Journal of Korea Multimedia Society Vol. 21, No. 8, August 2018(pp. 934-941)
https://doi.org/10.9717/kmms.2018.21.8.934

HL7 FHIR 기반 의료 데이터 처리 시스템에서 YCSB를 통한 RDBMS와 MongoDB의 성능 분석 연구

전동철[†], 이병문[††], 황희정[†††]

Performance Analysis of RDBMS and MongoDB through YCSB in Medical Data Processing System Based HL7 FHIR

Dong-cheol Jeon[†], Byung Mun Lee[††], Heejoung Hwang[†††]

ABSTRACT

There are some limits on cost and efficiency for large amount of data in RDBMS, and NoSQL is starting to gain popularity. In medical institutions, data forms are different between organizations, and that makes difficulty for interoperability between organizations. In this paper we focused on performance issues between RDMBS and NoSQL in medical documents. We had built two different environment and had experiment comparative analysis of NoSQL with RDBMS based on medical data. We used medical HL7 FHIR as a medical data standard. Also YCSB benchmark tool was used for performance comparison. Experiments shows that NoSQL has better performance in large amounts of medical data processing systems that have over 10,000~100,000 records.

Key words: MongoDB, MySQL, HL7, FHIR, RDBMS, NoSQL, YCSB, Database

1. 서 론

최근 IT 기술들의 급격한 발전으로 인하여 데이터의 양이 기하급수적으로 증가하게 되었다. 스마트폰 보급률이 높아지고 웨어러블 디바이스 및 IoT 기술이 높아지면서 데이터들은 하루에도 셀 수 없을 정도로 많이 발생하고 있고, 사람들의 모든 행동들이 데이터가 되어 활용되고 있다. 이로 인해, 처리해야하는 데이터의 양 또한 과거에 비해 급격히 증가하게 되었고, 이를 저장하고 처리하는 능력에 대한 중요성 역시 증가하게 되었다[1]. 기존에 유행했던 MySQL, OracleDB, Postgre 등의 RDBMS(Relational Database Management System)는 대량의 데이터를 처리하는 것에 대한 비용과 효율적인 측면에서 한계를 보이기 시작하였고, 그에 따라 효과적으로 대량의 데이터를 관리하고 처리하기 위한 NoSQL이 각광받기 시작했다[2]. Fig. 1은 데이터베이스의 순위로 아직까지 Oracle, MySQL, MSSQL이 부동의 상위권을 유지하고 있지만, NoSQL의 가파른 성장세를 보여주는 현황이다[3].

의료 분야 역시 IT의 발전으로 이전과는 다른 새로운 시대의 길을 열기 시작했다. 기존의 방대한 의료 데이터와 IT 기술의 융합을 기반으로 과거의 의료 기록 및 지식들이 방치되지 않고 데이터로 사용 되면

※ Corresponding Author : Heejoung Hwang, Address: (13120) Seongnamdaero 1342, Seongnam-si, Gyeonggi-do, Korea, TEL : +82-31-750-4758, FAX : +82-, E-mail : hwanghj@gachon.ac.kr
Receipt date : Jul. 12, 2018, Approval date : Jul. 20, 2018
[†] Dept. of Computer Engineering, Gachon University (E-mail : aseah@naver.com)
[††] Dept. of Computer Engineering, Gachon University (E-mail : bmlee@gachon.ac.kr)
[†††] Dept. of Computer Engineering, Gachon University
※ This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center)support program(IITP-2018-2017-0-01630) supervised by the IITP(Institute for Information & communications Technology Promotion).

