

# 분산 병렬 질의 처리를 지원하는 CUBRID 기반 미들웨어 개발

김형일\*, 윤민\*, 조아라\*, 최문철\*, 장재우\*\*†

\*전북대학교 컴퓨터공학과, \*\*전북대학교 IT정보공학과  
e-mail:jwchang@jbnu.ac.kr

## Development of CUBRID based Middleware supporting Distributed Parallel Query Processing

Hyeong-Il Kim\*, Min Yoon\*, Ahra Cho\*, Mun-Chul Choi\*, Jae-Woo Chang\*\*†

\*Dept of Computer Engineering, Chonbuk University

\*\*Dept of Information Technology and Engineering, Chonbuk University

### 요 약

최근 SNS의 발전으로 인해 정보의 양이 급격히 증가하였으며, 이에 따라 빅데이터 처리를 위한 NoSQL에 대한 연구가 활발히 진행되고 있다. 그러나 NoSQL은 데이터베이스의 ACID 조건을 만족하지 못하는 문제점이 존재한다. 따라서 RDBMS를 기반으로 빅데이터 처리를 수행하는 연구가 활발히 진행되고 있다. 이를 위한 대표적인 기법인 CUBRID Shard는 데이터베이스를 Shard 단위로 수평 분할하여 각기 다른 물리 노드에 데이터를 분산 저장한다. 그러나 해당 기법은 한 클라이언트의 질의가 다수의 서버에서 실행되어야 하는 경우를 예는 질의를 처리하지 못하는 단점을 보인다. 따라서 본 논문에서는 병렬 질의 처리를 지원하는 CUBRID 기반 분산 미들웨어를 제안한다.

### 1. 서론

최근 SNS(Social Network Service)의 발전으로 인해 정보의 양이 급격히 증가하였으며, 이에 따라 빅데이터에 대한 연구가 활발히 이루어지고 있다[1~3]. 빅데이터는 기존의 IT 기술로 저장/처리/분석이 어려울 정도로 데이터의 생성 속도가 빠른 데이터를 의미한다. 빅데이터는 분량의 방대함으로 인해 그 자체가 정보로 활용되기 어려우며, 빅데이터에서 의미있는 정보를 추출하기 위한 분석 과정이 필요하다. 빅데이터 분석을 수행하기 위해서는, 대규모의 컴퓨팅 자원 및 대용량 데이터에 대한 효율적인 관리를 지원하는 빅데이터 관리 시스템이 요구된다. 이를 위해, Hadoop[4], MongoDB[5], Cassandra[6]를 대표로 한 NoSQL에 대한 연구가 활발히 진행되고 있다[4~12]. 그러나 NoSQL은 데이터베이스의 ACID 조건을 만족하지 못하는 단점을 보인다. 특히, NoSQL에서 확장성(Partition Tolerance) 및 유효성(Availability)에 초점을 맞추게 될 경우, 데이터의 일관성(Consistency)이 위배된다는 점은 NoSQL의 큰 단점으로 지적되고 있다. 이와 같은 NoSQL의 문제점으로 인해 최근 RDBMS를 기반으로 빅데이터 처리를 하고자하는 움직임이 활발해지고 있다.

RDBMS 기반의 대표적인 빅데이터 처리 기법으로는 CUBRID Shard[13]가 존재한다. CUBRID Shard는 데이터베이스를 Shard 단위로 수평 분할하여 각기 다른 물리 노드에 데이터를 분산 저장함으로써, 데이터의 분산 저장

및 병렬 처리를 지원한다. 그러나 CUBRID Shard는 각 클라이언트의 질의가 단일 서버에서 실행되는 경우만 병렬 처리가 가능하다는 단점을 보인다. 따라서 한 클라이언트의 질의가 다수의 서버에서 실행되어야 하는 경우에는 질의를 처리하지 못한다. 아울러, 해당 기법은 질의 처리 시 SQL 외에 shard hint를 요구하기 때문에, 사용자 편의성이 저하되는 단점이 존재한다.

