

다양한 NoSQL 데이터베이스의 성능 평가 연구

박홍진*

상지대학교 컴퓨터정보공학부

A Study about Performance Evaluation of Various NoSQL Databases

Hong-Jin Park*

요약 대용량의 데이터를 처리하기 위해서는 기존 관계형 데이터베이스인 MySQL, PostgreSQL, Oracle 등 보다 NoSQL 데이터베이스가 우수함을 나타내고 있다. 본 논문은 일반적으로 널리 사용되고 있는 NoSQL 데이터베이스 중 HBase, Cassandra, MongoDB, Redis의 성능을 비교 평가한다. 본 논문에서 대용량의 데이터를 분산 처리하기 위해 서버 12대를 스위칭 허브를 통해 연결하였으며 운영체제로 우분투를 설치하였다. 벤치마킹 도구로는 YCSB를 이용하였으며, read, update를 각각 50%, 50%, 두 번째로는 95%, 5%, 마지막으로는 100%, 0%로 3가지로 변경하였고, 200,000번에서 1,200,000 까지 명령어를 수행하여 비교 평가하였다. 초당 트랜잭션 처리는 Cassandra가 가장 우수하였고, 단위 시간당 프로세스 수행 처리 개수는 MongoDB가 우수함을 보였다.

Abstract Various NoSQL databases are more excellent to process a large amount of big data than existing relational databases such as MySQL, PostgreSQL and Oracle. Among widely used NoSQL databases, performance of HBase, Cassandra, MongoDB and Redis was comparatively assessed. For distributed processing of a large amount of data, 12 servers were connected through switching hub and Ubuntu was installed as operating system. As for benchmark tool, YCSB was applied. Read and update ratios changed from 50% and 50%, 95% and 5% and finally, 100% and 0% and each of them was assessed as 200,000 commands developed into 1,200,000 commands for each case. Cassandra was most excellent with transaction processing per second while MongoDB was most excellent with the number of processes carried out per unit time.

Key Words : HBase, Cassandra, MongoDB, Redis, NoSQL

1. 서론

정보가 단순한 검색과 포털 위주였던 시대에서 개인화 서비스와 SNS를 제공하는 환경으로 빠르게 진화 되고 있다. 정보 통신 기술 시장조사 기관인 IDC 보고서에 따르면 전 세계 디지털 데이터 양이 제타바이트(Zetabyte) 단위로 2년마다 2배씩 증가하여 2020년에는 약 40제타바이트가 될 것이라

고 예측하고 있다. 데이터 양이 폭발적으로 증가함에 따라 이를 해결하기 위한 빅 데이터 관심이 고조되고 있는 것이다. 기존 데이터에 비해 대용량의 빅 데이터는 데이터 양(Volume), 속도(Velocity) 그리고 다양성(Variety) 측면에서 다르게 정의되어진다. 데이터의 양은 기존 메가바이트나 기가바이트에서 위치정보, 동영상 같은 데이터를 다루기 위해 테라바이트나 제타바이트 정도의 대용량의 데이

This work has been supported by a Sangji University Research Grant 2015

* Corresponding Author : School of Computer, Information and Communication Engineering, SangJi University(hjpark1@sangji.ac.kr)

Received June 14, 2016

Revised June 16, 2016

Accepted June 21, 2016

터 양을 의미한다. 속도 측면에서는 다양한 센서나 스마트 폰 등에서 실시간 적으로 생성되는 데이터 생성 속도를 해결할 수 있어야 한다. 다양성 측면은 기존 정형 데이터 뿐만 아니라 SNS으로 전달되는 동영상이나, 개인 사진, RFID 태그정보, 센서 정보등 비정형 데이터 처리를 지원해야한다. 이러한 대용량 데이터를 처리하기 위해서는 기존 정보화 시대에서 사용되었던 관계형 데이터베이스(Relational database)에서 비(Non) 관계형 데이터베이스로 변화하고 있다. 기존 데이터베이스 시스템은 구조적인 데이터를 기반으로 저장하기 때문에 데이터 일관성과 정규화를 사용하고 있고 고성능을 제공하고 있다. 그러나 이러한 구조로는 비구조적인 데이터 처리와 제타급 이상의 빅 데이터 처리 수행은 매우 어렵거나 복잡하다. 즉, 대용량, 비정형, 실 시간적인 빅 데이터 속성을 처리할 수 있는 새로운 저장 방식이 필요하다 기존 데이터베이스의 약점을 보완하면서 빅 데이터를 해결하기 위해 연구되는 기술은 NoSQL이다[1-3].