Rank Jul 2018	Rank Jun 2018	Rank Jul 2017	DBMS	Database Model	Score Jul 2018	Score Jun 2018	Score Jul 2017
1.	1.	1.	Oracle	Relational DBMS	1277.79	-33.47	-97.09
2.	2.	2.	MySQL	Relational DBMS	1196.07	-37.62	-153.04
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1053.41	-34.32	-172.59
4.	4.	4.	PostgreSQL	Relational DBMS	405.81	-4.86	+36.37
5.	5.	5.	MongoDB	Document store	350.33	+6.54	+17.56
6.	6.	6.	DB2	Relational DBMS	186.20	+0.56	-5.05
7.	7.	↑ 9.	Redis	Key-value store	139.91	+3.61	+18.40
8.	8.	↑ 10.	Elasticsearch	Search engine	136.22	+5.18	+20.25
9.	9.	↓ 7.	Microsoft Access	Relational DBMS	132.58	+1.59	+6.45
10.	10.	↓ 8.	Cassandra	Wide column store	121.06	+1.84	-3.07

Fig. 1. Database Ranking Top-10.

서 이의 중요성이 증가하기 시작했다. 하지만, 현실적으로 기관에 따라 데이터 표현 형식의 차이가 존재하고 그에 대한 효율적인 교류가 이루어지지 않아 어려움을 겪고 있다[4]. 이를 극복하기 위해 HL7 FHIR, ISO, OHDSI 등에서 다양한 의료 데이터 표준이 나오게 되었다. 그 중에서 HL7 FHIR는 JSON, XML 등의 개발자들에게도 편리한 표준들을 제공한다는 점에서 큰 주목을 받고 있다[5].

본 논문에서는 의료 데이터에 대해 의료 기관 별 데이터 표현 방법의 차이를 극복하고 효과적인 의료 데이터 교환 플랫폼을 구축하기 위해 HL7 FHIR의 환자 의료 데이터 표준을 이용해 RDBMS와 NoSQL 간의 성능 분석을 진행했다. 성능 비교 분석은 Yahoo의 YCSB(Yahoo Cloud Serving Benchmark)라는 데이터베이스 벤치마크 툴을 사용하였고, 실험의 결과로 MySQL과 MongoDB 간의 성능 차이는 MongoDB가 MySQL에 비해서 대량의 데이터의 처리에 있어서 더 우수한 성능을 보이는 것을 확인했다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문 연구의 기반이 되는 RDBMS와 NoSQL, 그리고 HL7 FHIR에 대한 이론에 대해서 알아본다. 3장에서는 의료 데이터에 대한 NoSQL 적용 제시, 그리고 4장에서는 제안한 실험에 대해서 비교 분석하고 결과를 도출한다. 마지막으로 5장에서는 결론을 내린다.

2. 관련 연구

2.1 RDBMS

RDBMS는 데이터베이스 중에서 가장 많이 사용되는 관계형 데이터베이스의 관리 시스템으로 행(row)과 열(column)로 구성이 된 2차원 테이블 형태의 데이터 모델에 기초를 둔 데이터베이스를 생성, 수정, 삭제 등의 관리를 할 수 있는 시스템을 말한다. RDBMS의 특징은 SQL(Structured Query Language)이라는 언어를 사용하여 데이터를 다룰 수 있으며, 직관적인 문법을 사용한다. 또한, 데이터를 속성(Attribute)와 값(Value) 사이에서 관계를 찾아내고 테이블 모양으로 도식화하는 구조화를 한다. RDBMS는 처음 데이터베이스를 만든 후, 관련 프로그램의 변형 없이 새로운 데이터 항목을 데이터베이스에 추가하는 것이 가능하다. 그리고 데이터를 분류, 정렬, 탐색하는 속도가 빠르다.

대표적인 RDBMS 중 하나는 MySQL이다. MySQL은 Oracle에서 제작한 무료 오픈 소스 DBMS로 다양한 운영체제에서 지원이 가능하고, 다중 사용자 및 다중 쓰레드를 지원하며, 다양한 프로그래밍 언어들을 위한 API도 제공하기 때문에 가장 많이 쓰이고 일반 사용자도 사용하기 쉬운 DBMS 중 하나이다. 하지만 RDBMS 및 MySQL은 2차원 테이블 형태를 기초로 하기 때문에 오히려 유연성이 떨어진다는 큰 단점이 있다. 즉, 상황에 따라 프로그래머가 데이터 모델을 쉽게 개선하기가 어렵다. 따라서 RDBMS로 정형화 되지 않고 가변성이 높은 데이터에 대응하는 것은 시간과 비용에 대한 효율성이 많이 떨어진다.

