오픈소스 DBMS 성능비교분석

장래영* · 배정민* · 정성재** · 소우영* · 성경***

*한남대학교 컴퓨터공학과, **(주)스컴씨엔에스, ***목원대학교 컴퓨터교육과

Performance Comparison and Analysis between Open-Source DBMS

Rae-Young Jang* · Jung-Min Bae* · Sung-Jae Jung** · Woo-Young Soh* · Kyung Sung***

*Hannam University, **Sky Computing C&S, ***Mokwon University

E-mail : rene402@hnu.kr, bjmin86@nate.com, posein@naver.com, wsoh@hnu.kr, skyys04@mokwon.ac.kr

요 약

DBMS(Database Management System)는 다수의 사용자들이 데이터베이스에 접근하여 손쉽게 데이터를 사용할 수 있도록 해주는 소프트웨어 시스템이다. 오라클(Oracle)을 선두로 한 상용서비스들과 MySQL을 중심으로 하는 오픈소스 DBMS가 있다. MySQL이 오라클에 인수된 이후 MariaDB가 발표되어 수요가 증가하고 있으며, 기존 SQL과 다른 성격의 NoSQL DBMS들도 상황에 따라 관심이 늘어나고 있는 추세이다. 동일한 형태의 대용량 데이터들을 바탕으로 오픈소스 DBMS간 실제 성능비교분석이 필요함에 따라 본고에서는 오픈소스DBMS의 MariaDB와 문서중심(Document-Centric) 데이터베이스인 MongoDB간의 성능비교분석을 연구하였고, 나아가 그에 따른 결과를 바탕으로한 빅데이터관련 데이터베이스관리시스템을 제안하고자 한다.

ABSTRACT

The DBMS is a database management software system to access by people. It is an open source DBMS, such as MySQL and commercial services, such as ORACLE. Since MySQL has been acquired by Oracle, MariaDB released increase demand. NoSQL also are increasing, the trend is of interest, depending on the circumstances. Based on the same type of mass data, Depending on the performance comparison between the open source DBMS is required, and The study compared the performance between MariaDB and MongoDB. This paper proposes a DBMS for big data to process.

키워드

DBMS, MySQL, MariaDB, MongoDB, Big Data

I. 서론

관계형데이터베이스 모델은 1970년 E. F. Cold의 논문에서 처음 정의되었다. 관계형 데이터베이스는 일련의 정형화된 테이블로 구성된 데이터 항목들의 집합체로서, 그 데이터들은 데이터베이스 테이블을 재구성하지 않더라도 다양한 방법으로 접근하거나 조합될 수 있다.[1] 관계형 데이터베이스의 가치는 현재까지 꾸준히 이용되고 발전해올 정도로 뛰어남은 분명하다. MySQL은 현재 세계에서 가장 많이 쓰이는 제품으로 다중스레드, 다중사용자형식의 SQL형식의 데이터베이스관리시스템이다. Oracle 에 Sun Microsystems 가 인수되면서 MySQL은 Oracle 에 의해 개발방향을 제한받게 된다. 또한, 공식적으로는 MySQL을 존속할 것이라 발표했지만, 같은 RDBMS 계열인 Oracle과 비교해 언제까지 오픈소스로 지원될 지 알 수 없다. 이에 MySQL 개발자인 Michael “Monty“ Widenius에 의해 새로운 MariaDB가 개발된다. MariaDB는 MySQL과 높은 호환성을 유지하며, 점유율을 높여가고 있다. 이와 별도로 NoSQL을 활용한 시스템이 활성화되고 있다. 빅데이터시대를 맞아 RDBMS의 불편한 점을 NoSQL로 해소하고자 하는 것인데, 근본적인 지향점이 다르나 하나의 흐름임은 분명하다. 본고에서는 Document-Centric 방식의 MongoDB과 MairaDB를 비교해보고자 하였다.

