

분산 병렬 환경에서 효율적인 이종 데이터 처리를 위한 SQL 데이터 전송 기법

양현식, 백나은, 성미래, 장재우*
 전북대학교 IT정보공학과
 e-mail:gustlr1222@jbnu.ac.kr

SQL Data Transport Technique for Efficient Hybrid Data Processing on Distributed and Parallel Environment

HyeonSik Yang, Naeun Baek, Mirae Sung, Jae-woo Chang*
 Dept. of Information Technology, Chonbuk Univ.

요 약

인터넷 발전이 가속화되고 SNS가 보급된 이후 과거와는 비교할 수 없을 정도로 큰 데이터 트래픽이 발생하고 있다. 기존의 DBMS는 이를 효과적으로 처리할 수 없었기 때문에 Hadoop과 같은 NoSQL이 탄생하였고, 최근 NoSQL 및 기존 SQL DBMS의 협업을 통해 유연하고 강력한 데이터 관리를 수행하는 연구가 진행되었다. 효율적인 질의 처리를 위한 대표적인 연구로 SQL 기반 분산 병렬 질의 처리 기법과 Hive등이 존재한다. 그러나 기존의 기법은 분산 병렬 환경을 고려하지 않아 SQL DBMS의 질의 결과를 효율적으로 Hive에 전송하지 못한다. 본 논문에서는 SQL DBMS에서 Hive로의 효율적인 SQL 데이터 이동을 위해 네트워크 비용을 최소화하는 기법을 제안하고, 제안하는 기법의 우수성을 제시한다.

1. 서론

인터넷 발전이 가속화되고 SNS가 보급된 이후 종래와 다른 형식의 데이터 트래픽이 발생하고 있으며 이를 빅데이터라 칭한다. 빅데이터는 크기가 매우 크고, 데이터의 종류가 매우 다양하며, 데이터 증가 속도가 매우 빠르다는 특징을 가져 기존의 DBMS로는 효과적으로 저장, 분석, 수행할 수 없었다. 이러한 문제를 해결하기 위해 MapReduce[1] 플랫폼이 탄생하였고 Hadoop[2]은 오픈소스 MapReduce 구현으로서 NoSQL 부문에서 선두를 달리고 있는 분산 병렬처리 프레임워크이다.

한편 SQL과 NoSQL의 장점을 취득하여 유연하고 강력한 질의 처리를 위해 분산 병렬 환경에서 SQL DB와 Hadoop 기반 질의 처리기인 Hive를 결합하는 연구가 진행되었다. 이를 위해 분산 병렬 환경에서 SQL 질의 결과를 Hive의 저장소인 HDFS[3]에 저장해야 한다. 그러나 기존 SQL 데이터를 HDFS에 저장하는 기법은 분산 병렬 환경을 고려하고 있지 않아 불필요한 데이터 전송을 지닌다는 문제점을 가지고 있다.

따라서 본 논문에서는 분산 병렬 환경에서 효율적인 SQL 데이터 이동을 위해, 지역성을 고려해 SQL 데이터를 통합하지 않는 방식으로 네트워크 비용을 최소화한다.

본 논문의 구성은 다음과 같다. 2장에서는 분산 병렬 SQL 질의 처리 기법과 HDFS에 대해서 설명한다. 3장에서는 기존 기법의 문제점을 지적하고, 제안하는 기법을 상세히 서술한다. 4장에서는 기존 기법과 제안하는 기법의 비교 성능 평가를 통해 제안하는 기법의 우수성을 증명하며, 5장에서는 결론과 향후 연구를 제시한다.