2.2 NoSQL

NoSQL은 RDBMS의 단점을 보완하기 위해 나온 데이터베이스로써 관계형 모델을 사용하지 않기 때문에, 데이터 모델에 대해서 유연하게 대처할 수 있

다. 또한 스키마가 없는 비정형화된 데이터 모델에 최적화 되어있다. 마지막으로 저비용 하드웨어의 분산 클러스터를 사용하여 스케일 아웃 하도록 설계가 되었다.

NoSQL의 종류는 크게 4종류로 나뉜다. 첫 번째로 열 기반(Columnar) 데이터베이스는 데이터의 열을 읽고 쓰는데 최적화된 데이터베이스이다. 전반적인 디스크 I/O 요구 사항과 디스크에서 로드해야 하는 데이터의 양을 획기적으로 줄이기 위해서 데이터베이스 테이블에서는 열 기반 스토리지가 분석 쿼리 성능에 중요한 요소가 된다. 두 번째로 문서(Document) 데이터베이스는 JSON, XML 형식의 문서와 같은 반정형 데이터를 저장하도록 설계되었다. 기존 관계형 데이터베이스와는 달리, NoSQL 문서에 따라 다른 스키마를 적용할 수 있어서 더 유연하게 데이터를 구성 및 저장하고 선택적 값에 필요한 스토리지를 줄일 수 있다. 세 번째로 그래프(Graph) 데이터베이스는 간선이라고 부르는 방향 링크와 정점을 저장해서 NoSQL 데이터베이스에 구축된 데이터들과 관련된 속성을 가진다. 마지막으로 인 메모리 키-값 스토어(In-Memory Key-Value Store)는 읽기 중심의 혹은 컴퓨팅 집약적 워크로드에 최적화된 NoSQL 데이터베이스이다. 핵심 데이터 조각을 메모리에 저장해 접근 지연 시간을 줄여주어 성능을 향상 시키는 데이터베이스이다[6].

MongoDB는 문서 데이터베이스로 10gen에서 개발한 데이터베이스이다. MongoDB는 데이터를 입력할 때 BSON(JSON), XML 형식으로 저장하기 때문에 스키마를 정의할 필요가 없다는 것이 큰 장점이다. 또한 관계형 데이터베이스와 마찬가지로 키-값에 대한 복잡한 검색이 가능하다. 관계형 데이터베이스에 비해 응답속도가 빠르고 인덱스를 추가하여 더 빠르게 처리할 수 있다. 하지만 MongoDB에서는 JOIN이나 트랜잭션 처리가 불가능하며, 데이터를 추가하는 것이 비동기식으로 이루어지기 때문에 데이터가 유실될 수 있다는 단점이 있다[7].

2.3 HL7 FHIR

HL7은 Health Level 7의 약자로 서로 다른 의료정보시스템 간에 효과적인 정보 교환과 유지 보수에 유연하게 대처하기 위한 표준을 설계하기 위해 1987년 미국에서 조직된 표준화 기구이다[8]. 늘어나는 헬스케어 서비스 및 융합 서비스에서 병원 및 지역마다 서로 다른 데이터 표준을 적용하고 있어서 어려움을 겪고 있었고, 이에 HL7은 헬스 데이터의 상호 운용성을 위한 표준을 제정하게 된다. HL7에 의한 의료 기관의 변화는 서로 다른 시스템 간의 의료 데이터를 교류할 수 있게 되었으며, 모든 의료 데이터의 전산처리로 의료 데이터 활용성이 높아졌다. 또한 표준화된 의료 코드로 인터페이스 시스템 개발이 용이해졌으며 체계적인 시스템 구성으로 시스템의 유지보수가 용이해졌다.