Ⅱ. 관련연구

1. MySQL

MySQL은 현재 세계에서 가장 많이 쓰이는 오픈소스방식의 관계형데이터베이스관리시스템(RDBMS:Relational DataBase Management System)이다. 다중 스레드, 다중 사용자 형식의 구조질의어 형식의 데이터베이스 관리 시스템으로서 MySQL AB가 관리 및 지원하고 있었다. MySQL AB는 MySQL 라이선스에 의한 판매 지원 및 서비스 계약 시스템을 개발, 유지하고 또한 인터넷을 통한 전 세계의 협력자들을 고용해 왔다. 2008년 Sun MircoSystemsㅔ 인수되었다. 그러나, 2010년 Oracle이 Sun MircroSystems를 인수해버리면서, MySQL AB와 개발지침을 두고 이견이 발생한다. 이는 후에 MariaDB를 탄생시키는 시발점이 되었다. MySQL은 데이터베이스를 관리하거나 자료를 관리하기 위한 GUI 관리툴은 내장되어 있지 않다. 따라서, 사용자들은 CUI 도구들을 이용하거나 또는 데이터베이스를 만들고, 관리, 데이터를 백업, 상태 검사등을 하기 위해 MySQL 관리용 데스크톱 소프트웨어나 웹 애플리케이션을 사용해야 한다. 공식적인 MySQL 관리툴인 MySQL WorkBench는 오라클에 의해 개발되었으며, 공개되어 자유롭게 사용할 수 있다. 이를 이용하여 사용자들은 보다 쉽게 데이터베이스를 설계하고, 데이터베이스 모델링, SQL 관리, 데이터베이스 관리까지 할 수 있게 되었다.

2. MariaDB

앞서 언급했다시피 MySQL이 Oracle에 넘어가면서 MySQL AB와 Oracle 간에 개발방향에 있어 이견이 발생했다. MariaDB는 Monty Program AB라는 회사에서 만든 독립적인 RDBMS이다. Monty Program AB는 Michael “Monty“ Widenius가 동료들과 Sun MicroSystems를 나와 만든 회사로, MySQL 코드 베이스를 기반으로 MariaDB를 만들어냈다. MariaDB역시 오픈소스이며, 개발자인 Monty의 개발방향이 반영된 업그레이드된 MySQL이라고 봐도 무방하다. MariaDB에는 새로운 저장 엔진인 아리아(Aria)뿐만 아니라, InnoDB를 교체할 수 있는 XtraDB 저장 엔진을 포함하고 있다. 현재까지의 MariaDB는 MySQL과의 호환성이 뛰어난 편이다. 사용방법과 구조가 MySQL과 동일하다. 실제 MySQL의 Database를 가져와 그대로 사용해도 대부분 문제없이 돌아가는 편이다. 차이점은 순수한 오픈소스 프로젝트이기에 오라클로부터 자유롭다는 것이고, 앞으로 Monty Program AB의 개발방향에 따라 점차 바뀌어갈 것이라는 예상이다. 성능면에서는 MySQL과 비교해 리플리케이션 부분 속도가 약 4~5천배 정도 빠르며, 최고 70%의 향상을 보이고 있다고 개발사는 주장한다. HeidiSQL같은 관리툴을 사용할 수 있다.

3. NoSQL

NoSQL(Not Only SQL)은 비정형 스키마, 비관계형의 특성을 가진 대용량 데이터에 적합한 분산 데이터 저장소이다. 관계형 데이터베이스는 구조적 데이터 저장소의 솔루션으로서 가장 많이 사용되어 왔다. 하지만, 대규모 데이터를 저장하기 위해 스케일 아웃 방식으로 확장하기 어렵고, 일반적으로 데이터간 조인을 통해 조합하여 결과를 내고 있어 대용량 데이터 처리 시 병목현상이 발생할 수 있으며, 수십억 건 이상의 데이터를 저장하고 있는 테이블에 인덱스를 재구성한다거나 칼럼을 추가한다거나 하는 작업은 시스템 운영 상황에서 손쉽지 않다는 단점이 있다. 게다가, 스마트폰의 등장과 SNS의 이용으로 인해 이전에는 상상할 수 없었던 데이터와 트래픽이 발생하게 되었다. 이런 데이터처리를 위해 많은 양의 컴퓨팅 자원이 필요하게 되었는데, 기존 데이터베이스 시스템에서는 고비용의 문제에 직면하게 된다. 이에 기존의 관계형 데이터베이스와는 다른 형태의 데이터 저장소에 대한 요구 사항이 제시되었고, 여기에 부합한 데이터 저장모델을 별도로 개발해 나가기 시작했다. 구글의 Bigtable, 아마존의 Dynamo등과 같은 데이터모델과 관련 논문이 발표되면서 NoSQL에 대한 관심이 증가했고 그동안 특별한 용도로만 사용되던 일부 오픈소스들도 NoSQL로 분류되면서 NoSQL은 새로운 흐름이 되고있다. NoSQL은 인덱스와 데이터가 분리돼 별도로 운용되며, 스키마에 대한 정의가 자유롭다. 또, 저렴한 PC서버로 수평적 확장이 가능한 분산 아키텍쳐를 구성할 수 있어 초대용량 비정형 데이터 처리에 적합한 경량화 된 분산 데이터저장소라 할 수 있다. 또 관계형 데이터 모델이 아닌 key-value 또는 key-value를 응용한 데이터모델, 안정적이고 고가의 하드웨어가 아닌 다수의 값싼 하드웨어 이용, 데이터를 분산된 노드에 복제하여 저장되는 특성을 가지고 있을 때 NoSQL로 분류하며, 데이터모델에 따라 key-value, column, document, graph 로 구분할 수 있다.

