SQL-on-Hadoop을 위한 벡터처리기반 질의 실행 엔진 설계

김성수°, 정문영, 이태휘, 송혜원, 원종호
한국전자통신연구원
{sungsoo, mchung, taewhi, hwonsong, jhwon}@etri.re.kr

Vectorized Query Execution Engine Design for SQL-on-Hadoop

Sung-Soo Kim°, Moonyoung Chung, Taewhi Lee, Hyewon Song, Jongho Won
ETRI

요 약

빅데이터가 보편화되면서 하둡기반 데이터 웨어하우스의 구축과 관리 및 질의 처리 등이 빅데이터기반의 비즈니스 인텔리전스 응용과 관련된 중요한 문제로 부상되고 있다. 이러한 빅데이터를 분석하여 필요한 의사결정 시점에 활용하기 위해서는 질의 처리 성능 개선이 중요한 문제다. 데이터 과학자는 의사결정이나 심층 분석를 위해, 하둡 분산 파일 시스템 (HDFS)에 저장된 데이터에 대해 SQL 질의를 통해 인터랙티브하게 처리할 수 있는 시스템을 사용하고 있다. 본 논문에서는 질의처리 성능 개선을 위해 데이터수준 병렬화를 적용한 벡터처리반 질의 실행 엔진에 대한 시스템 설계를 제안한다.

1 서론

빅데이터가 보편화되면서 하둡기반 데이터 웨어하우스의 구축과 관리 및 질의 처리 등이 빅데이터기반의 비즈니스 인텔리전스 응용과 관련된 중요한 문제로 부상되고 있다. 특히, 하둡기반의 데이터 웨어하우스 관리는 웨어하우스 크기와 복잡성에 비례하는 작업이며, 초기 데이터 탐색을 위해서는 인터랙티브한 질의를 활용한 분석이 필수적이다. 이러한 빅데이터를 분석하여 필요한 의사결정 시점에 활용하기 위해서는 질의 처리 성능 개선이 중요한 문제다.

하둡 기반의 데이터 웨어하우스에서 분석을 위한 질의 처리를 수행하는 SQL-on-Hadoop 시스템들이 각광 받고 있다. 이전에 데이터 분석가나 데이터 과학자는 하둡상의 데이터 분석을 위해 복잡한 맵리듀스 프로그램을 작성하는 데 어려움을 겪었다. 이러한 문제를 극복하기 위해, 초기 페이스북 (facebook)에서 SQL과 유사한 질의 (HiveQL)를 맵리듀스로 변환하여 수행할 수 있는 하이브 (Hive)를 소개하였다. 이와 같이, SQL-on-Hadoop이란 SQL 질의문을 통해 하둡의 HDFS에 저장된 데이터를 분석할 수 시스템을 말한다 [1]. 초기 하이브는 배치처리와 유연성에 초점을 두고 설계되었기 때문에, 인터랙티브한 질의처리를 위한 고려는 반영되어 있지 않았지만, 스팅거 (Stinger) 프로젝트를 통해 현재 (Hive 2.1.0)는 인터랙티브한 질의처리를 지원한다 [2]. 대표적인 SQL-on-Hadoop 시스템들로는 아파치 타조 (Apache Tajo), 아파치 임팔라 (Apache Impala), 아파치 드릴 (Apache Drill), 페이스북의 프레스토 (Presto), SparkSQL 등이 있다.

하둡 버전 1.0에서의 분산처리는 맵리듀스 (MapReduce) 프레임워크를 이용해서 처리했으나, 하둡 2.0 이후에는 자원관리는 YARN에서 수행하고, 분산처리 실행엔진으로 테즈 (Tez)나 스파크 (Spark) 엔진과 같은 DAG (Directed Acyclic Graph) 기반의 향상된 성능을 제공하는 프레임워크들을 사용한다. 본 논문에서는 벡터처리기반 질의실행 엔진 설계를 위해 DAG 기반 계산 프레임워크를 구현한 엔진을 사용하였다 [3, 4].

