정렬 버퍼 크기에 따른 타조의 외부합병정렬 성능 연구

이종백○ 강운학 이상원
성균관대학교
hundredbag@skku.edu, woonagi319@skku.edu, swlee@skku.edu

External merge sorting in Tajo with variable Sort Buffer Size

Jongbaeg Lee○ Woon-hak Kang Sang-won Lee
Sungkyunkwan University

요 약

거대한 양의 데이터로부터 가치 있는 정보를 추출해 내는 빅데이터 기술의 필요성은 나날이 커지고 있다. 빅데이터 분석을 위해 사용되는 하둡 시스템은 맵리듀스 기술을 통해 데이터를 처리하였으나, 맵리듀스 프레임워크는 비유연성, 실시간 질의 처리의 한계 등의 단점을 가지고 있다. 이를 극복하기 위한 연구로 SQL-on-Hadoop이라 불리는 하둡 기반의 SQL 질의 처리 기술이 주목을 받고 있다. SQL-on-Hadoop 기술 중 하나인 타조는 국내 개발진이 주축이 되어 개발되었다. 타조는 데이터의 처리와 분석을 위해 외부병합정렬 알고리즘을 사용하며, 정렬 버퍼 사이즈 매개변수를 이용하여 정렬에 사용되는 청크의 크기를 조절한다. 본 논문은 타조의 정렬 연산에 영향을 미치는 정렬 버퍼 사이즈 매개변수를 변경함에 따라 발생하는 성능의 차이를 보인다. 또한, 정렬 버퍼 사이즈가 증가함에 따른 CPU 캐시 미스의 비율 증가가 성능 차이의 큰 원인임을 보인다.

1. 서 론

빅데이터란 단순히 거대한 양의 데이터만을 의미하는 것이 아니며, 많은 양의 데이터로부터 가치 있는 정보를 추출하고 결과를 분석하는 기술을 의미한다. 대표적인 소셜 네트워크 서비스인 '페이스북'에서는 하루 500테라바이트 이상의 데이터를 처리한다. 또한, 뉴욕 증권거래소, 유로넥스트와 같은 여러 증권 거래소를 운영하는 'NYSE 유로넥스트'에서도 하루 2테라바이트 이상의 데이터가 생성된다[1]. 발생하는 데이터의 양이 증가함에 따라 거대한 양의 데이터로부터 의미 있는 정보를 추출하기 위해 빅데이터 분석의 필요성은 크게 증가하였다. 그러나 기존의 관계형 데이터베이스 시스템을 이용한 데이터 처리는 처리 시간과 비용 측면에서 한계를 나타냈고, 이를 극복하기 위하여 분산 환경에서 데이터를 처리하는 빅데이터 프레임워크 하둡[2]이 등장하였다.

하둡은 분산 환경에서 빅데이터를 처리할 수 있는 프레임워크로, 높은 확장성과 신뢰성을 보인다. 하둡은 크게 데이터를 분산 저장하는 하둡 분산 파일시스템(Hadoop Distributed File System, HDFS)과 빅데이터를 분산 처리하는 맵리듀스(MapReduce)로 구성된다. 맵리듀스는 맵 함수와 리듀스 함수를 이용해 데이터를 처리하는 단순한 모델을 제공하며, 데이터 일괄 처리에 효율적인 구조를 지니고 있다.

데이터 일괄 처리에 효율적인 맵리듀스는 증가하는 실시간 질의 처리의 요구를 충족시키지 못하였다. 또한 맵리듀스는 중간 데이터의 처리에 있어서 많은 디스크 I/O와 네트워크 트래픽을 발생시키는 문제점을 가진다. 이러한 맵리듀스의 한계를 극복하고자 HDFS에 저장된 대용량의 데이터에 대해 SQL 질의를 처리하는 SQL-on-Hadoop 기술이 주목받고 있으며, 대표적인 예로 Apache Hive, Cloudera의 Impala, UC Berkeley의 Spark, 국내 개발진이 주축을 이뤄 개발된 타조(Tajo)가 있다.