NoSQL의 성능을 비교하는 논문으로 Barbierato의 논문은 맵리듀스(MapReduce) 기반의 새로운 모델링 언어인 SIMTHESysBigData를 통해 HBase 데이터베이스를 평가 하였다[4]. Bernardino 논문은 NoSQL 데이터베이스 중 MongoDB와 Cassandra를 비교 하였으며[5], Gandimi가 제시한 논문은 HBase, Cassandra, MongoDB 데이터베이스를 코어(Core)나 쓰레드(Thread)의 개수를 변경하면서 YCSB 도구를 통해 비교 평가 하였다[6].

본 논문은 기존 NoSQL 성능 비교 논문[4-6]과 다르게 일반적으로 널리 사용되고 있는 NoSQL 데이터베이스 중 HBase, Cassandra, MongoDB, Redis를 비교 평가 하였으며, 코어나 쓰레드의 개수가 아닌 물리적 분산 환경을 위해 서버 12대를 스위칭 허브로 연결하여 비교 평가 하였다. 수행 평가 도구로는 야후의 오픈 소스 기반의 NoSQL 벤치마킹 도구인 YCSB를 이용하였다. YCSB는 다수의 NoSQL 솔루션에서 공통적으로 사용할 수 있는 작업부하(Workload)를 생성하여 쉽고 표준화된 방식으로 성능 비교를 수행할 수 있도록 해주

는 벤치마크 도구이다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문에서 비교 평가되는 NoSQL 데이터베이스인 HBase, Cassandra, MongoDB, Redis를 알아본다. 3장에서는 본 논문에서 성능 평가를 위해 성능 평가 구성도와 NoSQL 데이터베이스를 비교 평가를 수행한다. 끝으로 4장에서 본 논문에 결론을 맺는다.

2. 기반 연구

NoSQL 데이터베이스는 크게 키-값 저장(key-value store) 방식, 문서 저장(document store) 방식, 컬럼 저장(column store) 방식으로 구분될 수 있다[7]. 키-값 저장 기반의 NoSQL 데이터베이스는 키(key)와 값(value)이라는 형태로 자료를 보관한다. 문서 저장 기반 NoSQL 데이터베이스는 MongoDB와 CouchDB 가 있으며 키-값 형과 다르게 복잡한 검색 조건을 통해 데이터를 추출할 수 있다. 컬럼 기반 NoSQL 데이터베이스는 뛰어난 확장성을 지니고 있으며, Cassandra, HBase등이 있다. 본 논문에서 성능 평가에 사용되는 데이터베이스는 키-값 저장 방식으로 Redis, 문서 저장 방식으로 MongoDB, 컬럼 기반 방식으로 Cassandra, HBase 이다.

2.1 HBase

HBase는 데이터의 일관성과 스케일 아웃이 뛰어난 컬럼 기반 데이터베이스이다[8]. 페이스북 북, 트위터에서 사용되고 있는 HBase는 하둡 플랫폼을 위한 공개 NoSQL 데이터베이스이다. 대용량 데이터를 다루는 NoSQL 데이터베이스 중 Cassandra와 함께 가장 많이 사용되고 있다. HBase는 중앙에 완전 분산 시스템을 통제하는 마스터를 두고 복제된 전체 데이터의 일관성을 관리하기 때문에 분산된 복제 데이터의 일관성이 보장된다. 또한 HBase는 HDFS(Hadoop Distributed File System) 및 맵리듀스 등과 사용하면 최적화시킬 수 있다. HBase의 장점은 데이터를 효율적으로 압축할 수 있다. 행 기반에 비해 컬럼 기반으로

