맵리듀스 함수 지원을 위한 SQL 질의의 확장 방법

정문영, 이태휘, 김성수, 원종호
한국전자통신연구원
e-mail : {mchung, taewhi, sungsoo, jhwon}@etri.re.kr

SQL Extension for Supporting MapReduce Functions

Moonyoung Chung, Taewhi Lee, Sung-soo Kim, Jongho Won
Electronics and Telecommunications Research Institute

요 약

SQL 질의와 분산 처리를 위한 맵리듀스 함수를 통합 제공하면 쉽고 인터렉티브한 SQL 질의에서 맵리듀스 프로그래밍의 풍부한 표현력을 이용할 수 있다. 본 논문에서는 SQL 질의와 맵리듀스 함수를 통합하기 위해서 확장연산자를 이용하여 SQL 질의를 확장하는 방법을 제안한다.

1. 서론

맵리듀스는 맵과 리듀스라는 함수로 구성된 프로그래밍 프레임웍으로, 맵 함수와 리듀스 함수의 분산된 프로세싱을 허용함으로써 여러 컴퓨터에 분산되어 저장되어 있는 대용량 데이터를 빠르게 처리할 수 있다. 그러나, 맵리듀스 프로그램은 대용량 데이터를 배치 처리하는 작업에 적합하며, 복잡한 알고리즘을 구현하기 위해서는 여러 번의 맵리듀스 작업을 반복해야 하는 불편함과 그에 따른 성능 저하가 많다. 따라서, 인터랙티브하고 신속한 결과를 요구하는 분석에서는 한계가 있다. 또한, 데이터 분석가나 데이터 과학자와 같은 사용자에게는 맵리듀스 프로그래밍이 어려워 여전히 복잡한 분석을 위해서는 프로그래머에게 의지해야 하는 단점이 있다.

따라서 이러한 사용자에게 익숙한 언어인 SQL 을 이용하여 하둡 분산 파일 시스템에 저장된 데이터를 인터렉티브한 방법으로 신속하게 분석할 수 있도록 SQL-on-Hadoop 시스템이 등장하게 되었다. 아파치 하이브(Apache Hive)[1]는 SQL 과 유사한 HiveQL 이라는 언어로 질의를 작성하면 이를 맵리듀스로 변환하여 처리한다. 아파치 타조[2], 클라우데라 임팔라[3] 등은 ANSI SQL 로 정의된 SQL 질의를 맵리듀스로 변환하지 않고 분산 처리한다.

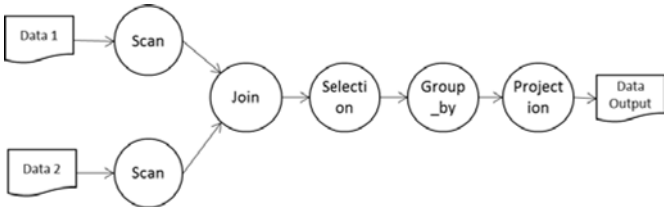
SQL-on-Hadoop 에서 지원하는 SQL 문법은 임의 질의를 쉽게 작성하고 빠르게 처리하는 데는 적합하지만, 데이터 마이닝이나 통계분석과 같은 복잡한 알고리즘을 표현하기에는 오히려 어렵다는 단점이 있다. 이러한 복잡한 알고리즘을 표현하기에는 오히려 맵리듀스 프로그램이 더 적합할 수 있으며, 이미 Mahout[4]과 같은 다양한 레거시 맵리듀스 라이브러리들이 제공되고 있다. 따라서, 한편에서는 SQL 과 맵리듀스 통합을 위한 질의 인터페이스를 제공하려는 연구가 진행되고 있다[5].

본 논문에서는 SQL 질의와 맵리듀스 함수의 처리 방법을 파악하고 이를 토대로 SQL 질의에서 맵리듀스 함수를 통합하여 제공하는 방법을 제안한다.

2. 분산 시스템에서 맵리듀스와 SQL 질의 분석

SQL 질의에서 맵리듀스 함수를 지원하기 위해서 분산 시스템에서 SQL 질의와 맵리듀스 함수가 처리되는 방법을 분석한다.