FHIR는 Fast Healthcare Interoperability Resources의 약자로 의료 기기 간에 상호 운용성을 보장하기 위해서 HL7에서 개발한 프레임워크이다. FHIR의 특징은 다음과 같다. 웹 기반 기술로 사용 환경이 편리하다. 기존의 HL7의 프레임워크인 V2 Message, V3 RIM(Reference Information Model), CDA(Clinical Document Architecture)의 장점들을 결합하여 비용 절감의 효과를 기대할 수 있다. JSON 및 XML 등 개발자가 활용하기 쉬운 표준을 제공하여 개발자들에게 편리함을 준다. 기존 패러다임에 유연한 아키텍처를 제공한다.

HL7 FHIR의 표준을 활용하는 대표적인 사례로는 SMART on FHIR가 있다. SMART(Substitutable Medical Applications and Reusable Technology)는 개방형 표준 기반 기술 플랫폼으로 개발자가 의료 시스템 전체를 원활하고 안전하게 실행할 수 있는 응용 프로그램을 만들 수 있는 플랫폼을 제공한다. CEM(Clinical Element Model)의 FHIR 프로파일링을 통해, EHR 시스템을 재활용 및 공유 가능한 헬스케어 플랫폼으로 확장하는 것이다. Fig. 2는 SMART on FHIR의 아키텍처이다.

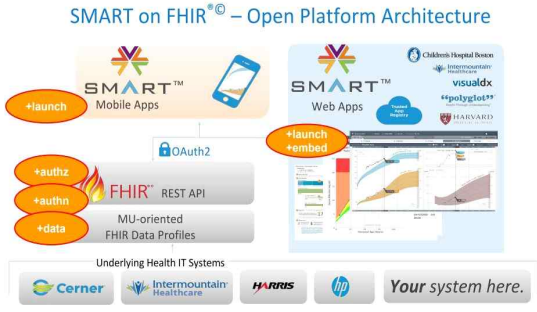

Fig. 2. SMART on FHIR architecture.

3. 의료 정보 데이터에 대한 NoSQL 적용

3.1 NoSQL 적용의 필요성

서론에서 언급했듯이, 의료 데이터는 IT 기술의 융합과 빅데이터의 발전으로 인하여 데이터에 대한 중요성이 높아지고, 의료 데이터 역시 활용 분야가 넓어지고 있다. 하지만 의료 기관 별 데이터 유형 차이로 상호 데이터 교류의 어려움을 겪고 있다. RDBMS는 데이터 스키마가 정해져있고 수정이 어렵다는 점에서 의료 정보 데이터의 효과적인 상호 교환을 기대하기 어렵고 방대한 양의 데이터에 대해서 효율적인 측면이 다소 떨어지는 편이다. 이를 해결하기 위해서 NoSQL의 적용이 절실하다. NoSQL은 데이터 스키마가 RDBMS와 달리 유동적으로 변할 수 있기 때문에 각기 다른 데이터의 구조에 있어 차이가 있더라도 대처하기가 쉽다. 또한 NoSQL은 분산 클러스터를 이용하여 분산 저장이 가능하기 때문에 의료 데이터의 양이 커지더라도 쉽게 대응이 가능하다. 또한 NoSQL을 이용한 의료 데이터 표준이 많이 나타나고 있고, HL7 FHIR 역시 이러한 특징에 맞추어 XML, JSON 같은 표준을 제공하고 있다.

3.2 의료 정보 데이터 스키마

HL7 FHIR에는 다양한 의료 데이터에 대한 포맷들이 존재한다. 그 중에서 이번 실험에서는 환자(Patient) 표준을 참조하여 데이터 구조를 제작했다. RDBMS의 스키마는 Table 1과 같은 형태를 가진다.