데이터를 저장하는 것이 데이터 압축을 더 효율적으로 할 수 있다. 데이터를 압축해서 저장함으로써 대량 데이터를 보다 경제적으로 관리하고 대량 조회가 발생할 경우 압축된 데이터를 가져오기 때문에 대역폭 측면에서도 우수한 성능을 지니고 있다. HBase는 이전 버전의 데이터 값도 관리가 가능하다. 지난 버전 데이터를 통한 통계나 변경 내용에 대한 분석도 가능하다. 또한 웹 콘솔을 통해 데이터베이스 관리 및 모니터링을 할 수 있는 기능을 제공하고 있다. 단점으로는 다른 NoSQL 데이터베이스에 비해 이해하기가 복잡하다. 효율적으로 관리하기 위해서는 HDFS, 주키퍼(Zookeeper) 등 다른 시스템을 파악해야만 한다. 또한, 특정 범위의 키-값의 저장 요청이 집중되는 경우에 응답 속도가 느려지는 문제가 발생할 수 있다. 또한, 파일 정보를 관리하는 네임노드에서 장애가 발생하면 전체 서비스를 중단이 발생하게 된다. HBase는 3가지 모드, 즉 단일 모드(stand-alone mode), 의사 분산 모드(pseudo-distributed mode), 완전 분산 모드(fully-distributed mode)를 지원한다. 본 논문은 여러 대의 서버 상에서 수행되는 완전 분산 모드로 수행한다.

2.2 Cassandra

Cassandra는 자바 기반으로 페이스 북에서 개발하여 현재는 아파치 프로젝트로 진행되고 있는 키-값 방식의 NoSQL 데이터베이스이다[9]. Cassandra 데이터베이스에서는 데이터를 먼저 메인 메모리에 저장되며, 메모리의 데이터가 일정 크기가 되면 하드 디스크로 옮겨지는 방식으로 데이터를 저장한다. Cassandra 데이터베이스의 특징은 일관성 있는 해싱(Hashing) 방식으로 로드 밸런싱을 수행하고 있다. 요청이 많은 서버의 작업들이 요청이 적은 서버로 부하를 전송시킴으로써 전체 시스템의 서버의 부하를 줄일 수 있다. 따라서 대량의 요청이 특정 서버에 몰리는 것을 방지하며, 장애를 예방하고 성능을 높일 수 있다. 또한 새로운 서버가 추가될 경우 로드 밸런싱을 고려하여 부하 관리가 효율적으로 이루어지게 된다. 새로운 서버가 분산 시스템

에 추가할 경우 서비스가 중단 없이 쉽게 추가할 수 있는 장점도 있다. Cassandra 데이터베이스의 단점으로는 복잡한 조건의 검색은 불가능하다. 로우 키(Row key)와 컬럼 두 가지에 대한 인덱스만 가능하기 때문에 검색 조건이 단순한 서비스에 적합하다. 데이터 대한 갱신 및 입력 시 원자적(Atomic) 처리가 어렵다.

2.3 MongoDB

MongoDB는 10gen 회사에서 개발된 높은 성능과 확장성을 지니고 있는 문서 기반 방식의 데이터베이스이다[10]. MongoDB의 가장 큰 특징은 스키마를 정의할 필요가 없다는 점이다. 데이터를 입력할 때 데이터 구조 정보를 포함하여 BSON(JSON을 바이너리 한 것) 형식으로 저장하기 때문에 스키마를 별도로 정의할 필요가 없는 것이다. 스키마가 없으므로 프로그램 코드에만 관리하면 된다. MongoDB는 스키마를 정의하진 않지만 관계형 데이터베이스와 마찬가지로 임의의 키-값에 대한 복잡한 검색이 가능하다. 관계형 데이터베이스와 비교하여 응답 속도가 빠르고 관계형 데이터베이스처럼 인덱스를 추가하여 처리 속도를 더 빠르게 수행할 수 있다. MongoDB에서 JOIN이나 트랜잭션 처리가 불가능하다. 또한 데이터 갱신 및 입력 시 바로 디스크에 쓰지 않는다. 디스크에 쓰기가 비동기식으로 이루어지기 때문에 경우 따라서는 데이터가 유실될 가능성이 있다.

2.4 Redis

Redis 데이터베이스는 데이터를 키와 값이라는 형태로 저장한다[11]. Redis 데이터베이스의 큰 특징은 기본적으로 데이터를 메모리에 저장하기 때문에 처리 속도가 빠르다. Redis는 스냅샷(Snapshot)라는 기능이 있어 데이터베이스 현재 상황을 하드 디스크에 저장할 수 있다. 스냅샷 시점에서 전체 데이터에 대한 I/O가 발생되나, 해당 I/O는 순차적 접근으로 처리되므로 I/O로 인한 부하는 크지 않다.