이러한 문제점을 해결하기 위해, 본 논문에서는 CUBRID 기반 분산 미들웨어를 제안한다. 제안하는 미들웨어는 분산 저장된 데이터에 대해 병렬 질의 처리를 지원하는 관계형 DBMS로써, 기존 SQL에 익숙한 사용자들이 분산 저장된 데이터에 대한 병렬 처리 부담을 최소화하여 빅데이터 처리를 수행할 수 있다. 아울러, 집계 질의를 지원함으로써 다양한 질의 수행이 가능하다.

본 논문의 구성은 다음과 같다. 2장에서는 대표적인 RDBMS 기반 기법들을 소개하며, 3장에서는 제안하는 미들웨어의 전체 구조와 미들웨어 기반 질의 처리 수행 과정에 대해 상세히 설명한다. 4장에서는 제안하는 기법과 기존 기법과의 성능 비교를 수행한다. 마지막으로 5장에서는 결론 및 향후 연구를 제시한다.

### 2. 관련 연구

CUBRID[14]는 국내에서 개발된 대표적인 RDBMS 데이터베이스로써, 관계형 데이터베이스에 객체지향의 장점을 접목하여 만든 객체관계형 데이터베이스이다. CUBRID의 특징은 다음과 같다. 첫째, ACID 성질을 만족함으로써, 완벽한 트랜잭션을 지원한다. 둘째, 다양한 인덱스 기능을 지원함으로써, 질의 타입에 따른 최적화된 성능을 보

† Corresponding author

본 연구는 산업통상자원부 및 한국산업기술평가관리원의 산업기술혁신사업 결과로 수행되었음 (SW컴퓨팅산업원천기술개발 10041311)

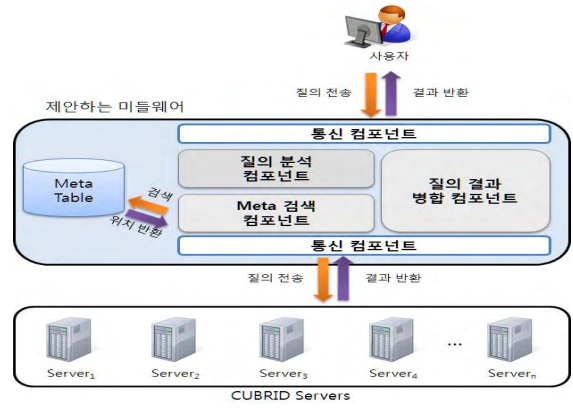
장하는 환경을 지원한다. 마지막으로 하드웨어, 소프트웨어, 네트워크 등에 장애가 발생해도 지속적인 서비스가 가능하게 하는 HA(High Availability) 기능을 제공한다. 하지만 CUBRID는 데이터의 분산 저장을 지원하지 못하기 때문에, 빅데이터에 대한 효율적인 처리를 지원하지 못하는 문제점이 존재한다. 이러한 문제점을 해결하기 위해 분산 처리기술을 지원하는 CUBRID Shard[13]가 개발되었다. CUBRID Shard는 데이터베이스를 Shard 단위로 수평 분할하여 각기 다른 물리 노드에 데이터를 분산 저장함으로써, 데이터의 분산 저장 및 병렬 처리를 지원한다. 수평 분할이란 스키마가 동일한 데이터를 행을 기준으로 두 개 이상의 테이블에 나누어 저장하는 방식을 말한다. 그러나 CUBRID Shard는 각 클라이언트의 질의가 단일 서버에서 실행되는 경우만 병렬 처리가 가능하다는 단점을 보인다. 따라서 한 클라이언트의 질의가 다수의 서버에서 실행되어야 하는 경우에는 질의를 처리하지 못한다. 아울러, 질의 처리 시 SQL 외에 shard hint를 요구하기 때문에, 사용자 편의성이 저하되는 단점이 존재한다.

### 3. CUBRID 기반 분산 미들웨어

본 장에서는 CUBRID 기반 분산 데이터에 대한 병렬 질의 처리를 지원하는 미들웨어를 제안한다. 제안하는 미들웨어는 분산 병렬 질의 처리를 지원하는 관계형 DBMS 로써, 기존 SQL에 익숙한 사용자가 편리하게 빅데이터 처리를 수행할 수 있도록 지원한다. 또한, 기존 분산 병렬 환경에서 지원하지 못하는 집계질의 (e.g., sum, avg, min, max, count)를 지원함으로써 다양한 질의 처리 수행이 가능하다.