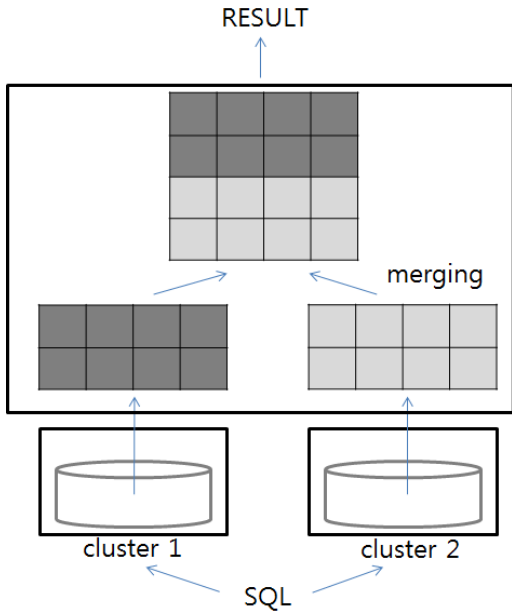
2. 관련 연구

2.1 분산 병렬 SQL 질의 처리 기법

SQL DBMS에서 질의 처리 능력을 극대화하기 위한 대표적인 기법은 테이블 수평 분할 기법과 테이블 수직 분할 기법이 존재한다[4]. 이 기법들은 테이블을 여러 클러스터에 분산 저장한 후 병렬적으로 동시 질의하고, 각 클러스터의 질의 결과를 결합하여, 하나의 클러스터에서 수행한 질의 처리와 동일한 결과를 더 빠른 처리 속도로 수행하는 기법이다. 본 논문에서는 두 기법 중 더 널리 쓰이는 테이블 수평 분할 기법을 분산 병렬 SQL DBMS를 가정한다.

수평 분할 기법 상에서의 select 질의 처리는 (그림 2)와 같이 이루어진다. 분산 클러스터에 동일한 질의를 복사 전송하여, 각 클러스터가 병렬적으로 질의를 수행하고 이를 취합해 최종 결과를 완성한다.

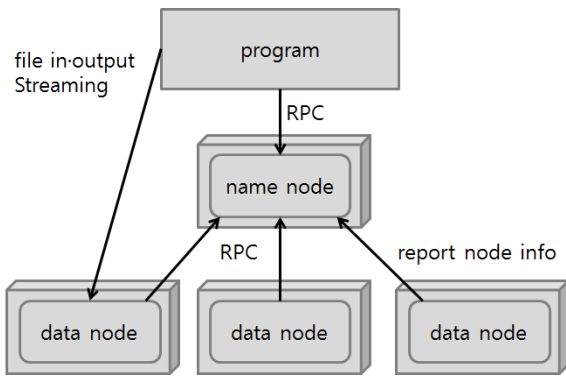
* Corresponding author



(그림 1) 테이블 수평분할 기법에서 select 처리

2.2 HDFS

HDFS(Hadoop Distributed File System)는 Hadoop과 Hive에서 분산 데이터 처리를 위해서 데이터를 저장하기 위한 파일 시스템이다. HDFS는 다중 노드에 파일을 블록 단위로 분산해 저장하며, 이를 위해 (그림2)처럼 다수의 데이터 노드와 하나의 네임 노드의 구조를 사용한다. 이 다수의 데이터 노드에 하나의 블록을 복제해 저장하기 때문에 특정 노드의 장애시에도 전체 시스템을 정지하지 않고 장애를 복구(fault tolerance)할 수 있는 장점을 지닌다.

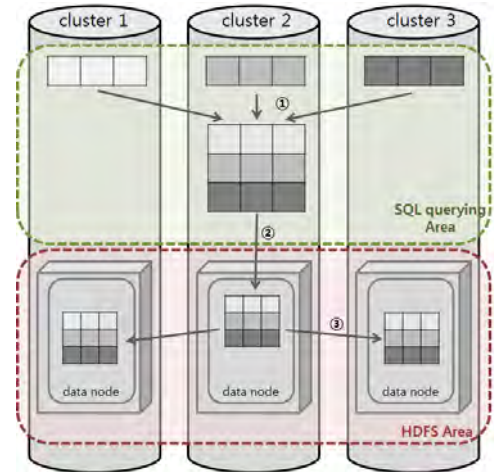


(그림 2) HDFS 구조

3. 동기

본 장에서는 기존 기법이 불필요한 네트워크 통신을 필요로 한다는 문제점을 지적한다. 즉, 하나의 Client에서 분산 병렬 SQL 질의 결과를 획득할 때, 모든 클러스터의 결과를 받기 위해 필수적으로 한 클러스터에서 SQL 질의

