

분산 CUBRID 상에서 세타 조인 및 중첩 SQL 병렬 질의처리를 지원하는 질의 재구성 기법

Query Reorganization Scheme supporting Parallel Query Processing of Theta Join and Nested SQL on Distributed CUBRID

양 현 식, 김 형 진, 장 재 우*
전북대학교 IT정보공학과

Hyeon-Sik Yang, Hyeong-Jin Kim, Jae-Woo Chang*
Dept. of Information Technology and Engineering,
Chonbuk University

요약

최근 SNS의 발전으로 인해 데이터의 양이 급격히 증가하였으며, 이에 따라 빅데이터 처리를 위한 분산 DBMS 기반 질의 처리 연구가 활발히 진행되고 있다. 이를 위해 CUBRID는 CUBRID Shard 서비스를 통해 데이터베이스를 shard 단위로 수평 분할하여 각기 다른 물리 노드에 데이터를 분산 저장하도록 지원한다. 그러나 CUBRID Shard는 shard간 데이터가 독립적으로 관리되기 때문에 세타 조인 및 중첩 질의와 같이 다수 서버에서의 테이블 참조가 필요한 질의는 처리가 불가능하다. 따라서 본 논문에서는 분산 CUBRID 상에서 세타 조인 및 중첩 SQL을 지원하는 질의 재구성 기법을 제안한다.

I. 서론

최근 SNS(Social Network Service)의 발전으로 인해 데이터의 양이 급격히 증가하였으며, 이에 따라 빅데이터에 대한 연구가 활발히 이루어지고 있다[1~3].

빅데이터 처리를 위한 DBMS로는 CUBRID[4]가 있으며 CUBRID는 다음과 같은 장점을 가진다. 첫째, 기존 데이터베이스 시스템들은 빅데이터 처리를 위한 ACID 성질을 만족하지 못하는 반면, CUBRID는 이를 만족함으로써 완벽한 트랜잭션을 지원한다. 둘째, 다양한 인덱스 기능을 지원함으로써, 질의 타입에 따른 최적화된 성능을 보장한다. 마지막으로 하드웨어, 소프트웨어, 네트워크 등에 장애가 발생해도 지속적인 서비스가 가능하게 하는 HA(High Availability) 기능을 제공한다. CUBRID Shard[5]는 CUBRID의 빅데이터 처리 성능 향상을 위해 제안된 것으로 CUBRID 데이터베이스를 shard 단위로 수평 분할하여 각기 다른 물리 노드에 분산 저장함으로써, 데이터의 분산 저장 및 병렬 처리를 지원한다. 그러나 CUBRID Shard는 shard간 데이터가 독립적으로 관리되기 때문에, 분산된 데이터에 대한 세타 조인과 중첩 질의가 불가능하다.

이러한 문제점을 해결하기 위해, 본 논문에서는 분산 CUBRID 상에서 다수의 서버에 대한 세타 조인 및 중첩 SQL의 병렬 처리를 지원하는 질의 재구성 기법을 제안한다. 분산 CUBRID는 수평 분할된 다수의 CUBRID 서버를 사용하여 데이터를 분산 저장하며, 제안하는 기법은

세타 조인 및 중첩 질의 질의문을 CUBRID 상에서 질의 가능한 SQL로 재구성함으로써, 사용자들이 CUBRID 환경에서 분산 저장된 데이터에 대한 세타 조인 및 중첩 질의 처리를 수행할 수 있도록 한다.

본 논문의 구성은 다음과 같다. 2장과 3장에서는 제안하는 기법의 시스템과 질의 수행 과정에 대해서 상세히 설명하고, 4장에서는 결론 및 향후 연구를 제시한다.

II. 제안하는 분산 CUBRID 시스템 구조

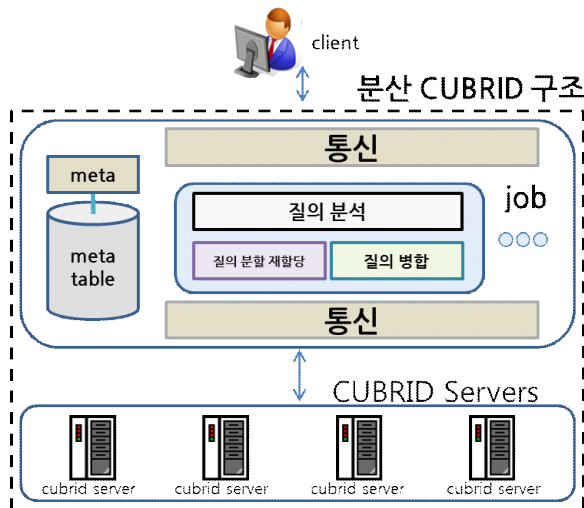
(그림 1)은 본 논문에서 제안하는 분산 CUBRID 시스템의 전체 구조를 나타낸다. 시스템은 크게 (i)통신 모듈, (ii)meta 테이블 및 meta 검색 모듈, (iii)job 컴포넌트로 구성된다. 첫째, 통신 모듈은 사용자와 본 시스템은, 혹은 시스템과 CUBRID 서버와의 데이터 송수신을 담당한다. 해당 모듈에서 송수신하는 데이터는 SQL 및 질의 결과 등을 포함한다. 둘째, meta 테이블은 데이터가 저장되어 있는 서버 정보를 저장하며, meta 검색 모듈은 이를 토대로 사용자의 질의를 수행하기 위해 어느 CUBRID 서버에 질의를 전송해야 하는지를 결정한다. 셋째, job 컴포넌트는 제안하는 질의 재구성 기법의 핵심 요소로 질의 타입에 따라 동적으로 생성되며, 질의 분석 모듈, 질의 분할 재할당 모듈, 질의 결과 병합 모듈로 구성된다. 먼저, 질의 분석 모듈은 job컴포넌트(이하 job이라 칭함)가 할당 받은 질의가 단순 select인지, 세타 조인 또는 중첩 질의인지 확인한다. 넷째, 질의 분할 재할당 모듈은 질의가 세타 조인 또는 중첩 질의일 경우, 질의를 CUBRID 서버에서 처리 가능한 다수의 질의로 변환하여

* Corresponding author

이를 새로운 job에 할당시킨다. 마지막으로, 질의 병합 모듈은 CUBRID 서버 또는 하위 job에서 반환한 질의 결과를 병합하여 최종 결과를 도출한다.

