빅데이터 처리에 관한 NoSQL 비교연구

장래영* · 배정민* · 정성재** · 소우영* · 성경***

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

Comparative study on NoSQL for Processing a Big Data

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

빅데이터의 등장은 RDBMS로 대변되던 기존 데이터베이스 시장에 다양한 변화를 가져왔다. 빅데이터들은 데이터의 양은 증가했으나 개개의 데이터크기는 작아지고, RDBMS상의 데이터들과 비교해 단순해졌다. 이런 특징은 새로운 데이터처리기술을 요하게 되고, 그에 따라 빅데이터처리에 특화된 다양한 데이터베이스기술이 등장하게 되었다. 이를 NoSQL이라고 정의한다. NoSQL은 각각 데이터의 특성에 따른 처리방식이 달라 하나로 정의하기는 어렵다. 이에 본 논문에서는 다양한 NoSQL의 종류별 특징에 따라 분류하고 실제 빅데이터 운용에 있어 적합한 NoSQL을 알아보고자 한다.

ABSTRACT

The emergence of big data has brought many changes to the database management environment. the each amount of big data will increase, but each data size is smaller and simpler. This feature was required to a new data processing techniques. Accordingly, A variety database technology was provided to Specializing in big data processing. It is defined as NoSQL. NoSQL is how to use each different, according to the data characteristics. It is difficult to define one. In this paper, Classified according to the characteristics of each type of NoSQL Appropriate NoSQL is proposed.

키워드

MongoDB, HBASE, Redis, Cassandra, CouchDB, NoSQL

I. 서 론

현재는 빅데이터의 시대이다. 사람들은 이전의 그 어느때보다 더 방대하고 다양한 데이터들의 흐름속에 살아가고 있다. 빅데이터의 개념은 이전부터 존재하였지만 차츰 그 의미가 확장되어 간다. 인터넷과 SNS의 발달은 그 흐름을 더욱 빠르게 가져갔다. 이런 흐름은 데이터를 처리하는 방법의 변화에도 영향을 미쳤다. RDBMS로 대변되던 기존의 데이터베이스 처리기술과 다른 새로운 기술이 요구되었다. 한정적이고 전문적이고 예상가능한 데이터만 처리했던 것에서 시시콜콜한 일상의 이야기며 온라인상에서 새로운 소통, 관계를 맺어가고, 무거운 주제를 가볍게 대화해나가기도 한다. 심지어 누가 현재 어디에서 무얼하고 있는지 같은 신변잡기같은 이야기도 빠르게 전달되고 있다. 이런 일상적인 데이터들은 데이터의 양적증가를 불러왔다. 이런 빅데이터들은 데이터의 양은 증가했으나 개개의 데이터 크기는 작아졌고 기존의 RDBMS에서 처리하던 데이터들과 비교해 단순해졌다. 이런 특징은 RDBMS로 빠르게 처리, 분석이 어려워졌고 이에 따라 새로운 데이터처리기술이 필요해졌다. 이런 기술들은 기존의 SQL접근방식과 다른 접근방식을 제공하고 이를 NoSQL이라 정의한다. NoSQL은 SQL과 다른 다른 형태의 데이터베이스를 통칭하며 각각 데이터의 특성에 따른 처리방식이 달라 하나로 정의하기는 어렵다. 이에 본 논문에서는 다양한 NoSQL의 종류별 특징에 따라 분류하고 실제 빅데이터 운용에 있어 적합한 NoSQL을 알아보고자 한다

