듈간에 교환하는 경우 미리 정의된 표현방법이 필요하며 본 시스템에서는 XML 기반의 표현을 사용하였다.

2.8 질의 실행기(Query Executer)

질의계획이 생성되면 저장시스템에 접근하여 실제 데이터를 가져오기 위하여 논리적 실행계획인 연산 트리를 저장시스템 단위연산 API로 변환하고 실행한다. 데이터 로딩, filter, projection, 저장 등 질의처리 기본 기능으로 지원되는 기능에 대하여 API 실행 기능이 대응되어 지원된다.

이러한 API의 실행은 다음 예와 같이 *PlanBuilder* 클래스 객체를 생성하고 생성된 객체의 메서드를 순차적으로 호출하는 형태로 수행된다.

```
marmot.PlanBuilder builder =
       marmot.Plan.builder();
builder.loadLayer("layerName")
      .project("colname1", "colname2")
      .filter("colname1<200")
   …;
```

질의언어는 연산트리로 생성된 뒤 최종적으로 저장시스템 API 메서드들의 순차적인 호출을 조합한 형태로 변환된다(표 3 예시).

표 3. 질의언어와 API의 대응
Table 3. Query language and corresponding API

Query language	INSERT INTO result SELECT the_geom, SUB_NM, SIG_ID, st_area(the_geom) FROM subway_station WHERE SIG_CD='11350' ORDER BY KOR_SUB_NM LIMIT 100
API	LoadFile .loadLayer(subway_station) .transform("geom_area= the_geom.area()") .filter("SIG_CD='11350'") .project("the_geom","SUB_NM", "SIG_ID") .sort("KOR_SUB_NM", SortOrder.ASC) .limit(100) .storeLayer(result)

2.9 저장 시스템(Storage System)

공간 빅데이터 저장시스템은 질의처리 시스템의 기반이 되는 저장 시스템이며, Hadoop 환경에서 분산 처리를 위한 데이터의 저장 및 접근 기능을 제공한다. 공간 데이터를 포함하는 데이터셋은 레이어(layer) 단위로 HDFS상의 파일로 저장하며, 기본적으로 MapReduce 방식을 통하여 데이터를 처리한다. 레코드 스트림 기반 처리 구조를 가지며 단위 기능은 표 4와 같은 연산자(operator)가 API 형태로 제공된다. 연산자의 조합을 통하여 다양한 형태의 데이터 접근 기능을 구현할 수 있다. 단위 기능의 모듈화에 따라서 재사용성 및 확장성을 가진다. 그림

3은 연산자를 조합하여 순서대로 실행하여 결과를 출력하는 실행 흐름의 예를 개념적으로 나타낸다.

표 4. 저장시스템의 레코드 스트림 연산자
Table 4. Record stream operator of storage system

Category	Operator
Unary	Aggregate, Filter, Project, GroupByAggregate, Distinct, Skip, Limit, NoOp, Shard, Transfrom
Loader	LoadMarmotFile, LoadTextFile
Collector	Store, StoreAsCsv
Spatial	AttachMapFile, Buffer, BBox, AggrUnion, AggrConvexHull, ConvertCoordinates, Intersection
Spatial loader	LoadShapeFile
Spatial collector	StoreShapeFile

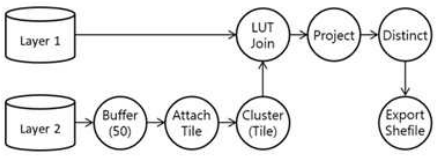


그림 3. 단위 연산의 실행 흐름
Fig. 3. Execution flow of unit operations

연산자의 조합으로 이루어진 논리적 연산 흐름은 Hadoop에서 수행하기 위한 MapReduce 작업으로 자동으로 변환되어 Hadoop 프레임워크 내에서 실행됨으로써 HDFS에 저장된 데이터 파일에 대한 접근 및 처리가 가능하다. 이러한 연산자 기반 처리 구조는 연산자를 사용하는 개발자가 MapReduce에 대한 지식이 필요없다는 장점을 가진다.

공간 빅데이터 질의처리 시스템과 저장시스템 API와의 연결은 스트림 처리 방식으로 이루어진다. 질의를 실행하면 질의 결과 데이터셋에 대한 스트림을 가지고 있는 *Stream* 객체가 반환되며 레코드의 처음에서 시작하여 다음 데이터를 가져오기 위한 함수를 반복하여 호출하여 전체 레코드를 가져오는 방법으로 동작한다.

3. 결과

본 시스템의 기능을 실험하기 위하여 공간 데이터를 Hadoop 클러스터에 저장하고 클라이언트 UI를 통하여 질의문장을 입력하여 시스템의 동작을 확인하였다. 시스템의 질의처리 성능은 기존 유사 시스템인 Spatial Hadoop 시스템과 비교하였다.