SQL 질의는 파싱, 실행계획 생성, 최적화 과정을 거쳐 분산 실행계획을 작성하게 된다. 예를 들어, 그림 1 과 같이 생성된 SQL 질의 실행계획을 보면, “Scan” 연산에서 파일시스템으로부터 데이터셋를 읽어서 “Join”, “Selection”, “Group by”, “Projection”의 연산을 수행하고 결과 데이터를 파일시스템에 저장한다. 이 과정에서 각 연산에서 처리된 중간 결과는 다음 연산을 위해 파일시스템에 저장되기도 한다.

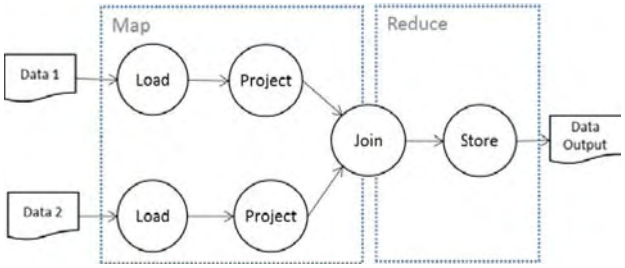


(그림 1) SQL 질의 실행계획의 예

맵리듀스의 맵(Map)과 리듀스(Reduce)를 처리하는 과정에서도 비슷한 데이터 흐름을 볼 수 있다. 그림 2 와 같이 맵리듀스도 파일시스템으로부터 데이터를 로드하여 맵 함수와 리듀스 함수를 실행하고 결과 데이터를 파일시스템에 저장한다. 이 과정에서 중간 결과를 로컬 파일시스템에 저장하기도 하며 맵 함수가 분산 파일시스템에 저장한 데이터를 읽어 리듀스 함수를 실행하고, 최종 결과 데이터를 분산 파일시스템에 저장한다.

SQL 질의와 맵리듀스 질의 처리 과정을 분석해보

면, 파일시스템으로부터 데이터를 읽어서 처리하고 최종 결과 데이터를 파일시스템에 저장하는 공통점이 있으며, 각 연산자는 파일시스템으로부터 데이터를 읽어서 처리하고, 처리된 중간 혹은 최종 결과 데이터를 파일시스템에 저장한다.

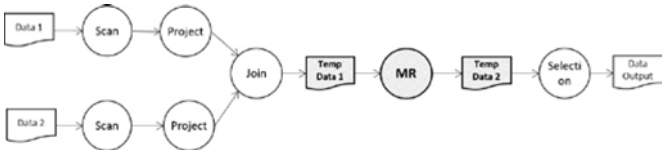


(그림 2) MapReduce 질의 실행플랜의 예

3. SQL 확장 질의 처리를 위한 질의 플랜 생성과 확장연산자의 구현

SQL-on-Hadoop 시스템에서는 SQL 질의의 질의 플랜을 분산 처리하기 위해서 각 연산자를 한 개 이상의 태스크로 나누어 처리한다. 태스크의 종류에 따라 데이터의 임시 저장이 필요한 경우 파일시스템에 중간 결과를 저장하게 된다. 태스크 실행 시에 입력과 출력은 전후 태스크의 입력과 출력에 의존적이지만, 실행 로직은 각각 독립적이다. 이러한 특성을 이용하여 맵리듀스 함수를 하나의 연산자로 이용하는 SQL 확장 방법을 제안한다.

맵리듀스 함수의 입력과 출력을 SQL 연산에서 처리할 수 있는 형태로 맞춰준다면, SQL 질의의 중간에 맵리듀스 함수를 태스크(혹은 연산자)로 삽입하여 처리할 수 있다. 아래 그림 3 의 SQL 질의에서 “join” 연산 후에 중간 결과 데이터(Temp Data1)를 파일시스템에 저장하고, 맵리듀스 연산에서 이 중간 데이터를 읽어서 처리한 후 다시 중간 결과 데이터(Temp Data2)를 “selection” 연산의 입력으로 사용하여 처리할 수 있다.

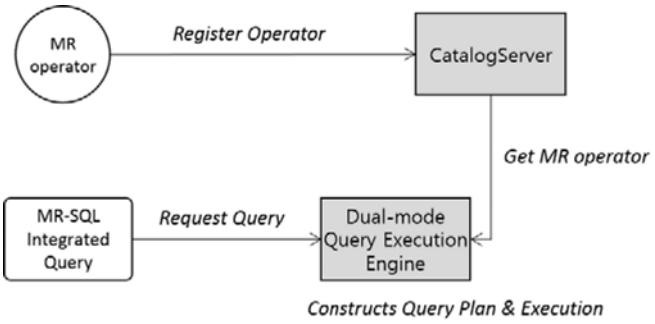


(그림 3) 통합 질의 실행계획의 예

위와 같은 질의 실행 계획을 생성하기 위해서는 SQL 질의에서 맵리듀스 함수를 호출할 수 있는 형태로 정의하여 처리하여야 하는데, 본 논문에서는 “확장연산자(Extended Operator)”라는 개념을 도입한다. 확장연산자란 SQL 문법에서 미리 정의되지 않은 연산자로, SQL 질의에서 사용할 수 있는 연산자를 말하며 데이터의 입력이나 출력이 튜플 셋으로 표현될 수 있어야 한다.