비슷한 형태로 실험을 진행하기 위해 NoSQL의 스키마도 RDBMS의 스키마와 유사하게 구성했다. NoSQL은 JSON 형태의 Document(문서) 형태로 예

Table 1. Schema of Patient in RDBMS

Table Name	Column Name	Type
Patient	id	Int
	resourceType	Text
	meta	JSON
	text	JSON
	identifier	JSON
	active	Text
	gender	Text
	birthDate	Date
	deceasedBoolean	Boolean

```
{
  "_id":ObjectId,
  "resourceType":String,
  "id":String,
  "meta":Object{
          "versionId":String,
          "lastUpdated":Date
  },
  "text":Object{
          "status":String,
          "div":String
  },
  "identifier":Array[
                Object{
                 "use":String,
                 "type":Object{
                         "text":String,
                         "value":String,
                         "assigner":Object{
                                 "display":String
                         }
                 }
  ],
  "active":Boolean,
  "gender":String,
  "birthDate":String,
  "deceasedBoolean":Boolean
}
```

Fig. 3. Schema of Patient in JSON structure.

시는 Fig. 3과 같다.

3.4 성능 분석 방법

성능 분석 방법으로는 YCSB를 사용했다. YCSB는 Yahoo Cloud Serving Benchmark로 클라우드 서비스를 제공하는 데이터베이스의 성능을 평가하기 위한 벤치마크 프레임워크이다. YCSB는 두 가지로 구성이 된다. 첫 번째는 YCSB Client로 확장 워크로드를 생성해준다. 두 번째는 Core workloads로 확장 워크로드 생성기에서 실행되는 작업 부하 시나리오 세트를 제공한다[9]. 클라우드 서비스의 요청 패턴과 유사하도록 다중 클라이언트 기능을 제공하고, 워크로드를 사용자의 요구에 맞게 수정이 가능하다는 장점이 있다. YCSB는 실제 서비스 요청과 유사한 패턴을 낼 수 있다. YCSB의 아키텍처는 Fig. 4와 같다. YCSB를 사용하여 CRUD 동작의 비중을 다르게 설

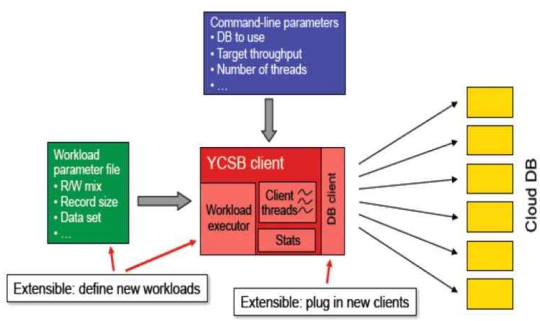

Fig. 4. Architecture of YCSB [10].

정하여 요청 패턴을 실험할 수 있으며, YCSB에서 제공하는 대표적인 워크로드는 Table 2와 같다. 워크로드 별로 초당 명령어 처리속도와 평균 대기시간 등 성능 비교에 필요한 지표들을 얻을 수 있다[11].

현재 YCSB가 제공하는 대표 워크로드는 A부터 F 타입까지 있다. Table 2에서 Operation은 실행하는 워크로드의 연산 비중을 뜻하는데, A의 Update heavy를 예로 들자면 전체 연산 중에서 Read 연산을 50%, Update 연산을 50% 진행하는 것을 뜻한다. 또한 Record selection은 연산할 레코드를 검색하기 위해 어떤 방법을 쓸 것인지를 뜻한다. Zipfian은 지프 분포를 적용해 레코드를 탐색하는 것이고, Latest는 최근의 데이터, Uniform은 Uniform Distribution으로 연속균등분포를 통해 레코드를 검색하는 것이다. Table 2의 각 값들은 기본으로 설정이 되어있으며 수정이 필요하면 Operation의 비중, Record Selection의 타입 등을 설정이 가능하다. 각각 연산에서 사용되는 쿼리문의 경우, 삽입에는 임의의 값들을 삽입하여 데이터를 입력하는 것이고, 수정, 삭제, 검색은 기본 키 값에 의한 단순 수정, 삭제, 검색으로 이루어져 있다.