타조에서 Order by, Join 등의 질의를 수행하기 위해 정렬 연산을 사용하며, 이 연산은 데이터 분석 작업에서 중요한 역할을 한다. 타조의 정렬 연산에 영향을 주는 정렬 버퍼 사이즈(Sort Buffer Size) 매개변수는 외부합병정렬의 청크(Chunk)의 크기를 결정할 때 영향을 주며, 해당 값을 변경하였을 때 정렬 질의의 수행 성능이 변경하는 것을 확인할 수 있다. 본 논문에서는 타조의 정렬 버퍼 크기 매개변수의 값의 변화에 따른 정렬 연산의 성능을 측정하여 비교하고, 발생하는 성능 차이의 원인에 관하여 분석한다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 본 연구의 배경이 되는 하둡 분산 파일시스템과 SQL-on-Hadoop 기술에 관하여 설명한다. 3장에서는 타조와 타조에서 사용하는 정렬 알고리즘에 관하여 설명하고, 4장에서는 정렬 버퍼 크기를 변경하며 측정한 타조의 성능 평가와 분석결과에 대하여 논할 것이다. 마지막으로 5장에서는 결론과 향후연구로 본 논문을 마무리한다.

2. 하둡 분산 파일시스템과 SQL-on-Hadoop

2.1. 하둡 분산 파일시스템

하둡(Hadoop)은 클러스터를 이용하여 커다란 데이터 집합에 대한 분산 처리를 제공하는 소프트웨어 프레임워크이다. 하둡은 여러 호스트로 데이터의 저장과 연산을 나누어 처리하며, 병렬 처리 시에 데이터를 연산이 동작하는 호스트로 옮기는 것이 아니라 연산을 데이터가 저장된 호스트로 이동하여 처리하는 특징을 가지고 있으며, 높은 확장성과 신뢰성을 제공한다. 하둡에서는 하둡

분산 파일 시스템(Hadoop Distributed Filesystem, HDFS)에 데이터를 저장함으로써 위의 특징들을 제공한다.

HDFS는 상용 하드웨어에서 실행할 수 있도록 디자인된 분산 파일시스템이다. HDFS는 네임노드와 데이터노드로 구성되는 마스터-슬레이브 구조를 가진다. 네임노드는 HDFS의 파일과 디렉토리의 메타데이터를 관리하며, 데이터노드에 저장된 실제 데이터 블록의 매핑 정보를 제공하는 역할을 한다. 데이터노드는 블록 단위(128MB default)로 나누어지는 실제 데이터를 저장하는 역할을 한다. 데이터노드는 내고장성을 위하여 같은 데이터 블록을 여러 데이터노드에 복사하여 저장한다.

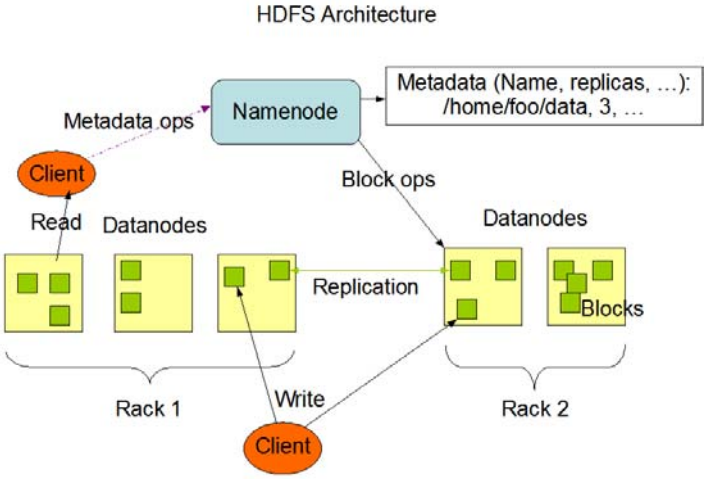

그림 1. HDFS 구조[2]