확장연산자는 SQL-on-Hadoop 시스템의 카탈로그에 미리 등록되어 있어야 한다. SQL 질의에서 확장연산자가 사용되면, 질의 처리 엔진에서는 카탈로그에서 확장연산자를 참조하여 질의를 파싱하고 질의 실행계획을 생성하게 된다.

그림 4 는 확장연산자를 등록하는 과정과 확장연산자를 사용하는 SQL 질의가 처리되는 과정을 보여준다. 맵리듀스 함수와 같은 확장연산자를 이름, 입력 형식, 출력 형식, 라이브러리 경로, 파라미터 등과 함께 정의하여 카탈로그에 등록한다. SQL 질의가 확장연산자를 이용하면, 이에 대한 정보를 카탈로그에서 가져와서 질의 실행계획을 생성하고 실행한다.



(그림 4) 확장연산자의 등록과 실행

그림 5 는 “Grep” 을 하는 맵리듀스 함수를 확장연산자로 정의한 예를 보여준다. LIBRARY 의 ‘hadoop-mapreduce-examples-2.6.0.jar’ 맵리듀스 프로그램을 사용하며 입력으로 파일 형태를, 출력은 grep_output 이라는 테이블을 정의하여 사용할 것임을 알 수 있다. 이 정의는 DDL 로 처리되어 카탈로그에 등록된다.

```
CREATE EXTENDED OPERATOR grep
LIBRARY '/hadoop/hadoop-mapreduce-examples-2.6.0.jar'
USING MR
PARAMETER (ARG 'grep', IN FILE, OUT grep_output, ARG)
SCHEMA grep_output (count INT, key TEXT)
    USING text WITH ('text.delimiter'='\t');
```

(그림 5) 확장연산자의 정의

카탈로그에 등록된 확장연산자는 SQL 질의의 FROM 절이나 서브질의(Subquery)에서 호출된다. 그림 6 은 SQL 질의가 단일 연산자만 호출하는 경우의 문법으로, FROM 절에서 연산자를 호출하며, 생성된 연산자의 호출 방식에 따라 입력, 출력, 파라미터를 FROM 절에 기술하여 질의를 표현한다.

```
SELECT *
FROM grep('/input', out, 'dfs[a-z]+')
WHERE count > 10;
```

(그림 6) 질의에서 확장연산자 사용 예

4. 결론

본 논문에서는 SQL 질의와 맵리듀스 함수를 통합하기 위한 방법으로, 맵리듀스 함수를 연산자로 정의하고 이를 SQL 질의에서 호출하여 사용할 수 있도록 SQL 질의를 확장하는 방법을 제안하였다. 이를 SQL-on-Hadoop 시스템에 구현하여 이 방법이 실현 가능하고 유용함을 보였다.

Acknowledgment

이 논문은 ETRI R&D 프로그램("듀얼모드 배치-쿼리 분석을 제공하는 빅데이터 플랫폼 핵심 기술 개발, 15ZS1400")의 일환으로 수행됨.

참고문헌

[1] Thusoo, Ashish, et al. "Hive: a warehousing solution over a map-reduce framework." Proceedings of the VLDB Endowment 2.2 (2009): 1626-1629

[2] Choi, Hyunsik, et al. "Tajo: A distributed data warehouse system on large clusters." Data Engineering (ICDE), 2013 IEEE 29th International Conference on. IEEE, 2013.

[3] Cloudera Impala, http://www.cloudera.com/content/cloudera/en/products-and-services/cdh/impala.html

[4] Apache Mahout, http://mahout.apache.org/

[5] 이태휘, 정문영, 임동혁, 원종호, “SQL-맵리듀스 통합을 위한 질의 인터페이스 및 질의 최적화 연구 현황”, 한국정보처리학회 춘계학술대회, 2015.