3. 성능 비교 평가

3.1 성능 평가 구조도

본 논문은 일반적으로 널리 사용되고 있는 NoSQL 데이터베이스 중 HBase, Cassandra, MongoDB, Redis의 성능을 비교 평가한다. 성능을 비교 평가를 위해 [그림 1]처럼 서버 12대를 스위칭 허브로 연결하여 수행했다. 본 논문에서는 여러 노드가 하나의 클러스터로 함께 작동하는 분산 모드로 수행한다.



그림 1. 성능 평가를 위한 물리적 구조
Fig. 1. Physical structure for performance evaluation

또한 [그림 1] 서버 설정에서 설치된 소프트웨어 구조는 [표 1]과 같다. 12대의 서버를 연결하여 우분투 12.04 LTS 운영체제를 설치 한 후 HBase, Cassandra, MongoDB, Redis의 성능을 비교 평가하였다. 성능을 비교 평가하기 위해 사용한 도구는 YCSB이다. YCSB는 다수의 NoSQL 솔루션에서 공통적으로 사용할 수 있는 작업부하(Workload)를 생성하여 쉽고 표준화된 방식으로 성능 평가를 수행할 수 있도록 해주는 벤치마크 도구이다.

표 1. 성능 평가를 위한 소프트웨어
Table 1. Software for performance evaluation

벤치 마킹 도구	YCSB			
NoSQL	HBase 0.94.9	Cassandra 1.2.6	MongoDB 2.4.5	Redis 2.6.14
OS	Ubuntu 12.04 LTS			