II. 본 론

2.1 빅데이터

먼저 “빅데이터란 무엇인가”라는 것에 대한 정의가 필요하다. 일반적으로 빅데이터란 데이터의 크기가 너무 크고 복잡하여 현재의 데이터처리기술로 다루기 어려운 것을 의미한다.[1] 추상적인 정의지만 정확한 맥을 짚고 있다. 특히 인터넷과 SNS의 결합으로 인한 데이터의 폭발적인 축적은 이를 뒷받침하고 있다. 더 이상 SNS분석을 통해 특정 결과를 예측하는 것이 불가능한 것이 아니게 되었다. 예를 들어 트위터(Twitter)를 실시간으로 분석해 이슈가 되는 단어를 나열해보면 현재 SNS사용자들의 흐름을 알아볼 수 있다. Tweetarchivist, Tweetmix, Tweettrend 등 분석사이트들을 이용하면 실시간으로 트위터의 데이터를 분석해 정보의 흐름을 한눈으로 파악해볼 수 있다. 게다가 이런 데이터의 축적은 컴퓨터를 비롯한 IT기기에 한정된 것만이 아니다. Jupiter Research에 따르면 스마트폰과 자동차의 결합이 활발히 연구중이며 이를 통해 자동차에서 이메일, SNS등을 통해 실시간 데이터를 생산해낼것으로 보고하고 있다. 실제로 최근의 자동차들은 인터넷과 결합된 서비스를 제공하고 있는데 이를 통해 위치정보, 개인의 SNS정보등이 더더욱 많이 축적될 것으로 예상된다.

2.2 NoSQL 정의와 특징

이런 다양한 데이터들의 축적이 예상되던 시기, 즉, 새로운 데이터유형에 따라 기존의 데이터저장시스템으로 처리하기 어려운 문제점을 해결하고자 새로운 형태의 데이터처리기술이 필요로 해졌고, 그즈음 빅테이블(Bigtable)과 다이나모(Dynamo)라는 두 개의 논문이 발표되었다. 각각 구글(Google)과 아마존(Amazon)의 프로젝트에서 제시한 결과물인데, 이 논문들로부터 새로운 데이터처리기술의 태동이 시작되었다. 그로부터 다양한 NoSQL기술이 등장하였는데, NoSQL은 Not Only SQL의 약자로 RDBMS로 대변되는 기존 데이터저장기술이 아닌 새로운 형태의 기술을 뜻한다. 하나 각각의 제품에 따라 그 특성이 달라 NoSQL은 무엇이다라 하나로 정의할 수 없는 특징이 있다. 그럼에도 NoSQL은 기존의 RDBMS와 다른 몇가지 공통된 특징을 지니고 있다. 첫째, NoSQL은 데이터간의 관계를 정의하지 않는다. 관계형데이터베이스(RDBMS)의 가장 큰 특징중 하나를 전면 부정하고 있다. Foreign Key(외래키), Primary Key(기본키), Join등의 개념을 갖고 있지 않다. 단순히 데이터테이블은 하나의 테이블일뿐이며, 테이블간의 관계도 정의하지 않고 일반적으로 Join연산도 불가능하다. 둘째, RDBMS에 비해 더 많은 데이터를 저장할 수 있다. 일반적으로 NoSQL들은 RDBMS를 뛰어넘은 대용량의 데이터를 저장하고 처리할 수 있다. 셋째, 분산형구조를 갖는다. NoSQL은 분산처리에 특화된 구조를 갖는다. 특히 하나의 고성능서버에 데이터를 저장하고 이를 동일한 다른서버에 분산저장하는 기술을 추가로 구성해줘야하는 RDBMS와 달리 다양한 일반PC서버 여러대를 연결하여 데이터를 저장하고 처리하는 구조를 갖는다. 이를 통해 유사시 데이터손실과 서비스중지를 최소화할 수 있다.

2.3 NoSQL의 분류

2.3.1 문서(Document)기반 데이터베이스

기본적으로 키/값 구조이나 저장되는 값의 데이터타입이 문서(Document)인 경우이다. 이런 구조는 값이 JSON, XML 등과 같이 정형화된 데이터타입으로 복잡한 계층구조를 표현할 수 있다. 데이터는 유연한 문서들의 묶으로 저장되며 조직화된다. 대표적인 NoSQL은 MongoDB, CouchDB, Riak 등이 있다. 그중 MongoDB같은 경우 일부 RDBMS적인 특징이 있어 기존 RDBMS개발자들이 접근하기 쉬워 현재 가장 많이 사용되고 있는 NoSQL이다.