결과를 통합을 수행해야 한다. 그러나 SQL과 NoSQL 모두의 데이터를 참조한 질의를 수행하기 위해서 분산 병렬 SQL 질의 결과를 HDFS로 전송할 때는, HDFS의 분산 정책상 SQL 질의 결과의 통합은 불필요해진다. (그림3)은 기존 기법을 통해 SQL 데이터를 HDFS에 저장하는 방법을 나타낸다. ① 각 클러스터의 데이터를 취합하고, ② 취합된 데이터를 HDFS에 전송하며, ③ HDFS가 이를 다시 분산 저장하는 모습이다. 이때 각 Cluster 1과 3은 자기가 본래 가지고 있던 데이터를 Cluster 2에 전송한 후 다시 전송 받는 불필요한 네트워크 통신을 수행하고 있다.



(그림 3) 기존 기법에서의 SQL 데이터 이동

4. 제안하는 기법

제안하는 기법은 분산 병렬 SQL 질의 처리기에서, 요청된 질의가 바로 Client에게 전달될 것인지 HDFS로 전송될 것인지 확인하여, HDFS로 전송된다면 기존 분산 병렬 SQL 질의 처리 최종 단계인 취합 단계를 제외하고 각 클러스터가 local에 위치한 HDFS에 SQL 질의 결과를 전송하여 해당 문제를 해결한다. <표1>은 제안하는 기법의 의사 코드이다.

```

schedule( JobInfo job, ClusterInfo cluster ){
  if( job.type == JOB_TYPE_TO_CLIENT ){
    define Integer MaxValue = cluster.at(0).value
    define Integer MaxIndex = 0
    for i=1 to cluster.clusternum {
      cluster.at(i).select( )
      if( MaxValue < cluster.at(i).value ){
        MaxIndex = i
        MaxValue = cluster.at(i).value
      }
    }
  }
}
    
```

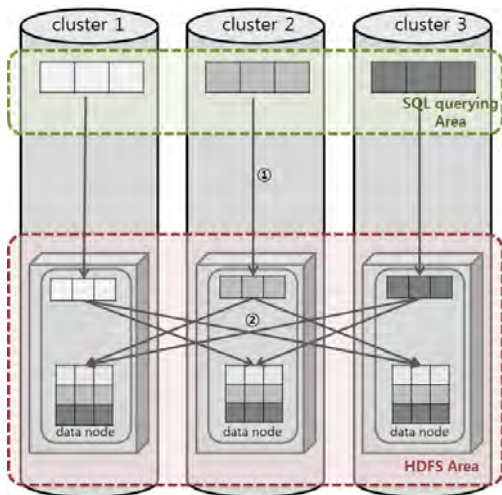
```

cluster.at(MaxIndex).setMerger( )
for i=1 to cluster.clusternum {
  cluster.at(i).result.goto(MaxIndex)
}
}
else if( job_type == JOB_TYPE_TO_HDFS){
for i=1 to cluster.clusternum {
  cluster.at(i).select( )
  cluster.at(i).result.toHDFS( )
}
}
}
}
    
```

<표 1> 제안하는 기법의 의사 코드

schedule함수는 먼저 수행이 요청된 질의의 종류를 파악한다. 요청된 질의가 관계형 데이터만을 이용한 질의일 경우, 해당 질의 수행에 있어서는 관계형 데이터가 HDFS로 이동할 필요가 없으며 이때에는 관계형 데이터를 가장 많이 가지고 있는 클러스터를 찾아 이 클러스터에서 취합을 진행한다. 만약 요청된 질의가 이중 데이터 처리가 필요할 경우 취합 없이 동일 클러스터의 HDFS로 이동하게 된다.

(그림4)는 select 질의에서 제안하는 기법을 통해 SQL 데이터를 HDFS에 저장한다. ① 각 클러스터의 데이터를 local에 위치한 HDFS에 전송하고, ② HDFS가 분산 저장을 시행하여 데이터 이동이 완료된다.