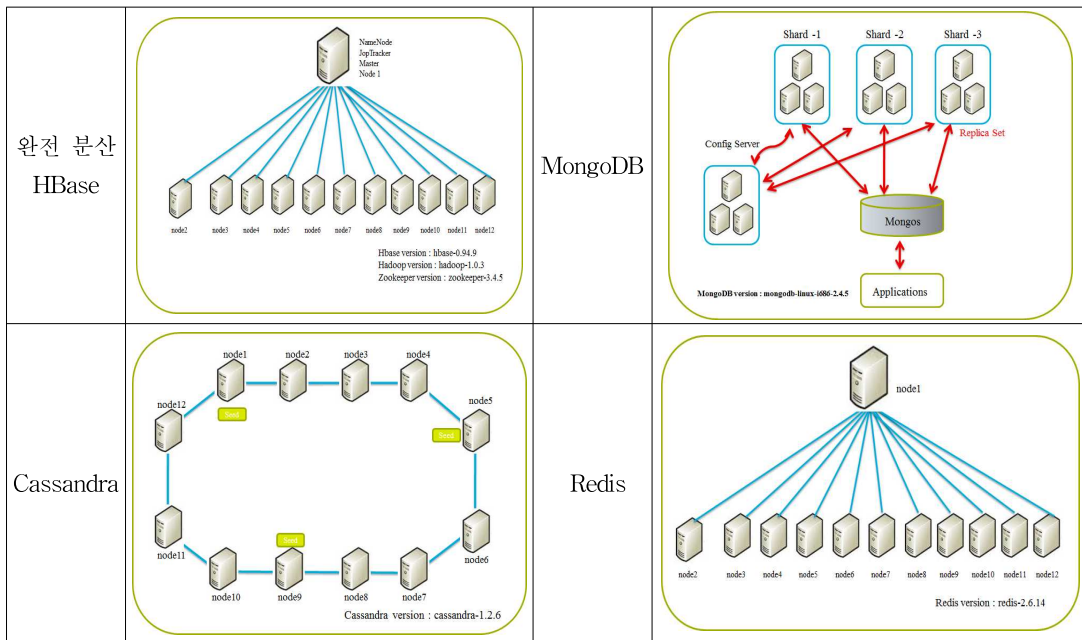


그림 2. 성능 평가를 위한 논리적 구조
Fig. 2. Logical Structure for performance evaluation

[그림 2]는 본 논문에서 사용된 NoSQL의 논리적 구조를 나타내고 있다. HBase는 완전 분산 구조로 구성되어 있다. 중앙에 네임노드에서 전체 시스템을 관리하며 중앙 노드를 포함한 나머지 노드는 데이터 노드로 구성되어 있다. Cassandra의 각 노드들이 서로 통신할 때 가십(Gossip) 프로토콜을 사용하며, 씨드(Seed)는 새로운 노드에 연결된 링 구조를 알려주는 역할을 수행한다. MongoDB에서 샤딩(Sharding)을 통해 데이터를 다른 서버에 나누어서 저장하는 모습을 나타낸다. MongoDB에서 샤딩은 데이터를 분할하여 다른 서버에 나누어 저장하는 방식을 의미한다. 샤딩을 설정할 때에는 샤드 키를 선택하고 그 키의 값을 데이터를 분할하는 데 사용한다. MongoDB에서 Mongos는 샤드를 관리하고 요청을 라우팅하는 역할을 수행한다. 또한, 클러스터의 메타 데이터를 지속적으로 유지하기 위해 설정 서버(Config Server)를 설정해야한다. Redis 데이터베이스는 키 값의 형태를 가지는 사전 기반의 데이터베이스이다 Redis 하나의 노드가 중앙 노드로 나머지 노드를 관리하는 구조 구성되어 있다.

3.2 성능 평가 결과

본 논문에서는 12대의 서버 위에 HBase, Cassandra, MongoDB, Redis를 설치 한 후 YCSB를 이용하여 성능을 평가 하였다. YCSB로 read, update 명령어를 200,000번, 400,000번, 600,000번, 800,000번, 1,000,000번. 그리고 1,200,000번으로 각각 변화시키면서 초당 트랜잭션(transactions) 처리의 수를 측정하였다. 다양한 비교 평가하기 위해 크게 3가지로 구분하여 평가하였다. 먼저 작업 로드 A에서는 read 50%, update 50%를 수행하였으며, 작업 로드 B에서는 read 95 %, update 5%로 비교평가 하였다. 마지막으로 작업 로드 C에서는 read 100%, update 0%일 때 비교평가 하였다.

[그림 3]은 작업 로드 A에서 read 50%, update 50%를 200,000번에서 1,200,000번 변화 하면서 초당 트랜잭션 처리 수를 나타내고 있다. [그림 3]을 보면 Cassandra가 초당 트랜잭션 처리 수 가 가

장 우수함을 나타내고 있으며, MongoDB는 가장 적게 트랜잭션이 처리 되고 있다. 특히, 1,200,000번 처리할 경우 HBase와 비교하면 약 2.5배 빠르게 트랜잭션이 처리되고 있다.

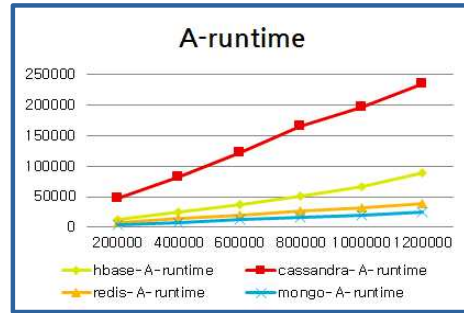


그림 3 초당 트랜잭션 처리 개수 - read 50%, update 50%

Fig. 3. The number of transactions per second 4 read 50%, update 50%

[그림 4]는 작업 로드 A에서 read 50%, update 50%를 200,000번에서 1,200,000번 변화 하면서 단위 시간당 프로세스 수행 개수를 나타낸다. 이 경우를 보면 MongoDB가 가장 우수함을 보이고 있다. 두 번째로는 Redis가 우수하였으며, Cassandra와 HBase는 비슷한 성능을 보였다.

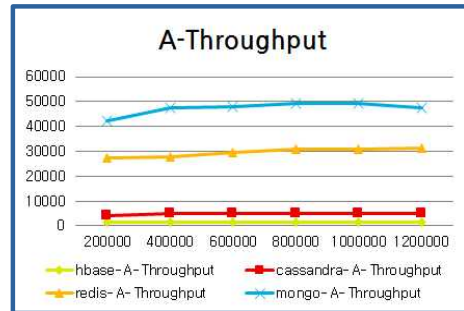


그림 4. 처리량 - read 50%, update 50%

Fig. 4. Throughput - read 50%, update 50%

[그림 5]은 작업 로드 A에서 read 50%, update 50%를 200,000번에서 1,200,000번 변화 하면서 평균 지연 시간을 나타내고 있다. 평균 지연 시간이 높을수록 성능이 낮게 보이고 있다. 전체적으로 Cassandra가 다른 NoSQL DB에 비해 3.5배

의 높은 평균 지연 시간을 나타내고 있다. HBase, Redis, MongoDB의 평균 지연 시간을 비슷한 결과를 보이고 있다.