2.3.2 키(Key)/값(Value) 저장 데이터베이스

데이터를 Key/Value 쌍으로 저장한다. 값은 Key를 통해 인출한다. 가장 기본적인 데이터베이스모델이다. Value에 다수의 값을 저장할 수 없어 Column Family라는 확장된 개념을 이용한다.

2.3.3 컬럼(Column)기반 데이터베이스

RDBMS와 유사하게 테이블로 데이터베이스를 저장하나 Row 대신 Column에 데이터를 저장한다. HBase, Cassandra등이 있다.

Ⅲ. 관련연구

3.1 MongoDB

MongoDB는 JSON형태의 문서로 데이터를 저장하는 오픈소스 문서지향 데이터베이스이다. RDBMS와 유사하기때문에 기존의 개발자들이 쉽게 접근할 수 있고, 객체지향프로그래밍방법론을 지향해 개발자들의 선호도가 높다. 또한, JSON문서의 내부요소를 색인으로 사용하면 검색속도를 높일 수 있어 많은 데이터가 추가되더라도 최적의 성능을 유지할 수 있다. 특히 MongoDB는 웹애플리케이션과 인터넷기반서비스를 위해 최적화되어있어 읽기/쓰기 효율이 높고 장애에 대한 대처가 쉬우며 확장이 용이하다. 더불어 문서 데이터모델의 사용으로 간결한 SQL 유사쿼리를 이용해 보다 직관적으로 데이터베이스를 제어할 수 있다. MongoDB의 특징은 다음과 같다. 첫째, 배우기쉽다. 최소한 다른 NoSQL보다 처음 접하기가 쉽다. 이는 RDBMS와 기초개념이 유사한점이 많기 때문이다. 둘째, 유연한 스키마를 지원한다. 데이터를 저장하기에 앞서 데이터구조를 정의할 필요가 없다. 셋째, 확장성이 높고 무료로 제공된다. 현재 SoundForge, Foursquare등에 적용되어 있다.

3.2 HBase

HBase는 빅데이터와 클라우드 서비스를 위한 대표적인 분산저장솔루션이다. 특히 빅데이터를 실시간으로 읽고 쓰기위한 데이터처리기술로 구글의 Bigtable을 계승하고 있다. 아파치 하둡 프로젝트의 일환으로 개발된 HBase는 HDFS(Hadoop Distributed File System)에서 동작하여 분산, 가변형 파일시스템으로 하둡과 마찬가지로 java로 개발되었다. HBase는 클라우드환경에 최적화되어 대용량의 데이터처리, 수백만에서 수백억개의 데이터를 다루는데 적합하다. HBase는 아파치재단 홈페이지(http://hbase.apache.org/, http://www.apache.org/dyn/closer.cgi/hbase)에서 다운로드할 수 있다. 현재 페이스북의 메시징플랫폼, 네이버의 라인등에서 사용되고 있다. 하둡은 사실상 기업형 빅데이터 처리에 있어 Map/Reduce 작업에 최선의 솔루션이다. 전에 Hadoop/HDFS를 사용해본 경험이 있다면 HBase는 좋은 선택이 될 것이다.

3.3 Cassandra

카산드라는 페이스북에 의해 설계, 개발되었다. 초창기 페이스북이 오픈소스를 기반으로 SNS를 제공하면서부터 대용량의 데이터를 저장, 관리하고자 하는 목적에서 출발하였다. 2008년 카산드라는 아파치에서 인큐베이터 프로젝트로 시작하여 2010년 탑레벨 프로젝트로 전환되었다. 현재 Adobe, eBay, HP, 페이스북, 트위터등과 같은 전세계 사용자들의 빅데이터를 처리하는 시스템에 적용되어 있다. 카산드라는 무료, 오픈소스, 분산 데이터처리솔루션이다. 기본설계개념은 peer to peer symmetric 노드를 기반으로 하고 시스템 실패시 싱글포인트 에러를 미연에 방지해주는 기능등을 제공한다. 더불어 매우 빠르게 쓰기를 수행할 수 있고, 수백 테라의 데이터를 저장할 수 있다. 또한 카산드라는 고가용성을 지원하며, 스키마가 없는 데이터 모델을 제공한다.