4. 실험 및 결과

4.1 실험 환경

성능 분석 실험을 위해 동일한 환경에서 데이터베이스를 구축하였다. 1대의 Ubuntu 16.04 운영체제의 컴퓨터에서 실험을 진행했으며 실험에 사용한 데이터베이스로는 RDBMS에서 MySQL, NoSQL에서 MongoDB를 사용하였다. 성능 분석은 앞 장에서 말한 YCSB 툴을 이용하여 특정 워크로드를 실행하는 방식을 사용하였다. 실험 환경에 대한 구조는 Table 3과 같다.

4.2 실험 내용

의료 데이터에 대해 앞에서 언급했듯이 HL7 FHIR의 환자 데이터 표준을 참고하여 데이터를 구성하였고 데이터는 임의의 랜덤 값으로 저장했다. 데이터에 대해서 CRUD 중 Read를 비교하기 위해 Read only workload인 workload C를 사용하여 의료 데이터의 수를 500, 1000, 5000, 10000, 100000 개까지 늘리며 실험을 진행하였다. C 타입의 workload는 Read 100%의 연산을 진행하고 그 결과를 Runtime, Through-

Table 2. The core workload in YCSB [12]

Workload	Operation	Record selection	Application Example
A:Update heavy	Read: 50% Update: 50%	Zipfian	Session store recording recent actions in a user session
B:Read mostly	Read: 95% Update: 5%	Zipfian	Photo tagging; add a tag is an update, but most operations are to read tags
C:Read only	Read: 100%	Zipfian	User profile cache, where profiles are constructed elsewhere (e.g., Hadoop)
D:Read latest	Read: 95% Insert: 5%	Latest	User status updates; people want to read the latest statuses
E:Short ranges	Scan: 95% Insert: 5%	Zipfian/ Uniform	Threaded conversations, where each scan is for the posts in a given thread (assumed to be clustered by thread id)
F:Read-modify-write	Read: 50% Read-modify-write : 50%	Zipfian	User database, where user records are read and modified by user or to record user activity

Table 3. Environment of performance system

Performance Evaluation	Yahoo Cloud Serving Benchmarking(YCSB) 0.14.0	
Database	RDBMS	NoSQL
	MySQL 8.0.11	MongoDB 3.6.5
OS	Ubuntu 16.04 LTS	

out, AverageLatency를 기준으로 각각 도식화했다. Runtime은 테스트하는 workload의 실행 시간, Throughout은 초당 명령어 처리 시간(Operation per second), AverageLatency는 평균 지연 시간을 뜻한다.

4.3 비교 분석 결과

MongoDB와 MySQL의 데이터의 수를 500 개에서 10만 개까지 늘리며 실험한 결과는 Fig. 5와 같다.

Fig. 5의 (a) Runtime은 수치가 낮을수록 성능이 좋은 것을 나타내는데, 데이터 수가 적을 때는 비슷한 양상을 보이는 반면 데이터의 양이 커질수록 점점 격차가 벌어져 10만 개의 데이터에서 는 약 2.9배의 차이를 보였다. (b) Throughout는 초당 명령어 처리를 나타낸다. 수치가 클수록 성능이 좋다는 것을 보여준다. MongoDB 같은 경우에 10만 개의 데이터에서 실험 했을 때의 수치는 약간 감소하였지만, MySQL에 비해 초당 처리량이 크게 증가하였고, 가장 격차가 클 때는 약 4.5배의 차이를 보인다. 마지막으로 (c) AverageLatency는 평균 지연 시간으로 지연 시간이 낮을수록 성능의 우수함을 나타내는데, 둘 다 비슷한 추세로 감소를 하였지만, MongoDB에서의 평균 지연 시간이 더 짧으며, 10만 개의 데이터 기준 지연 시간은 MySQL과 약 2.9배의 차이를 보인다. Fig. 5의 결과로, NoSQL인 MongoDB가 데이터의 양이 커질수록 MySQL보다 더 우수한 성능을 보여준다.