2.2. SQL-on-Hadoop

하둡에서 대용량 데이터에 대한 분산 처리를 위해 사용되는 맵리듀스 기법은 데이터의 일괄 처리에 효율적으로 동작한다. 그렇지만 맵리듀스는 중간 데이터 처리에 많은 디스크 I/O와 네트워크 트래픽을 발생시키며, 고차원 언어의 부재로 인한 코드 재사용성의 한계, 복잡한 알고리즘 구현의 한계를 보인다. 위의 문제점과 더불어 실시간 질의 처리에 대한 수요가 증가함에 따라 데이터 처리를 위한 맵리듀스의 대안으로 SQL-on-Hadoop 기술이 주목을 받고 있다.

SQL-on-Hadoop 이란 HDFS에 저장된 데이터에 대하여 SQL 스타일의 질의처리를 제공하는 시스템으로, Apache Hive, 타조(Tajo), Cloudera의 Impala, UC Berkeley의 Sqark 등 다양한 SQL-on-Hadoop 시스템이 경쟁을 하고 있다. SQL-on-Hadoop은 공통적으로 데이터 파일을 HDFS에 저장하는 특징을 가진다. 또한 SQL문을 통하여 데이터를 분산 환경에서 실행하는 특징을 지닌다. 각각의 기술은 처리하고자 하는 질의 수행의 시간, 스케줄링 기법 등 다양한 특성을 지니고 있기 때문에 시스템의 특성을 이해하고, 워크로드에 알맞은 시스템을 선택하는 것이 중요하다.

3. 아파치 타조

SQL-on-Hadoop 기술 중 하나인 타조는 하둡 기반의 대용량 분산 데이터 웨어하우스 시스템이다[3]. 타조는 HDFS에 저장된 데이터에 대하여 짧은 지연시간을 요구하는 질의 처리를 위해 설계되었으며, 실시간 질의 처리뿐 아니라 긴 시간동안 수행되는 대용량 데이터에 대한 ETL 작업도 지원한다. 또한 SQL 표준을 지원하며, 성능을 위하여 질의 전체를 분산 처리하는 특징을 지닌다.

타조는 타조 마스터와 타조 워커로 이루어지는 마스터-슬레이브 구조를 이용하여 데이터를 처리한다. 타조 마스터는 타조 클러스터의 마스터 역할을 담당하며, 질의 수행 계획과 각종 통계 정보를 관리하고, 클러스터 전체의 자원을 관리하는 역할을 한다. 타조 워커는 마스터가 요청한 질의를 실행하는 역할을 하며, 또한 여러 워커 중 하나를 선택하여 쿼리마스터로써 동작하도록 한다. 쿼리마스터는 마스터-슬레이브 구조적인 특성으로 인해 마스터 서버에 문제가 생길 경우 질의 처리가 정상적으로 수행되지 못하는 경우를 극복하기 위한 마스터 서버로, 질의 별로 질의 실행에 관한 분산 실행 계획을 제어하는 역할을 한다.

Order by, Join 등의 질의를 수행 시 타조에서는 데이터의 정렬을 위한 알고리즘으로 외부합병정렬을 사용한다. 타조 워커는 정렬 버퍼 크기로 결정된 청크의 크기 만큼의 데이터를 메모리 내부에서 정렬하며, 해당 정렬의 임시 결과를 디스크에 저장함으로써 외부합병정렬 연산을 수행한다. 외부합병정렬에 영향을 주는 매개변수인 정렬 버퍼 크기는 tajo-site.xml 파일을 통해 변경할 수 있으며, 이 값을 이용하여 외부합병정렬의 청크의 크기를 변경할 수 있다.