3.4 Redis

Redis는 Remote Dictionary System 의 약어로 메모리기반의 Key/Value 데이터베이스이다. 성능은 Memcached에 버금가면서 다양한 데이터형을 지원하고 저장되는 Value가 단순한 Object가 아니라 자료구조를 갖는다. 자료저장시 순간적으로 메모리에 있는 내용 전체를 디스크에 옮겨 저장한다. 특정 Key에 Value를 저장하는 구조이며 기본적인 Get/Put 명령을 지원한다. 전반적인 데이터의 액세스는 메모리에서 일어나지만 서버 재시작같은 물리적인 상황에서는 데이터를 보장하기 위해 일부 디스크를 사용하기도 한다. Redis는 즉각적인 데이터 변화같은 상황에 적합하다. 예를 들어 실시간 주식 가격변화, 실시간 분석, 1위 분석, 실시간 커뮤니케이션등에 최적화되어있다.

[표 1] NoSQL별 종류 및 특징

NoSQL	기술기반	특징
MongoDB	Document	JSON을 이용하기때문에 웹서비스영역에 특화. 다양한 API제공으로 인한 상대적으로 쉬운 접근성. 많은 곳에서 사용. 적당히 좋은 성능 제공.
HBase	Column	클라우드를 위한 분산형 솔루션. Hadoop기반에서 동작하고, 다양한 Hadoop 관련 도구들 사용가능 페타바이트수준의 대용량 데이터 처리 적합.
Cassandra	Column	Bigtable과 Dynamo의 장점을 결합.
Redis	Key/Value	대표적인 인메모리 기반 데이터베이스 매우빠른 쓰기/읽기속도 제공. SNS 등에 적합.

Ⅳ. 결 론

소개한 NoSQL이외 많은 종류의 데이터베이스솔루션이 존재한다. Riak, CouchDB, Accumulo, Hypertable, Neo4j 등등 수많은 NoSQL을 모두 알 수는 없다. 더군다나 필요에 의해 NoSQL 기술이 보급, 발전하고 있지만 기존의 RDBMS가 앞으로 사장될 기술인 것도 아니다. 하지만 분명한 것은 데이터베이스 기술은 RDBMS에서 빅데이터처리를 위한 NoSQL들로 점차 바뀌어나갈 것이다. 그에 새로운 NoSQL기술들을 기반기술별/특징별로 알아둘 필요성이 있다. 기본적으로 NoSQL은 수백만건 이상의 빅데이터처리에 적합하고, 그중에서도 상황에 맞는 적합한 솔루션을 선택해야하며, NoSQL에 맞는 적합한 데이터모델링작업도 필요하고, 적합한 하드웨어설계와 전문적인 DBA가 필요함은 분명하다. 웹서비스에 적합한 MongoDB, 대용량데이터저장및 처리에 적합한 HBase, SNS 같은 짧고 대량의데이터처리에 적합한 Redis 같은 NoSQL별 특징을 이해할 필요가 있다. 추후 RDBMS와 NoSQL간 동일한 빅데이터처리에 관한 성능비교는 다음 과제로 남겨둔다.

참고문헌

[1] wikipedia, http://en.wikipedia.org/wiki/Big Data
[2] Hadoop, http://hadoop.apache.org
[3] MongoDB, http://www.mongodb.org/
[4] HBase, http://hbase.apache.org/
[5] Cassandra, http://cassandra.apache.org/
[6] Redis, http://redis.io/
[7] http://kkovacs.eu/cassandra-vs-mongodb-vs-couchdb-vs-redis
[8] CouchDB, http://couchdb.apache.org/
[9] Riak, http://basho.com/riak/