#### 3.1 시스템 구조

<그림 1>은 제안하는 분산 병렬 질의 지원 미들웨어를 포함하는 CUBRID의 전체 시스템 구조를 나타낸다. 제안하는 미들웨어의 구조는 크게 통신 컴포넌트, 질의 분석 컴포넌트, Meta 테이블 및 meta 검색 컴포넌트, 질의 결과 병합 컴포넌트로 구성된다. 첫째, 통신 컴포넌트는 사용자와 미들웨어 혹은 미들웨어와 CUBRID 서버와의 데이터 송수신을 담당한다. 해당 컴포넌트에서 송수신하는 데이터는 SQL 및 질의 결과 등을 포함한다. 둘째, 질의 분석 컴포넌트는 사용자로부터 전송된 SQL 질의의 파싱(parsing) 및 집계 질의에 대한 SQL 변경을 수행한다. 셋째, meta 테이블은 데이터가 저장되어 있는 노드 정보를 저장하고, 사용자의 질의를 수행하기 위해 어느 CUBRID 서버에 질의를 전송해야 하는지를 판단하여 meta 검색 컴포넌트에게 반환한다. 이를 바탕으로, Meta 검색 컴포넌트는 해당 서버로 사용자 질의를 전송하는 역할을 수행한다. 마지막으로, 질의 병합 컴포넌트는 각 CUBRID 서버로부터 처리된 결과를 수신하여 질의 결과 병합을 수행한다.

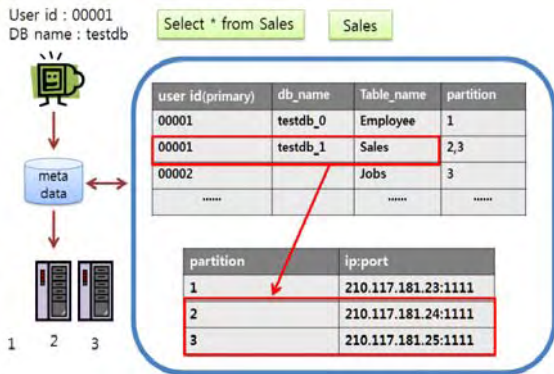


(그림 1) 제안하는 분산 미들웨어의 전체 구조

#### 3.2 병렬 질의 처리를 지원하는 분산 미들웨어

본 절에서는 제안하는 미들웨어에 대한 상세 설계 사항을 기술한다. 첫째, 통신 컴포넌트는 사용자와 미들웨어 혹은 미들웨어와 CUBRID 서버와의 데이터 송수신을 담당한다. 해당 컴포넌트에서 송수신하는 데이터는 SQL 및 데이터베이스 연결정보 {dbname, ip, port, user ID, password}를 포함한다. 둘째, 질의 분석 컴포넌트는 select 구문을 판별하고, 메타 정보 검색을 위해 from 구문을 추출한다. 아울러, 집계 질의를 분산적으로 처리하기 위해, 질의를 재구성한다. 예를 들어, average 질의의 경우, 각 CUBRID 서버에서 계산된 average를 통해서는 최종 결과를 도출할 수 없다. 따라서 각 CUBRID 서버가 sum 및 count를 반환하도록 질의로 재구성함으로써, 미들웨어에서 최종 average 값을 도출할 수 있다. 셋째, Meta 테이블은 {user ID, dbname, table name, server partition}을 저장한다. 이를 바탕으로, 사용자(user ID) 질의를 수행하기 위해 필요한 테이블(table name)이 어느 CUBRID server(server partition)에 저장되어 있는지를 판단한다. Server partition를 통해 table name에 대응하는 서버의 수와 각 서버의 ip, port 정보를 확인할 수 있다. 예를 들어, <그림 2>와 같이 미들웨어가 'select \* from Sales' 이라는 질의를 수신하면, user ID(00001)와 dbname(testdb), table name(Sales)을 기반으로 통해 질의 수행에 필요한 데이터를 보유하고 있는 CUBRID 서버를 확인한다. Sales 테이블은 2, 3번 서버에 분산 저장되어 있음을 확인하고, 이를 바탕으로 각 서버에 접속하기 위한 ip 및 port 목록을 탐색한다. 넷째, 마지막으로, 질의 병합 컴포넌트는 각 CUBRID 서버로부터 전송된 질의 결과를 병합한다. 이때, 각 서버에서 반환되는 질의 결과를 충돌 없이 수신하기 위한 메커니즘이 필요하다. 이를 위해, 미들웨어는 질의 처리를 수행하는 서버 수를 파악하여, 각 서버와의 통신을 위한 통신 프로세스를 생성한다. 또한, 각 서버에서 반환되는 질의 결과를 저장하기 위한 버퍼를 활성화한다. 이를 통해, 각 서버에서 병렬적으로 수신되는 질의 결과의 충돌 문제를 방지함으로써, 메시지 전송 효율을 향상시킨다.