본 논문은 하둡 기반의 데이터 웨어하우스의 질의 처리 성능개선 측면에서 기여한 바를 요약하면 다음과 같다.

- *데이터-레벨 병렬화 (Data-Level Parallelism):* 자바 (Java)와 같이 CPU 명령어 수준 병렬화 (instruction-level parallelism)를 활용하기 어려운 경우에 적용할 수 있는 데이터-레벨의 SIMD 병렬화를 적용하여 성능을 개선할 수 있다.
- *에드혹 (ad-hoc) 질의처리:* 벡터처리기반 질의실행 엔진의 성능 개선을 통해, 데이터 과학자들이 초기 빅데이터에 대한 데이터 탐색 시에 유용한 인터랙티브한 에드혹 (ad-hoc) 질의를 제공한다.

이와 같이, 본 논문은 하둡기반 데이터 웨어하우스에서 빅데이터 통계 분석과 에드혹 질의 처리를 가속화하기 위한 데이터-레벨 병렬화를 적용한 벡터기반 질의실행 엔진 설계를 제안한다.

2 연구 배경

최근 멀티코어 CPU 하드웨어를 이용한 병렬처리 메커니즘들은 쓰레드 병렬화와 벡터 병렬화로 크게 두가지 주요한 메커니즘으로 구분할 수 있다. 벡터 병렬화 (Vector parallelism)은 복수개의 데이터 컬렉션에 대해 하나의 연산을 반복해서 수행하는 방법이다. 멀티코어기반의 최근 CPU에서는 벡터처리기반 저수준 CPU 명령어 (Intel SSE3, AVX2 등)를 제공하여 처리속도를 높이는 데 이용하고 있다. 하지만, 자바 가상머신에서는 이러한 CPU 벡터처리기반 연산 명령어를 활용하는 데 어려움이 있다.

일반적으로 SQL 질의처리는 질의 파싱, 논리적 실행 계획 수립, 논리적 실행 계획 최적화, 물리적 실행 계획 수립, 물리적 실행 계획 최적화, 질의 실행 순으로 처리한다. 기존 질의처리 성능 개선에 대해, 질의최적화와 연관된 규칙기반 최적화와 비용기반 질의최적화 연구들이 제안되었다.

대부분의 데이터베이스 엔진에서 질의 실행 계획은 스캔, 셀렉션, 조인과 같은 관계 대수와 해당하는 연산자들로 구성된 질의 실행 트리 (QET; query execution tree)로 표현된다. 질의 실행 계층에서 표준화된 open(), next(), close() 인터페이스에 기반한 이터레이터 (iterator) 모델을 사용한다. 초기화 작업은 open()에서 자원 반납은 close()에서 수행한다. 이 트리의 루트 노드에서부터 리프 노드로 next() 를 호출함으로써, 쿼리 트리내에서 튜플들을 뽑아낸다. 쿼리 플랜은 이와 같이 파이프라인 방식으로 동작하는 관계연산자들의 집합으로 구성된다.

전통적인 데이터베이스 관리시스템은 쿼리플랜의 연산자 노드 처리 시, 한번에 하나의 튜플을 처리하는 *tuple-at-a-time* 모델 ("*Volcano-style*" 모델이라고도 함)로 동작한다. 한번에 하나의 튜플을 처리하는 이 모델의 쿼리 인터프리테이션과 튜플 조작하는 데 더 많은 오버헤드 때문에 처리속도가 느리다는 단점이 있다. 이 방법은 CPU 명령어 파이프라인과 슈퍼스칼라 프로세서를 비효율적으로 사용하며 캐시 지역성도 떨어진다. 이러한 tuple-at-a-time 모델의 오버헤드를 피하기 위해, 한번에 하나의 로우블록을 처리하는 *block-at-a-time* 모델, 그리고 한번에 하나의 컬럼 전체를 처리하는 *column-at-a-time* 모델, 한번에 하나의 컬럼벡터(column vector)를 처리하는 *vector-at-a-time* 모델이 제안되었다. 대표적인 컬럼기반 데이터베이스인 MonetDB [5]가 column-at-a-time 방식을 사용하고, 특히 상용 제품인 Vectorwise [6]가 vector-at-a-time 방식으로 관련 상용 제품들과 대비하여 최고의 OLAP 처리 성능을 제공한다.