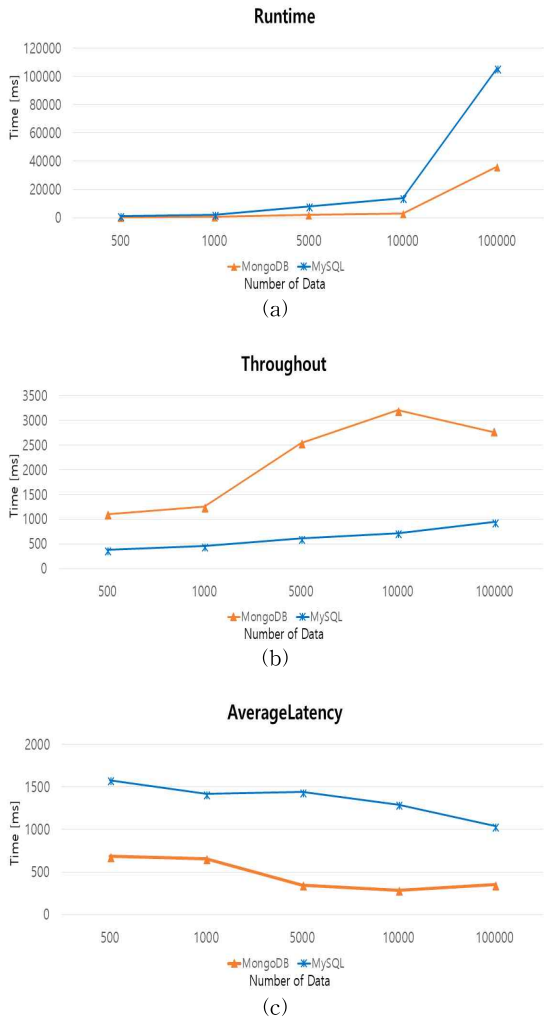

Fig. 5. Performances about MongoDB and MySQL in workload C. (a) Runtime, (b) Throughout, and (c) AverageLatency.

5. 결 론

최근 빅데이터에 대한 중요성이 커지고 ICT 융합 기술이 발전함에 따라 의료 분야 에도 이러한 기술들이 적용되는 추세이다. 그에 대한 의료 기관별 상호 정보 공유에 대한 중요성이 증가하고 있다. 하지만, 기존의 RDBMS만 가지고는 서로 다른 형태의 의료 데이터를 공유하기가 어려움을 겪고 있다.

본 논문에서는 RDBMS와 NoSQL를 이용한 HL7 FHIR 기반의 의료 데이터에 대해 비교 분석하였다. 데이터베이스는 RDBMS에서 MySQL을, NoSQL에서 사용하였고, 성능 분석은 YCSB라는 벤치마크 툴을 이용하여 진행했다. 실험에 대한 데이터로 환자 데이터를 RDBMS와 NoSQL의 스키마를 구성하고 랜덤 값으로 10만 개의 데이터를 입력했다. 그리고 Read-only 워크로드를 통해 기본 키 값에 의한 검색 쿼리를 진행 했고, 비교 분석한 결과 10만 개의 데이터를 기준으로 했을 때 실행 시간은 MySQL이 1만 개의 데이터부터 급격히 증가하며 MongoDB와 2.9배의 차이를 보였다. 초당 명령어 처리는 MongoDB가 5천 개의 데이터부터 급격한 증가를 보이며 1만 개의 데이터로 실험을 했을 때는 4.5배의 차이를 보이며 명령어 처리의 우수함을 보였으나, 10만 개부터는 약간의 감소를 보이며 명령어 처리 속도가 일정 수준을 넘어가지 않는 것을 보여줬다. 마지막으로 평균 지연 시간은 MongoDB이 더 짧은 지연 시간을

갖고 있지만, 두 데이터베이스 모두 비슷하게 감소하는 추세를 보이며 차이가 크게 벌어지지 않는 것을 보여줬다. 하지만, RDBMS에서는 NoSQL에 비해 보다 복잡한 쿼리를 수행하는 것에 대해 유리하다는 이점이 있다. 그렇기 때문에 상황에 맞춰 RDBMS와 NoSQL을 적합하게 사용할 필요성이 있다.