4. 성능 평가 및 결과

4.1. 실험 환경

정렬 버퍼 크기에 따른 타조 워커의 정렬 연산의 성능 평가 실험 환경은 <표 1>과 같다. 정렬 연산의 대상으로 TPC-H 벤치마크의 여러 테이블 중 가장 데이터의 양이 많은 'LINEITEM' 테이블을 이용하였으며, 해당 테이블의 'l_suppkey' 행을 정렬에 사용되는 키로 사용하였다. LINEITEM 테이블은 HDFS에 저장된 것을 이용하였고, 크기는 23.6GB로 설정하였다. 타조 워커에서 동시에 동작하는 태스크의 수는 8개이고, 정렬 버퍼 크기는 4MB, 64MB, 그리고 기본 값인 200MB 세 가지로 변경하며 성능을 측정하였다.

항목	설명
CPU	Intel(R) Core(TM) i5-4670 CPU @ 3.40GHz
RAM	8GB
Temp 영역	Samsung SSD 840 Pro 256GB
전체 데이터 크기	23.6 GB
태스크 수	8
정렬 버퍼 크기	4MB, 64MB, 200MB(default)

표 1. 성능평가 환경

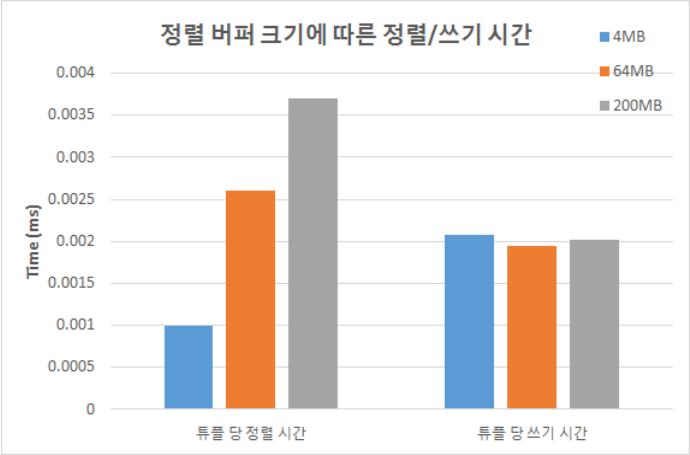

그림 2 정렬 버퍼 크기에 따른 정렬/쓰기 시간

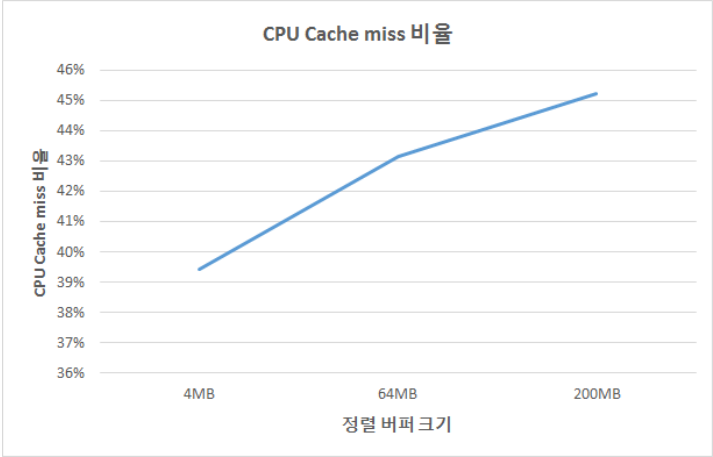

그림 3 정렬 버퍼 크기에 따른 CPU 캐시 미스 비율

4.2. 실험 결과 및 분석

정렬 버퍼 크기	총 수행 시간(초)
4MB	369
64MB	444
200MB(default)	537

표 2. 정렬 버퍼 크기에 따른 수행 시간

<표 2>는 정렬 버퍼 크기를 변경하며 수행한 정렬 연산의 총 수행 시간을 나타낸다. 정렬 버퍼 크기가 기본 값인 200MB 일 때의 정렬 성능은 537초로 64MB일 때 444초, 4MB일 때 369초 보다 느린 성능을 보이는 것을 확인했다.