본 논문에서는 캐시 사용성 (cache utilization) 및 CPU 효율성 (efficiency) 증대를 통해 처리량을 향상시킬 수 있는 벡터처리기반 질의 실행 엔진에 대한 설계를 소개한다.

3 시스템 구조

하둡기반 데이터 웨어하우스에서 SQL 질의처리를 위해, 본 논문에서 제안하는 시스템 구조는 그림 1과 같다. 사용자는 SQL 인터페이스 블록을 통해, 표준 SQL 질의를 사용해서 원하는 질의를 요청한다. SQL 인터페이스 통해 전달받은 질의는 질의 파서가 AST (Abstract Syntax Tree) 형태의 파서 트리를 생성하고, 이 파서 트리를 쿼리 옵티마이저 (Query Optimizer)에서 논리적 질의 계획에 대한 최적화를 수행과 물리적 질의 계획에 대한 최적화를 수행한다. 도출된 물리적 질의 계획을 벡터처리기반 질의 실행 엔진에서 하둡의 HDFS에 저장되어 있는 데이터를 이용해서 질의를 실행한다. 벡터처리를 위한 최적 실행 계획은 *규칙기반 질의 최적화* (rule-based optimization) 과정에서 추가된다.

기본적인 아이디어는 컬럼벡터들의 배열 형태로된 캐시에 적재 가능한 크기의 로우 묶음(배치, batch)를 처리함으로써 처리성능을 높이자는 것이다. 이런 자료구조를 *벡터화된 로우 배치* (vectorized row batch) 라고 정의한다. 하둡내 데이터를 벡터화된 로우 배치로 로딩하기 위해 RC 파일 (Record Columnar File) 또는 ORC 파일

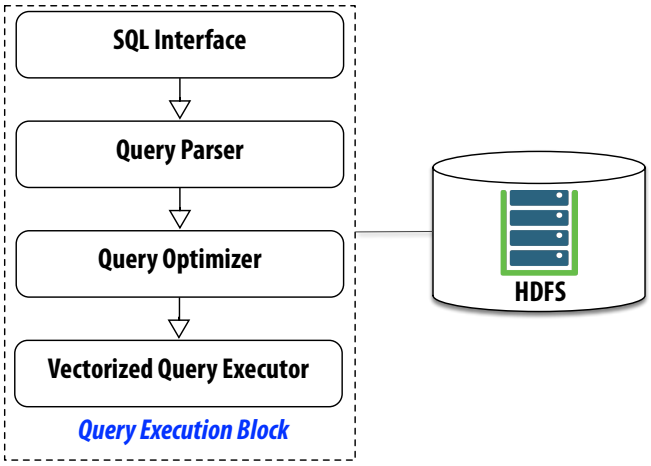

그림 1: 하둡기반 SQL 질의 처리 흐름도

(Optimized Record Columnar File) 과 같은 컬럼기반 데이터 포맷을 사용하는 것이 더 효과적이다. 이렇게 질의 실행 트리내 각 노드에서 수행되는 하나의 연산에 대해서 다수의 로우 묶음을 한번에 처리하는 SIMD 병렬화를 적용함으로써 성능을 개선하는 것이다.

제안하는 벡터처리기반 질의 실행 엔진의 다음과 같은 2가지 주요한 설계 요소를 포함하고 있다.

- *컬럼지향 데이터구조 로딩:* 질의 수행시 vector-at-a-time 모델 방식으로 수행하기 위해, 하둡의 HDFS내에 테이블을 컬럼지향 데이터 구조로 로딩하는 부분이다.
- *컬럼벡터 처리 및 최적화:* 질의 실행 계획 트리내 각 노드의 데이터 타입에 따라 표현식 평가하고 처리속도 향상하기 위한 최적화를 수행하는 부분이다.