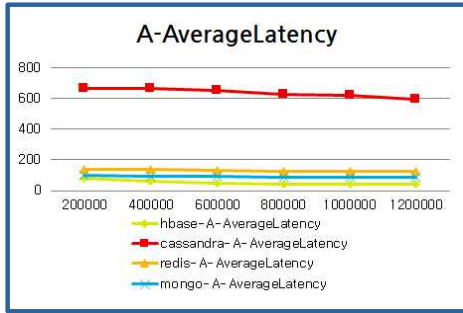


그림 5. 평균지연시간 - read 50%, update 50%
Fig. 5. Average Latency - read 50%, update 50%

[그림 6]은 작업 로드 B에서 read 95%, update 5%를 200,000번에서 1,200,000번 변화 하면서 초당 처리되는 트랜잭션 개수를 나타내고 있다. 작업 로드 A와 비슷하게 Cassandra가 가장 많이 트랜잭션을 처리하고 있다. 1,200,000번 처리할 경우 HBase와 비교하면 약 3.2배 빠르게 트랜잭션이 처리되고 있다. [그림 6]을 보면 Redis와 MongoDB는 비슷하게 트랜잭션을 처리하고 있으면 성능이 낮음을 보이고 있다.

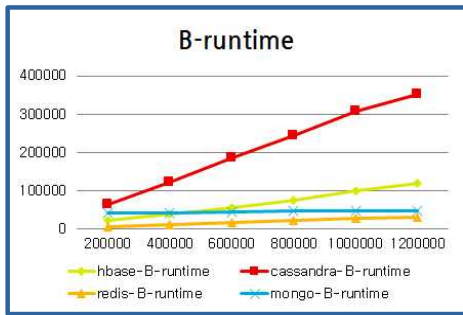


그림 6. 초당 트랜잭션 처리 개수 - read 95%, update 5%
Fig. 6. The number of transactions per second - read 95%, update 5%

[그림 7]은 작업 로드 A에서 read 95%, update 5%를 200,000번에서 1,200,000번 변화 하면서 단위 시간당 프로세스 수행 개수를 나타낸다. 이 경우를 보면 MongoDB가 가장 우수함을 보이고 있다.

두 번째로는 Redis가 우수하였으며, Cassandra와 HBase는 비슷한 성능을 보였다.

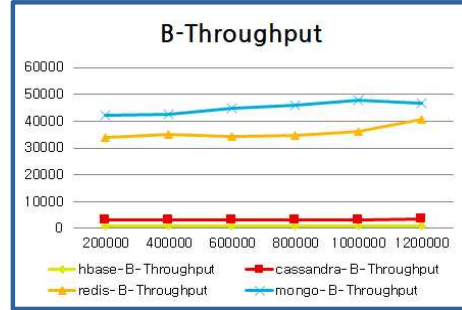


그림 7. 처리량 - read 95%, update 5%
Fig. 7. Throughput - read 95%, update 5%

[그림 8]은 작업 로드 A에서 read 95%, update 5%를 200,000번에서 1,200,000번 변화 하면서 평균 지연 시간을 나타내는 것으로 높을수록 성능이 낮게 보이고 있다. Cassandra가 다른 NoSQL DB에 비해 매우 높은 평균 지연 시간을 나타내고 있다. HBase, Redis, MongoDB의 평균 지연 시간을 비슷한 결과를 보이고 있다. HBase, Redis, MongoDB의 평균 지연 시간을 비슷한 결과를 보이고 있다.