III. 세타 조인 및 중첩 SQL 처리를 지원하는 질의 재구성 기법

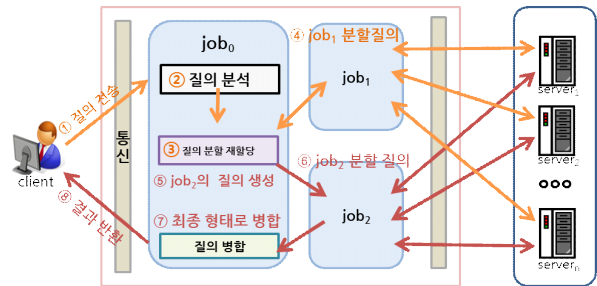
본 절에서는 제안하는 질의 재구성 기법에 대한 상세 설계 사항을 기술한다. 질의 재구성 기법은 전체 시스템 중 job 컴포넌트에서 작동하며 질의 타입이 단순 select 인지, 다수의 CUBRID 서버를 참조하는 세타 조인 및 중첩 질의인지 판단하여 적절하게 질의 분할 재할당과 질의 병합을 수행한다. 각 타입에 따른 수행과정은 다음과 같다.



▶▶ 그림 1. 제안하는 기법의 전체 시스템 구조

첫째, 질의 타입이 단순 select일 경우, meta 모듈을 통해 질의를 처리할 수 있는 모든 CUBRID 서버에 질의를 전송하고, 반환된 질의 결과를 병합한다. 둘째, 질의 타입이 세타 조인일 경우, 참조하는 테이블들을 단순 select로 분할하여, 새로운 질의문들을 구성한다. 예를 들어 <그림 2>와 같이 job₀이 “select * from employee as A, instructor as B where A.salary > B.salary”라는 질의를 받았다고 가정한다면 해당 질의는 2개의 테이블을 참조하기 때문에 job₀은 job₁과 job₂를 생성한 후, 먼저 job₁에 “select B.salary from instructor as B order by salary” 질의를 할당 시키고 결과 반환을 대기한다. job₁이 질의 결과를 반환하면 이를 기반으로 employee 테이블에 질의하기 위해 job₂에 “select * from employess as A where A.salary > θ ” 질의를 할당한다. 여기서 θ 는 job₁이 반환한 salary에 해당한다. 이때 job₁이 모든 결과를 반환할 때까지 대기하지 않고 결과가 도착하는 즉시 job₂에 질의를 할당함으로써 질의의 병렬 수행 지원한다. job₂가 결과를 반환하면 이를 job₁이 반환한 결과와 병합하여 최종 결과로 구성한 후 client에게 반환한다. 셋째, 질의 타입이 중첩 질의일 경우, 세타 조인 타입과 마찬가지로

참조하는 테이블을 단순 select하는 분할 질의문을 생성하여 하위 job에 할당 시킨 후 하위 job들이 각각 병합한 결과를 반환하기를 대기한다. 예를 들어 <그림 2>에서의 세타 조인 질의 과정과 마찬가지로 2개의 테이블을 참조하는 질의인 “select * from employee where salary > some(select salary from student where name=Yang)”를 job₀이 받았다고 가정하면, job₀은 job₁과 job₂를 생성 후, 먼저 job₁에 “select salary from student where name=Yang order by salary” 질의를 할당시키고 결과 반환을 대기한다. 결과가 반환되면 이를 토대로 job₂에 “select * from employee where salary > θ ” 질의를 할당한다. 여기서 θ 는 job₁이 반환한 salary에 해당하며, job₁과 job₂는 세타 조인 질의 처리와 마찬가지로 병렬 수행한다. job₂에서 반환되는 결과가 최종 결과 형태와 일치하기 때문에 job₀은 job₂의 결과를 추가 병합 없이 client에게 반환한다.



▶▶ 그림 2. 세타 조인 및 중첩 질의 수행 과정

IV. 결론 및 향후 연구

본 논문에서는 CUBRID 상에서 분산 데이터의 병렬 처리를 지원하기 위한 세타 조인 및 중첩 질의 재구성 기법을 제안하였다. 향후 연구로는 제안하는 기법을 대용량 데이터를 통해 성능 평가를 수행하는 것이다.

■ 참고 문헌 ■

- [1] DEAN, Jeffrey; GHEMAYAT, Sanjay. "MapReduce: simplified data processing on large clusters." Communications of the ACM, 2008, 51,1: 107-113.
- [2] Yang, Hung-chih, et al. "Map-reduce-merge: simplified relational data processing on large clusters." Proceedings of the 2007 ACM SIGMOD international conference on Management of data, ACM, 2007.
- [3] Rabl, Tilmann, et al. "Solving big data challenges for enterprise application performance management." Proceedings of the VLDB Endowment 5.12 (2012): 1724-1735.
- [4] CUBRID - <http://www.cubrid.com/>
- [5] CUBRID Shard - <http://www.cubrid.org/manual/91/en/shard.html>