향후 진행할 연구 계획으로는 의료 데이터에 대해 NoSQL과 RDBMS 간의 효율적인 트랜잭션이 가능하게 하고, 효과적인 의료 데이터 리파지토리를 구축하는 연구를 진행할 예정이다.

REFERENCE

[1] S.H. Jung, J.C. Kim and C.B. Sim, "A Novel Data Prediction Model using Data Weights and Neural Network based on R for Meaning Analysis between Data," *Journal of Korea Multimedia Society*, Vol. 18, No. 4, pp. 524-532, 2015.

[2] H.S. Kim and B.H. Lee, "Interoperability between NoSQL and RDBMS via Auto-mapping Scheme in Distributed Parallel Processing Environment," *Journal of the Korea Institute of Information and Communication Engineering*, Vol. 21, No. 11, pp. 2067-2075, 2017.

[3] Database Ranking, https://db-engines.com/en/ranking (accessed Jul., 1, 2018).

[4] S. Shin, "Standardization of Atypical Healthcare Data," *The Journal of The Korean Institute of Communication Sciences,* Vol. 35, No. 2, pp. 58-64, 2018.

[5] N. Na, S. Yun, and S. Park, "Design and Implement of Mapping System for Effective Health Information Data Exchange in Multi-Platform Environment," *The Journal of Korea Institute of Information Technology*, Vol. 10, No. 12, pp. 143-150, 2012.

[6] About NoSQL, https://aws.amazon.com/nosql/ (accessed Jun., 30, 2018).

[7] H.J. Park, "A Study about Performance Evaluation of Various NoSQL Databases," *Journal of Korea Institute of Information, Electronics, and Communication Technology*, Vol. 9, No. 3, pp. 298-305, 2016.

[8] G.H. Kim, S. Cho, E. Lee, H.S. Kim, and H. Cho, "Health Level 7 Version 3 based Generating Clinical Document Architecture for Medication Administration System," *Journal of Korea Multimedia Society*, Vol. 11, No. 3, pp. 386-397, 2008.

[9] About YCSB, https://github.com/brianfrankcooper/YCSB/wiki (accessed Jun., 29, 2018).

[10] C. Park, J. Lee, and K. Kang, "Possibility of Reducing Power Consumption of Storage Server Based on I / O Pattern and Performance Evaluation Method," *Korea Information Processing Society Review*, Vol. 23, No. 4, pp. 36-42, 2016.

[11] Y. Abubakar, T.S. Adeyi, and I.G. Auta, "Performance Evaluation of NoSQL Systems Using YCSB in a Resource Austere Environment," *International Journal of Applied Information Systems*, Vol. 7, No. 8, pp. 23-27, 2014.

[12] B.F. Cooper, A. Silberstein, E. Tam, R. Ramakrishnan, and R. Sears, "Benchmarking Cloud Serving System with YCSB," *Proceedings of the 1st ACM Symposium on Cloud Computing*, pp. 143-154, 2010.

전 동 철

2013년~현재 가천대학교 컴퓨터공학과 학사과정
관심분야 : 데이터베이스, 헬스IT, 모바일

이 병 문

1988년 동국대 전자계산학과 (공학사)
1990년 서강대 전자계산학과 (공학석사)
2007년 인천대 컴퓨터공학과 (공학박사)

1990년~1997년 LG전자(구 LG정보통신) 중앙 연구소 네트워크 연구실 선임연구원
1998년~현재 가천대학교 IT대학 컴퓨터공학과 교수
관심분야 : 사물인터넷, 유헬스, 센서네트워크, 운영체제, 헬스케어

황 희 정

2000년 인하대학교 컴퓨터공학과 (공학석사)
2008년 인천대학교 컴퓨터공학과 (공학박사)
2000년~현재 가천대학교 IT대학 컴퓨터공학과

관심분야 : Software Engineering, u-Health, Big Data, Medical Informatics, Ubiquitous Computing