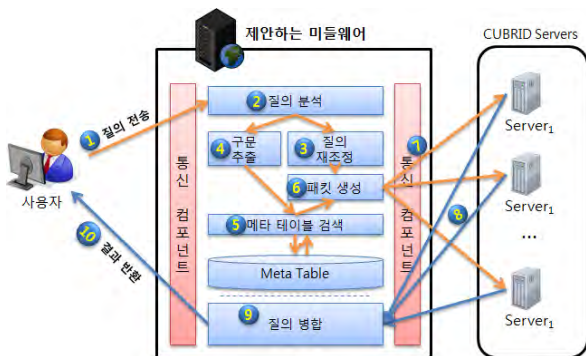
다. 아울러, 질의 병합 컴포넌트는 각 서버에서 전송된 질의 결과를 기반으로 중복 결과 제거 및 집계를 수행하여 최종 질의 결과를 추출하고, 이를 client에게 전송한다.



(그림 2) 메타 테이블 검색 과정

### 3.3 질의 처리 수행 과정

본 절에서는 제안하는 미들웨어에 대한 질의처리 수행 과정을 기술한다. <그림 3>은 제안하는 집계질의를 위한 분산 미들웨어의 질의 처리 수행과정을 나타낸다.



(그림 3) 제안하는 미들웨어 기반 질의 수행 과정

첫째, 사용자는 SQL 질의 및 데이터베이스 연결 정보를 미들웨어로 전송한다. 둘째, 미들웨어는 사용자의 SQL 분석을 통해 질의 유형을 파악한다. 해당 정보는 집계 질의에 대한 결과를 병합 시 활용된다. 셋째, 미들웨어는 분석한 내용을 바탕으로 SQL을 재구성한다. 넷째, 메타 테이블 검색을 위해 SQL에서 from 구문 추출을 수행한다. 다섯째, 메타 테이블을 기반으로 질의 수행에 필요한 데이터를 보유하고 있는 CUBRID 서버의 목록을 파악한다. 여섯째, 메타 테이블을 통해 탐색된 서버 접속 정보 및 재구성된 질의를 바탕으로 각 서버에 전송할 패킷을 생성한다. 일곱째, 미들웨어는 질의 처리를 수행할 각 서버에 질의를 포함한 패킷을 전송한다. 여덟째, 질의를 요청받은 각 CUBRID 서버는 자신이 보유하고 있는 데이터베이스 Shard를 기반으로 질의 처리를 수행하고 질의 결과를 미들웨어에 전송한다. 아홉째, 미들웨어는 질의 타입에 따라 각 서버에서 전송된 질의 결과를 병합하여, 최종 질의 결과를 추출한다. 이를 위해, 중복 결과 제거, 질의 결과 재정렬 및 집계 연산을 수행한다. 예를 들어, 'order by' 구

문이 포함된 'select' 질의의 경우, 각 서버에서 전송된 질의 결과에서 중복된 결과를 제거하고, 지정된 컬럼을 기반으로 질의 결과를 재정렬하여 최종 결과를 설정한다. 'min' 질의의 경우, 각 서버에서 전송된 질의 결과를 병합한 후, 병합된 질의 결과 내에서 최종 min 값을 추출한다. <표 1>는 질의 타입 별 최종 질의 결과 추출을 위해 수행할 내용을 나타낸다. 마지막으로, 질의 결과 병합을 통해 도출된 최종 질의 결과를 사용자에게 전송하여, 사용자의 질의 요청 처리를 완료한다.