그림 1의 벡터처리기반 질의 실행 엔진 (Vectorized Query Executor)은 성능을 최대화하기 위해, 이 두가지 설계요소를 고려하여 구현되어야 한다.

4 질의 실행 엔진 설계

질의 실행 엔진 설계를 위한 기본적인 아이이디어는 아파치 하이브의 벡터처리기반 질의 실행엔진에서 사용한 방법을 참조하였다 [2]. SQL-on-Hadoop에서 수행하는 대부분의 OLAP (Online Analytical Processing) 질의는 아래 질의 예처럼 SUM, COUNT, AVG, STD 등과 같은 집계연산으로 하나의 연산에 주어진 컬럼들에 대해 반복적으로 실행하는 경우가 빈번하다.

```
SELECT SUM(balance), COUNT(*), AVG(age)
FROM account WHERE age > 30;
```

먼저, 벡터처리를 위해 HDFS내 테이블을 컬럼지향 데이터구조로딩하는 부분은 벡터화된 로우 배치를 표현할 수 있다. 하지만, 텍스트 파일, 시퀀스 파일과 같은 컬럼기반 파일 포맷이 아니더라도 데이터를 로딩하는 시점에 벡터화된 열 배치로 로딩할 수 있는 리더

(vectorized reader)를 구현하면 된다. 다시 말해, 하위 파일 저장구조와 무관하게 데이터를 로딩하여 저장한는 어댑터 계층 (adapter layer)을 추가해서 구현할 수 있다. 이 중간 추상계층을 통해 벡터화된 로우 배치 (vectorized row batch)로 변환할 수 있다. 데이터 로딩 부분도 성능과 밀접한 관계가 있으므로 로우 배치 (row batch)를 효율적으로 읽어들이는 것이 중요하다. 따라서, 빅데이터 저장 계층으로부터 데이터를 처리시 컬럼벡터벡터 단위로 복수개 (1,024개) 로우 (row)를 포함하고 있는 로우 배치(row batch)를 얻어오는 처리기반 반복자(iterator) 인터페이스 정의가 필요하다. 그림 2는 기존 질의처리 대상이 되는 테이블에 대해서 벡터화 처리를 수행하기 위해 벡터화된 로우 배치를 저장하는 VectorizedRowBatch 클래스 다이어그램을 보여주고 있다.

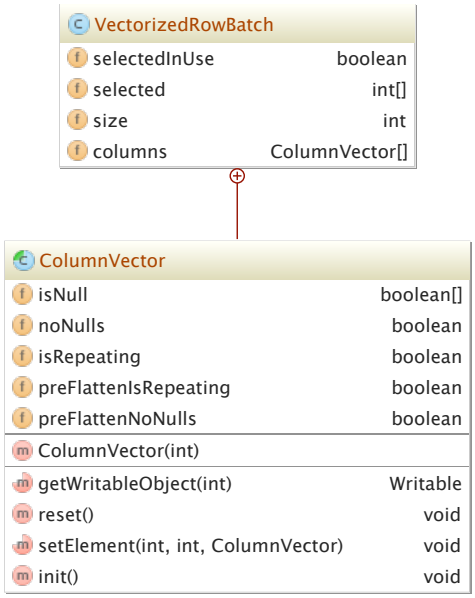

그림 2: VectorizedRowBatch 클래스 다이어그램

현재 사용하고 있는 SQL-on-Hadoop 시스템 [3]이 자바로 구현되어 있기 때문에, C/C++에서 사용하는 방식인 SSE, AVX 등을 사용하는 CPU 명령어 수준 병렬화 (instruction-level parallelism)을 적용하기가 어렵다. 따라서, 그림 2의 컬럼벡터 (ColumnVector) 객체의 배열을 로우 배치로 처리하는 데이터수준 병렬화 (data-level parallelism) 접근 방법을 사용한다. 여기서, 컬럼벡터 클래스는 처리할 데이터타입에 따라 LongColumnVector, DoubleColumnVector 등의 클래스로 상속받아 구현하면 된다.