(그림 4) 제안하는 기법에서 SQL 데이터 전송

기존 기법에서 Cluster 1과 Cluster 3이 소유하고 있던 데이터를 다시 전송받았던 데 반해, 제안하는 기법은 각 클러스터가 가지고 있던 데이터를 다시 전송 받지 않는다.

이를 통해 제안하는 기법으로 분산 병렬 환경에서 SQL 데이터 이동시 불필요한 네트워크 통신 비용을 줄일 수 있어 효율적인 SQL 데이터 전송이 가능하다.

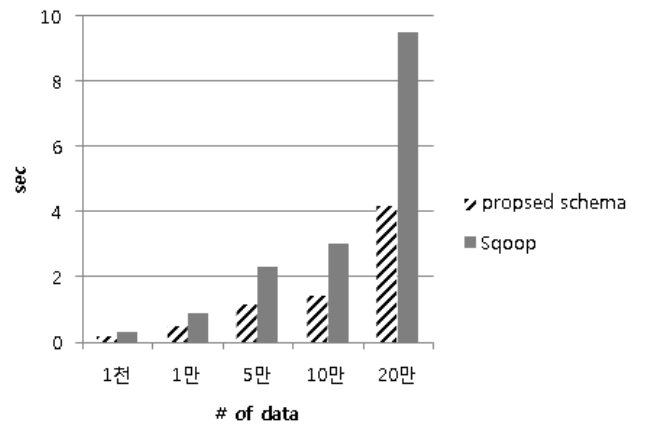
4. 성능평가

본 절에서는 제안하는 기법의 우수성을 보이기 위해 제안하는 기법을 기반으로 구현한 미들웨어 및 SQL 데이터를 HDFS로 전송하는 기존 프로그램인 Sqoop[5]을 비교한다. 성능평가의 실험 환경은 <표1>과 같으며, 동일한 환경의 클러스터 4대를 사용하였다.

항목	성능
CPU	2.9GHz Quad-Core Intel Core i5
Memory	4GB
OS	ubuntu 12.4

<표 1> 실험 환경

(그림 5)는 병렬적으로 MySQL 기반 select 질의를 수행한 후, 각각 Sqoop 및 제안하는 기법을 적용한 미들웨어의 성능을 비교한 것이다. 질의문은 위스콘신 벤치마크를 이용하였다. 실험 결과 평균 102%의 성능 향상이 있었으며, 질의하는 데이터 집합이 증가할수록 제안하는 기법을 적용한 미들웨어가 Sqoop보다 더 좋은 성능을 보여주었다.



(그림 5) 제안하는 연구

5. 결론 및 향후 연구

본 논문에서는 분산 병렬 환경에서 이중 데이터 처리를 위한 효율적인 SQL 데이터 이동 기법을 제안하였다. 향후 연구로는 분산 병렬 환경에서 이중 데이터 처리를 위해 join 질의에서 데이터 집적도를 고려한 효율적인 SQL 데이터 이동 기법을 연구하는 것이다.

참고문헌

- [1]Dean, Jeffrey, and Sanjay Ghemawat. "MapReduce: simplified data processing on large clusters." Communications of the ACM 51.1 (2008): 107-113.
- [2]Thusoo, Ashish, et al. "Hive: a warehousing solution over a map-reduce framework." Proceedings of the VLDB Endowment 2.2 (2009): 1626-1629.
- [3]Borthakur, Dhruba. "HDFS architecture guide." HADOOP APACHE PROJECT http://hadoop.apache.org/common/docs/current/hdfs_design.pdf (2008).
- [4]Agrawal, Sanjay, Vivek Narasayya, and Beverly Yang. "Integrating vertical and horizontal partitioning into automated physical database design." Proceedings of the 2004 ACM SIGMOD international conference on Management of data. ACM, 2004.
- [5]Ting, Kathleen, and Jarek Jarcec Cecho. Apache Sqoop Cookbook. " O'Reilly Media, Inc.", 2013.