<표 1> 미들웨어에서 지원하는 질의 유형별 수행 내용

질의 타입	미들웨어에서 수행할 내용
select	질의 결과 병합 후, 중복 제거를 통한 최종 결과 추출
oder by	질의 결과 병합 후, 정렬 기준(column 혹은 asc/desc)에 따라 질의 결과 정렬
min	질의 결과 병합 후, min 값 추출
max	질의 결과 병합 후, max 값 추출
count	질의 결과 병합 후, count 값 추출
sum	질의 결과 병합 후, sum 값 추출
average	질의를 sum 및 count로 재구성하여 전송하고, 질의 결과 병합 후 각 서버에서 전송된 sum 및 count 값을 기반으로 최종 average 값 추출

### 4. 성능평가

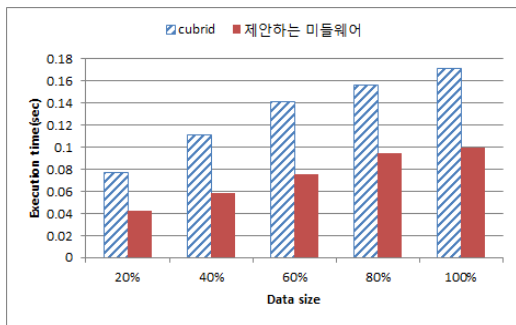
본 장에서는 제안하는 CUBRID 기반 분산 미들웨어의 성능을 검증하기 위해 질의 처리 시간 측면에서의 성능평가를 수행한다. CUBRID Shard는 데이터의 분산 저장은 지원하지 않지만, 한 클라이언트의 질의가 다수의 서버에서 실행되어야 하는 경우에는 질의를 처리하지 못한다. 즉, CUBRID Shard는 질의의 병렬처리를 지원하지 못하기 때문에, 성능평가 대상에서 제외한다. 따라서 제안하는 분산 미들웨어는 CUBRID와의 성능 비교를 수행한다. 한편, 기존 CUBRID는 분산 환경 및 병렬 처리를 제공하지 못하기 때문에, CUBRID의 데이터를 한 서버에 저장시켜 순차적으로 질의를 보내 처리하는 방식으로 성능평가를 수행하였다. 또한, CUBRID의 경우, 질의 결과 병합을 지원하지 못하기 때문에 병합 시간은 동일하다고 간주한다.

<표2>는 성능평가 실험 환경을 나타내며, 마스터 노드 1대와 슬레이브 노드 3대로 성능평가를 수행하였다. 성능평가는 CUBRID에서 제공하는 demodb의 game 테이블 데이터를 사용하여 수행한다. 해당 테이블은 총 11개의 속성으로 구성되며, 전체 튜플의 수는 8,653개이다.

<표 2> 실험 환경

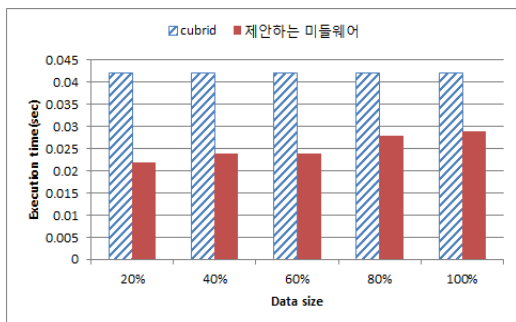
항목	성능
CPU	Intel@CoreTM i3-3240 CPU 3.40Ghz
Memory	8GB
O/S	Linux 3.5.0-23
Compiler	g++ 4.6.3
CUBRID 버전	2.2.0

<그림 4>는 데이터 수 변화에 따른 단순 select문 질의 처리 시간을 나타낸다. 데이터 양이 증가함에 따라, 제안하는 미들웨어 및 기존 CUBRID에서의 질의 처리 시간이 선형적으로 증가함을 확인할 수 있다. 전체 데이터의 60%에 대해 질의를 수행한 경우, CUBRID는 0.141초, 제안하는 미들웨어는 0.076 초가 소요된다. 이를 통해, 각 서버에서 병렬적으로 질의 결과를 수신하여 질의 병합을 수행하는 제안하는 미들웨어가 단순 select문 처리 시 보다 높은 처리 효율을 제공함을 확인할 수 있다.