효과적인 컬럼벡터 처리를 위해 논리(boolean)/필터 표현식에 대한 최적화처리, 연산수식에 대한 중간결과 저장처리 및 널(NULL) 값과 컬럼내 반복되는 값들에 대한 최적화 처리를 수행한다. 이를 위해, ColumnVector 추상 클래스내에서 널 값, 반복되는 값에 대한 처리하는 인터페이스를 정의한다. AND/OR와 같은 논리연산이 필요한 표현식의 경우, AND 연산은 연산식의 처음 연산결과가 거짓(false)이면 나머지 구성하고 있는 연산식 평가는 수행하지 않아도 거짓이므로 수행하지 않고 결과를 출력하는 *short circuit optimization*으로 처리할 수 있다 [2]. 그리고, 컬럼지향 데이터저장 (column-oriented storage) 방식의 장점을 이용하여 중복되는 값과 널값에 대한 처리, 데이터 압축, 벡터화된 사용자 함수 등을 최적화함으로써 성능 개선을 얻을 수 있다.

5 결론

본 논문은 하둡기반 데이터 웨어하우스에서 SQL 질의처리 성능 개선을 위한 벡터처리기반 질의실행 엔진 설계를 제안하였다. 제안한 시스템은 질의 대상인 테이블에 대한 컬럼벡터들을 데이터-레벨 병렬화하는 방식을 적용하였다.

제안한 벡터화된 질의 실행 엔진을 빅데이터 분석 서비스에 적용하면, 데이터 분석가와 데이터 과학자 같은 사용자는 OLAP의 주요 집계 연산을 보다 빠르게 분석할 수 있는 장점을 제공한다. 그리고, 빅데이터 초기 분석을 수행하기 위해 필요한 표준 SQL (ANSI-SQL) 에드혹 질의들도 벡터화된 질의실행 엔진을 통해 효과적으로 수행할 수 있다. 향후 연구주제로는 OLAP의 주요한 집계연산 결과로 근사결과와 오차정보를 함께 제공함으로써, 응답시간을 단축할 수 있는 근사 질의처리 기법에 관한 것이다.

감사의 글

본 연구는 한국전자통신연구원 연구운영비지원사업의 일환으로 수행되었음. [16ZS1410, 듀얼모드 배치·쿼리 분석을 제공하는 빅데이터 플랫폼 핵심 기술 개발]

참고 문헌

[1] A. Floratou, U. F. Minhas, and F. Özcan, "SQL-on-Hadoop: Full Circle Back to Shared-nothing Database Architectures," *Proc. VLDB Endow.*, vol. 7, pp. 1295–1306, Aug. 2014.

[2] Y. Huai, A. Chauhan, A. Gates, G. Hagleitner, E. N. Hanson, O. O'Malley, J. Pandey, Y. Yuan, R. Lee, and X. Zhang, "Major Technical Advancements in Apache Hive," in *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, SIGMOD '14, pp. 1235–1246, ACM, 2014.

[3] H. Choi, Y. D. Chung, J. Son, H. Yang, B. Lim, S. Kim, and H. Ryu, "Tajo: A Distributed Data Warehouse System on Large Clusters," in *Proceedings of the 2013 IEEE International Conference on Data Engineering (ICDE 2013)*, ICDE '13, pp. 1320–1323, 2013.

[4] 김성수, 정문영, 이태휘, and 원종호, "의료 데이터기반 빅데이터 분석 서비스 구현 ," in *한국정보과학회 학술발표논문집*, 제42회 한국정보과학회 동계학술발표회, pp. 157–159, 2015.

[5] P. A. Boncz, M. Zukowski, and N. Nes, "MonetDB/X100: Hyper-Pipelining Query Execution," in *CIDR*, vol. 5, pp. 225–237, 2005.

[6] M. Zukowski and P. A. Boncz, "Vectorwise: Beyond Column Stores.," *IEEE Data Eng. Bull.*, vol. 35, no. 1, pp. 21–27, 2012.