3.1 실험 환경

본 시스템의 실험은 6개의 노드로 구성된 Hadoop 클러스터에서 수행되었다. 각 노드는 4.0Ghz 4 core CPU, 32GB 메인메모리, 4TB 디스크를 가지는 데스크탑 PC이며 OS는 CentOS 6.7, Hadoop 버전은 Hortonworks HDP 2.3.2이다.

실험에서는 대용량 지도로서 지적도, 지적중심점, 건물 등 3가지 공간 데이터셋을 사용하였다(표 5). Shape 파일을 별도의 loader 프로그램으로 두 시스템에 맞는 포맷으로 각각 변환하여 저장하였다.

표 5. 실험에 사용된 데이터셋
Table 5. Datasets used in the tests

dataset	number of records	size	spatial data type	source format
cadastral	about 38 billions	16GB	Polygon	shape
cadastral_c	about 38 billions	5GB	Point	shape
buildings	about 660 thousands	170MB	Polygon	shape

3.2 클라이언트 프로그램

공간 빅데이터 질의처리 시스템으로 공간질의를 실행하고 결과를 확인하기 위하여 웹 기반 클라이언트를 사용하였다(그림 4). 질의언어를 입력할 때 공간객체의 좌표를 입력할 경우는 지도기반 UI를 사용하여 화면에서 입력한 좌표를 자동으로 WKT 형태로 생성하여 사용하였다. 지도 및 질의결과의 출력을 위하여 공간정보 관련 open API를 제공하는 공간정보오픈플랫폼(V-World) 의 API [7] 를 활용하였다.

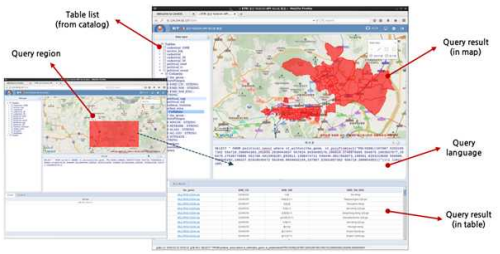


그림 4. 클라이언트 사용자 인터페이스
Fig. 4. Client user interface

3.3 성능 비교

공간 빅데이터 질의처리 시스템(표 및 그림에서 'SBQ'로 표기함)의 성능을 Minnesota Univ.의 Spatial Hadoop('SH'로 표기함)과 비교하였다. 두 시스템에 대하여 공간질의의 대표적인 영역질의(region query)의 질의 실행시간을 비교하였다. 영역질의는 주어진 직사각형 영역과 겹치는(Intesects) 객체를 찾는 질의로 실행하였으며, 질의영역을 표 6과 같이 지도 데이터 전체영역에 대한 상대적 크기에 따라 4가지로 구분하여 실행하였다.

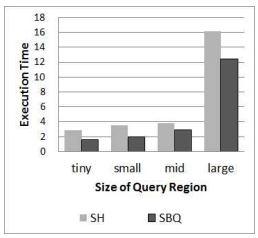
표 6. 실험에 사용된 질의영역
Table 6. Query regions used in the tests

region	average size (in kilometer)	ratio of size to overall region (%)
tiny region	0.75 X 0.58	0.00016
small region	3 X 2.3	0.00247
medium region	6 X 4.5	0.00968
large region	24 X 18	0.15484

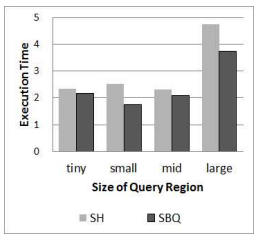
표 7 및 그림 5는 공간 빅데이터 질의처리 시스템과 Spatial Hadoop의 질의실행시간 비교 결과이다. 질의영역의 크기와 데이터셋에 따라 차이가 있으나 공간 빅데이터 질의처리 시스템은 Spatial Hadoop 시스템보다 빠른 질의실행시간을 나타냈으며 평균 약 58%의 성능향상을 보이는 것으로 나타났다.

표 7. 공간 빅데이터 질의처리 시스템 영역질의 실행 시간
Table 7. Region query execution time of spatial big data query processing system

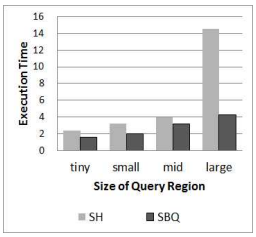
dataset	*cadastral*		*cadastral_c*		*buildings*	
	SH	SBQ	SH	SBQ	SH	SBQ
tiny region	2.91	1.53	2.34	2.16	2.34	1.55
small region	3.50	1.95	2.52	1.76	3.19	2.03
medium region	3.82	2.85	2.31	2.09	4.00	3.16
large region	16.17	12.42	4.75	3.75	14.50	4.27



(a) Time for *cadastral*



(b) Time for *cadastral_c*



(c) Time for *buildings*

그림 5. 공간 빅데이터 질의처리 시스템 성능 비교
Fig. 5. Performance comparison of spatial big data query processing system

본 시스템은 질의언어 기반으로 기능을 제공하는 반면 Spatial Hadoop은 명령어(command) 형태로 기능을 제공하므로 본 시스템은 Spatial Hadoop 시스템에 비하여 질의언어 파싱, 검증, 질의계획 생성 등 질의언어 처리 과정이 추가적으로 필요하다. 이러한 질의언어 처리 시간은 실험에 사용한 영역질의에 대하여 질의영역의 크기와 관계없이 약 0.1초가 소요되었다. 표 7의 실행시간은 이러한 질의언어 처리 시간을 포함한 시간이다.