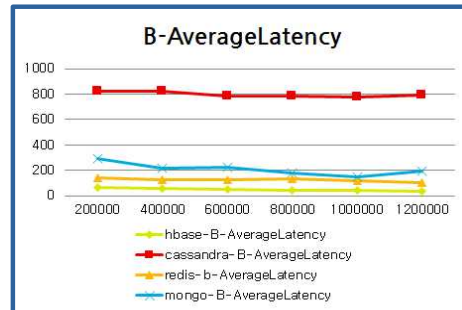


그림 8. 평균지연시간 - read 95%, update 5%
Fig. 8. Average Latency - read 95%, update 5%

[그림 9]는 작업 로드 C에서 read 100%, update 0%를 200,000번에서 1,200,000번 변화 하면서 초당 처리되는 트랜잭션 개수를 나타내고 있다. 작업 로드 A, B와 비슷하게 Cassandra가 가장 많이 트랜잭션을 처리하고 있다. 1,200,000번 처리할 경우 HBase와 비교하면 약 4배 빠르게 트랜잭션이 처리되고 있다. [그림 9]를 보면 Redis와 MongoDB는 비슷하게 트랜잭션을 처리하고 있으면 성능이 낮음을 보이고 있다.

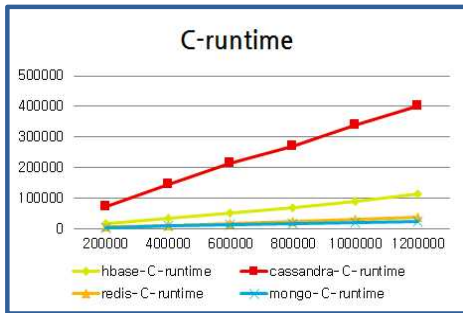


그림 9. 초당 트랜잭션 처리 개수 - read 100%, update 0%
 Fig. 9. The number of transactions per second - read 100%, update 0%

[그림 10]은 작업 로드 C에서 read 100%, update 0%를 200,000번에서 1,200,000번 변화 하면서 단위 시간당 프로세스 수행 개수를 나타낸다. 이 경우를 보면 MongoDB가 가장 우수함을 보이고 있다. 두 번째로는 Redis가 우수하였으며, Cassandra와 HBase는 비슷한 성능을 보였다.

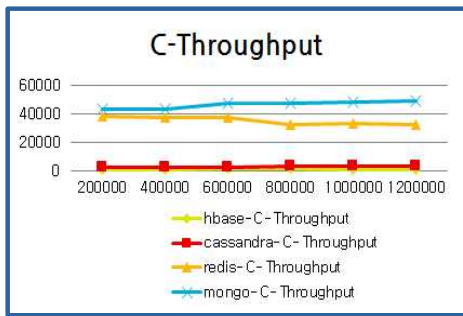


그림 10. 처리량 - read 100%, update 0%
 Fig. 10. Throughput - read 100%, update 0%

[그림 11]은 작업 로드 C에서 read 100%, update 0%를 200,000번에서 1,200,000번 변화 하면서 평균

지연 시간을 나타내는 것으로 높을수록 성능이 낮게 보이고 있다. MongoDB가 다른 NoSQL DB에 비해 매우 높은 평균 지연 시간을 나타내고 있다. HBase, Cassandra의 평균 지연 시간이 낮게 나타내고 있다.

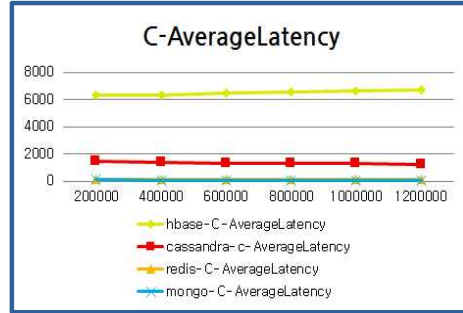


그림 11. 평균지연시간 - read 100%, update 0%
 Fig. 11. Average Latency - read 100%, update 0%

4. 결론

MySQL, PostgreSQL, Oracle 같은 관계형 데이터베이스보다 NoSQL 데이터베이스는 대용량의 데이터를 여러 서버에게 분산하여 저장 및 처리가 쉽다. 또한, 서버 처리 능력 관점에서 보면, 스케일업보다 가격이 비교적 저가인 여러 서버를 이용하여 데이터를 처리 시키는 스케일 아웃(scale-out) 방식이 더 적합하다. NoSQL 데이터베이스는 크게 키-값 저장 방식, 문서 저장 방식, 컬럼 저장 방식으로 구분될 수 있다.