<그림 2>는 정렬 버퍼 크기에 따른 성능 차이의 원인을 분석을 위한 튜플 당 정렬 시간, 튜플 당 쓰기 시간을 측정한 결과를 나타낸다. 정렬 버퍼 크기가 증가함에 따라 튜플 당 쓰기 시간은 비슷한 값을 보이는 반면, 튜플 당 정렬 시간은 크게 증가하는 것을 확인할 수 있다.

튜플 당 정렬 시간과 정렬 버퍼 크기의 관계를 분석하기 위해 리눅스의 시스템 성능 측정 도구인 Perf[4]를 이용하여 CPU 캐시 미스 비율을 측정하였다. <그림 3>은 정렬 버퍼 크기에 따른 CPU 캐시 미스 비율을 나타내며, 정렬 버퍼 크기가 4MB, 64MB, 200MB 일 때 CPU 캐시 미스 비율은 각각 39.4%, 43.1%, 45.3%로 측정되었다.

측정 결과를 통해 정렬 버퍼 크기가 증가함에 따라 CPU 캐시 미스 비율이 증가하며, 이것이 정렬 질의의 수행 시간에 영향을 준 것을 확인할 수 있다. 하나의 타조 워커에서 정렬을 수행하는 여러 태스크가 동작 할 때, 정렬 버퍼의 크기가 증가함에 따라 하나의 청크에 대한 인-메모리 정렬에 필요한 시간이 증가하게 된다. 인-메모리 정렬의 시간이 증가함에 따라 정렬 대상이 되는 데이터가 CPU 캐시로부터 쫓겨날 가능성이 증가한 것을 CPU 캐시 미스 비율이 증가한 원인으로 볼 수 있다.

5. 결론 및 향후 연구

본 논문에서는 SQL-on-Hadoop 기술 중 하나인 타조의 외부합병정렬 연산에 사용되는 정렬 버퍼 크기 매개변수를 변경함에 따른 정렬 성능을 평가하였고, 정렬 버퍼 크기에 따른 성능 차이가 발생하는 원인에 관하여 분석하였다.

성능 평가를 통해 정렬 버퍼 크기를 200MB에서 4MB로 줄일 경우 정렬 연산의 총 수행 시간은 537초에서 369초로 감소하는 것을 확인할 수 있었다. 정렬 버퍼 크기가 외부병합정렬 연산에 영향을 주는 원인을 분석한 결과 튜플 당 정렬 시간이 정렬 버퍼 크기가 클수록 증가하는 것을 확인하였으며, 정렬 버퍼 크기가 증가함에 따라 CPU 캐시 미스 비율의 증가하는 것이 이러한 현상의 원인임을 확인하였다.

향후 연구에서는 외부합병정렬 연산에 영향을 주는 또 다른 매개변수인 Fanout 값에 따른 성능을 측정할 것이며, 해시 연산에 관해서도 같은 방향의 연구를 진행할 것이다.

사사

이 논문은 2015년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (R0126-15-1108, FlashSQL:비휘발성 메모리 기반 개방형 고성능 DBMS 개발)

이 논문은 2015년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (10041244, 스마트TV 2.0 소프트웨어 플랫폼)

본 연구는 미래창조과학부 및 정보통신기술진흥센터의 SW중심대학-ICT/SW창의연구과정 지원사업의 연구결과로 수행되었음" (IITP-2015-R2215-15-1005)

참고문헌

[1] Cisco, "Data Virtualization Redefines the Stock Exchange", Cisco, 2013

[2] Apache Hadoop, http://hadoop.apache.org/, 2009

[3] Apache Tajo: A big data warehouse system on Hadoop, http://tajo.apache.org/, 2013

[4] Arnaldo Carvalho de Melo, "The New Linux perf tools", presentation from Linux Kongress, 2010

[5] Tom White, "Hadoop: The Definitive Guide", O'REILLY, 2012