4. 결론

본 논문에서는 Hadoop 환경에서 저장된 공간 데이터를 공간 데이터 처리 기능이 확장된 SQL기반 질의언어에 의하여 질의할 수 있는 공간 빅데이터 질의처리 시스템을 제시하였다. 공간 빅데이터 질의처리를 위한 질의언어 파서, 질의언어 검증기, 질의계획 생성기, 저장시스템 연동 등 질의처리 기본기능을 개발하였다.

개발한 시스템의 질의처리 성능을 기존 시스템인 Spatial Hadoop과 비교하였으며 공간 데이터에 대한 영역질의의 질의실행시간에 있어서 비교 시스템 대비 약 58% 의 성능향상을 보이는 것으로 나타났다.

공간 빅데이터 질의처리 시스템은 사용자에게 익숙한 SQL 기반 질의언어를 사용함으로써 공간 빅데이터를 활용하는 응용 개발자가 기존 공간 RDBMS 등의 시스템으로부터 Hadoop 기반 시스템으로 전환하여 상위수준의 분석 및 서비스를 쉽

게 개발할 수 있게 해줄 것으로 기대된다. 이를 위하여 향후 시스템이 추가적으로 지원할 내용 및 연구방향으로는, 첫째 공간 데이터 처리에서 중요한 기능 중의 하나인 공간 조인(spatial join)을 지원하는 것이다. 이를 위하여 공간 조인을 지원하는 질의언어 처리가 필요하며, 매우 많은 데이터 접근 및 연산이 필요한 공간 조인의 효율적인 수행을 위하여 공간 색인(spatial index)을 포함한 Hadoop 환경과 MapReduce 모델에 적합한 처리 기법에 대한 연구가 중요하다. 둘째로는 MapReduce 처리 방법의 특성, 공간 질의의 종류별 특성, 데이터셋의 특성 등을 고려하여 Hadoop 분산 환경에 적합한 질의 최적화 기법을 연구하여 적용하는 것이라고 할 수 있다.

REFERENCES

[1] Ahmed Eldawy & Mohamed F. Mokbel, “A Demonstration of SpatialHadoop An Efficient MapReduce Framework for Spatial Data”, *Proceedings of the VLDB Endowment*, 6(12), pp.1230-1233, 2013.

[2] Ablimit Aji, Fusheng Wang, Hoang Vo, Rubao Lee, Qiaoling Liu, Xiaodong Zhang & Joel Salt, “Hadoop-GIS: A High Performance Spatial Data Warehousing System over MapReduce”, *Proceedings of the VLDB Endowment,* 6(11), pp.1009-1020, 2013.

[3] Ashish Thusoo, Joydeep Sen Sarma, Namit Jain, Zheng Shao, Prasad Chakka, Ning Zhang, Suresh Antony, Hao Liu, & Raghotham Murthy, “Hive - a Petabyte Scale Data Warehouse using Hadoop”, *2010 IEEE 26th International Conference on Data Engineering(ICDE 2010)*, pp.996-1005, 2010.

[4] Choi, H. S., Son, J. H., Yang, H. M., Ryu, H. S., Lim, B. N., Kim, S. H., & Chung. Y. D, “Tajo: A distributed data warehouse system on large clusters”, *Data Engineering (ICDE), 2013 IEEE 29th International Conference*, pp.1320-1323, 2013.

[5] Open Geospatial Consortium Inc, OGC 06-103r4 “OpenGIS® Implementation Standard for Geographic information - Simple feature access - Part 1: Common architecture”, 2011.

[6] Open Geospatial Consortium Inc, OGC 05-134 “OpenGIS® Implementation Specification for Geographic information - Simple feature access - Part 2: SQL option”, 2005.

[7] Choi, W. G., Kim, M. S., Jang, I. S., & Chang, Y. S., “The Comparative Research on 2D Web Mapping Open API for Designing Geo-spatial Open Platform”, *Journal of Korea Spatial Information Society*, 22(5), pp.87 - 98, 2014.

저자약력

주 인 학 In-Hak Joo) **[정회원]**

- 1994년 2월 : 연세대학교 컴퓨터과학과 (석사)
- 2000년 8월 : 연세대학교 컴퓨터과학과 (박사)
- 2000년 9월 ~ 2012년 2월 : 한국전자통신연구원 선임연구원
- 2012년 3월 ~ 현재 : 한국전자통신연구원 책임연구원

<관심분야> 공간정보, GIS, 빅데이터