표 1. NoSQL 분류

Data Model	NoSQL
Key-Value	Dynamo, Redis, Riak
Column	Bigtable, Hbase, Cassandra
Document	MongoDB, CouchDB
Graph	Neo4j

4. MongoDB

10gen에서 개발한 대표적인 NoSQL(Not Only SQL)인 MongoDB 는 BSON(Binary JSON) 형식으로

데이터를 관리하는 문서 기반의 데이터베이스로써 신뢰성과 확장성에 중점을 둔 빠르고 쉬운 데이터베이스이다. MongoDB는 확장성이 좋고 입출력 성능이 우수하며 비정형 데이터와 더불어 다양한 데이터를 수용할 수 있다. 하지만 맵리듀스 수행 시 단일 노드에서 수행되기 때문에 대량 데이터를 분석하기에는 한계가 있다. 그러나, 어느 정도 관계형 데이터베이스의 확장성 문제를 해결하였고 Single Machine, Master-Slave, Replica-Set, Shard 구성 등 요구사항에 따라 다양한 구성이 가능하며, 사용방법과 개념도 단순하고 SourceForge, Newyork Times, Craigslist, eBay, Foursquare 등 이미 다양한 시스템에서 사용 중이다. 또 MongoDB는 데이터를 분할해 다른 서버에 나누어 저장하는 과정인 샤딩(Sharding)을 제공하는데, 자동 샤딩을 통해 데이터를 분할하고 자동으로 재조정한다. 이는 MongoDB가 분산 확장하는 방식으로 애플리케이션에 영향을 주지 않고 증가하는 부하와 데이터를 처리하며 장비를 추가할 수 있도록 해준다. 또, MongoDB는 자체적으로 MapReduce를 지원하는데, 이는 별도의 분산/병렬 컴퓨팅을 위한 플랫폼 없이도 MongoDB에 저장된 데이터를 분산/병렬 처리로 빠르게 분석할 수 있게 해준다.

Ⅲ. 성능비교

각 데이터베이스별 성능비교를 위해 물리적인 PC에 vmware player 6.0 을 이용해 가상머신을 구현하였다. 동일한 ubuntu 14.04 운영체제에 데이터베이스만 달리하여 설치하였다.

표 2. 성능비교환경

	물리PC	가상PC	
OS	Window7 64bit	ubuntu 14.04 64bit	
CPU	i5-3550	dual core	
RAM	8GB	2GB	
DATABASE		MariaDB 5.5.39 Stable	MongoDB 2.6.5

성능비교를 위해 동일한 부하를 각 시스템에 발생시키고자 부하생성용 클라이언트를 구성해 비슷한 수준의 부하를 지속적으로 발생시켰다. vmstat 명령어를 이용해 초당 cpu 사용량를 측정하고, 읽기·쓰기연산속도를 측정하였다. 테스트 방법은 시스템별로 동시접속자수를 증가시키며 부하를 주었고, 쓰기속도는 Table에 지정된 데이터를 입력하도록 하였다. 읽기속도는 입력한 36만건의 데이터를 대상으로 측정하였다. 측정결과는 다음과 같다.