본 논문은 NoSQL 데이터베이스인 HBase, Cassandra, MongoDB, Redis의 성능을 비교 평가하였다. 12대의 서버를 연결하여 우분투 12.04 운영체제를 설치 후에 각 NoSQL을 설치하였다. 설치된 NoSQL 데이터베이스를 비교 평가하기 위해 YCSB 벤치마킹 도구를 설치하였다. 본 논문은 NoSQL의 벤치마킹을 도구인 YCSB을 이용하여 크게 3가지로 구분하여 평가 하였으며 수행 명령어는 각 200,000번 부터 1,200,000번으로 변경하여 비교 평가 하였다. 먼저 작업로드 A에서는 read 50%, update 50%를 수행 하였으며, 작업 로드 B에서는 read 95%, update 5%로 비교평가 하였다. 마지막으로 작업 로드 C에서는 read 100%, update 0%일 때 비교평가 하였다. 작업

로드 A에서는 초당 트랜잭션 처리 개수는 Cassandra가 가장 우수하였으며, 처리율은 MongoDB가 우수하였다. 또한 평균 지연시간은 Cassandra가 다른 NoSQL 보다 평균적으로 3.5배 높은 결과를 나타내었다. HBase, Redis, MongoDB는 read 50%, update 50%일 때 평균 지연 시간이 비슷하였다. 작업 로드 B에서는 초당 트랜잭션 처리 개수는 Cassandra가 가장 우수하였으며, 처리율은 MongoDB가 우수하였다. 또한 평균 지연시간은 Cassandra가 다른 NoSQL 보다 높은 결과를 나타내었다. 작업 로드 C에서 초당 트랜잭션 처리 개수는 Cassandra가 가장 우수하였으며, 처리율은 MongoDB가 우수하였다. 또한 평균 지연시간은 MongoDB 다른 NoSQL 보다 높은 결과를 나타내었다.

본 논문에서 결론적으로 보면, 초당 트랜잭션 처리 수는 Cassandra가 가장 우수한 성능을 보였다. 또한, 당위 시간당 프로세스 수행 처리 수는 MongoDB가 우수함을 보였으며, 끝으로 평균 지연시간은 Cassandra가 가장 높은 수치를 나타내었다.

REFERENCES

[1] Jing Han, "Survey on NoSQL database", Proceedings of the International Conference on Pervasive Computing and Applications (ICPCA), 2011.

[2] Ameya Nayak, Anil Poriya, Dikshay Poojary, "Type of NOSQL Databases and its Comparison with Relational Databases", International Journal of Applied Information Systems, Vol. 5, No. 4, Mar. 2013.

[3] Rick Cattell, "Scalable SQL and NoSQL data stores", ACM SIGMOD, Vol. 39, Issue 4, Dec. 2010.

[4] Barbierato, E., Gribaudo, M., Iacono, M., "Performance evaluation of NoSQL big-data applications using multi-formalism models", Vol. 37, Future generations computer systems, 2014.

[5] Jorge Bernardino, "NoSQL databases: MongoDB vs cassandra", Proceedings of the International Conference on Computer Science and Software Engineering, 2013.

[6] A. Gandini, M. Gribaudo, W. J. Knottenbelt, R. Osman, P. Piazzolla, "Performance Evaluation of NoSQL Databases", No. 8721, Lecture Notes in Computer Science, 2014.

[7] Bogdan George Tudorica, "A comparison between several NoSQL databases with comments and notes", Proceedings of the International Conference on Networking in Education and Research, 2011.

[8] Mehul Nalin Vora, "Hadoop-HBase for large-scale data", Proceedings of the International Conference on Computer Science and Network Technology, 2011.

[9] Guoxi Wang, Jianfeng Tang, "The NoSQL Principles and Basic Application of Cassandra Model", Proceedings of the International Conference on Computer Science & Service System, 2012.

[10] <https://www.mongodb.com/>

[11] <http://www.redis.io/>

저자약력

박 홍 진(Hong-Jin Park)

[중심회원]



- 1995년 8월 : 중앙대학교 일반대학원 컴퓨터공학과 (공학석사)
- 2001년 8월 : 중앙대학교 일반대학원 컴퓨터공학과 (공학박사)
- 2001년 9월 ~ 현재 : 상지대학교 컴퓨터정보공학부 교수
- 2006년 1월 ~ 현재 : 조달청 심사위원

<관심분야>

분산 시스템, 서버