(그림 4) select문 수행 시 데이터 양에 따른 처리 시간

<그림 5>는 데이터 수 변화에 따른 집계 질의 select문의 average 질의 처리 시간을 나타낸다. 기존 CUBRID의 경우 분산된 환경에서 집계질의를 지원하지 못하기 때문에 한곳에 저장된 데이터에 대한 AVG 집계질의 처리 시, SUM과 COUNT 대체하여 집계질의를 수행하였다. 데이터 양이 증가함에 따라, 제안하는 미들웨어 및 기존 CUBRID에서의 질의 처리 시간이 선형적으로 증가함을 확인할 수 있다. 전체 데이터의 60%에 대해 질의를 수행한 경우, CUBRID는 0.042초, 제안하는 미들웨어는 0.024 초가 소요된다. 이를 통해, 제안한 미들웨어가 분산 환경에서의 집계질의를 효율적으로 지원함을 확인할 수 있다.



(그림 5) average 수행 시 데이터 양에 따른 처리 시간

## 5. 결론 및 향후 연구

최근 빅데이터의 등장으로 인해 Hadoop을 비롯한 NoSQL에 대한 연구가 활발히 진행되고 있다. 그러나 NoSQL은 데이터베이스의 ACID 조건을 만족하지 못하는 문제점을 보인다.

따라서 본 논문에서는 CUBRID 기반 분산 데이터에 대

한 병렬 질의 처리를 지원하는 미들웨어를 제안하였다. 제안하는 미들웨어는 분산 저장된 데이터에 대해 병렬 질의 처리를 지원함으로써, SQL에 익숙한 사용자들이 분산 저장된 데이터에 대한 병렬 처리 부담을 최소화하여 빅데이터 처리를 수행할 수 있다. 아울러, 집계 질의를 지원함으로써 다양한 질의 수행이 가능하다. 또한, 성능 평가를 통해 제안하는 분산 미들웨어가 기존 기법에 비해 질의 처리 시간 측면에서 우수한 성능을 보임을 검증하였다.

향후 연구로는 미들웨어에 데이터 분산 저장 기능을 추가하고, 대용량 데이터를 통해 제안하는 미들웨어의 성능을 검증하는 것이다.

## 참고문헌

- [1] DEAN, Jeffrey; GHEMAWAT, 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] Apache Software Foundation, Apache Hadoop: <http://hadoop.apache.org/>.
- [5] Chodorow, Kristina. MongoDB: the definitive guide. " O'Reilly Media, Inc.", 2013.
- [6] Dietrich, André, et al. "ROS meets Cassandra: Data Management in Smart Environments with NoSQL."
- [7] Apache Software Foundation, Hadoop Map-Reduce: <http://hadoop.apache.org/mapreduce>.
- [8] Shvachko, Konstantin, et al. "The hadoop distributed file system." Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on. IEEE, 2010.
- [9] Chang, Fay, et al. "Bigtable: A distributed storage system for structured data." ACM Transactions on Computer Systems (TOCS) 26.2 (2008): 4.
- [10] Stonebraker, Michael. "SQL databases v. NoSQL databases." Communications of the ACM 53.4 (2010): 10-11.
- [11] Cattell, Rick. "Scalable SQL and NoSQL data stores." ACM SIGMOD Record 39.4 (2011): 12-27.
- [12] Han, Jing, et al. "Survey on NoSQL database." Pervasive computing and applications (ICPCA), 2011 6th international conference on. IEEE, 2011.
- [13] CUBRID Shard - <http://www.cubrid.org/manual/91/en/shard.html>
- [14] CUBRID - <http://www.cubrid.com/>