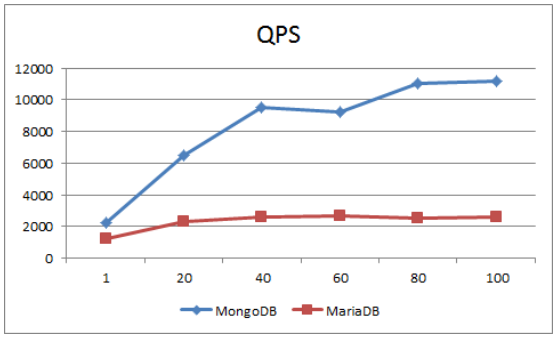

그림 1. QPS(Query Per Second) 측정값 비교

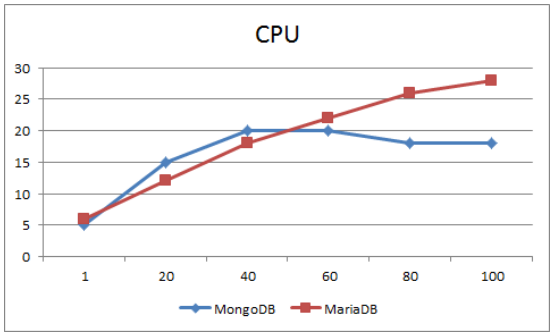

그림 2. CPU사용량 측정값 비교

[그림 1]과 [그림 2]는 DBMS별 쓰기,읽기 QPS와 CPU사용량을 측정/비교한 것으로 두 값을 비교해보면, 각 데이터베이스별 차이점을 알 수 있다. QPS부분에 있어서는 MongoDB가 관계형 데이터베이스보다 앞서는 값을 볼 수 있었다. 이는 대량의 데이터를 동시에 입력하는 경우 발생하는 결과값으로 상황에 따라 다를 것으로 예상된다. CPU사용량에 있어서도 MongoDB가 앞섬을 알 수 있었으나 그 차이는 크지 않았다.

4. 결론

빅데이터를 처리해야하는 상황에서의 NoSQL도입은 하나의 흐름이 되고 있다. 정확한 데이터 값이 필요한 경우가 아닌 상황, 예를 들어, SNS트렌드분석같은 경우 몇개의 데이터가 없다고 해서 결과값이 영향을 크게 미치진 않는 상황같은 데이터라든지, 분산처리, 병렬처리가 필요한 상황이라든지 이런 경우는 관계형 데이터베이스로는 한계가 있다. 그러나, 현재 서비스되고 있는 모든 시스템이 그런 것은 또 아니다. 상황에 따라서는 오히려 관계형 데이터베이스가 여전히 알맞은 경우도 많다. 본 고의 측정값을 보더라도 특정 경우에는 MongoDB가 앞서나 관계형 데이터베이스 역시 크게 떨어지는 수치는 아니었다. 오히려 방법을 달리해 적정한 인덱스처리를 해주었다면 다른 결과값이 나타났을지도 모른다. 분명한 것은

아직까지는 상황에 따른 데이터베이스를 선택해야한다는 것이다. 오랜 시간 축적되어진 데이터를 바탕으로 한 RDBMS는 안정적이고 어느정도의 성능이 보장되어있다. 특정환경에 있어서 관계형 데이터베이스의 한계를 극복하고자 도입된 NoSQL은 그 장점은 분명하지만 아직 안정성이나 대량의 데이터가 아닌 경우 관계형데이터베이스와 큰 차이가 없는 상황도 분명하다. 고로, 어떤 서비스를 제공할 것인지 계획하고 어떤 데이터를 처리할 것인지 예측하여 적절한 DBMS를 선택하는 것이 필요하다. 향후 이를 바탕으로 미흡한 점을 보강하여 실시간 대용량 데이터처리를 위한 NoSQL와 RDBMS간 성능비교를 진행하고자 한다.

참고문헌

[1] E. F. Cold, “A Relational Model of Data for Large Shared Data Banks”, 1970
[2] MySQL, http://dev.mysql.com/
[3] MariaDB, https://mariadb.org/en/
[4] MongoDB, http://www.mongodb.org