한국정보전자통신기술학회논문지(jkiiect)'23-10, Vol.16 No.5
논문 23-16-05-354 http://dx.doi.org/10.17661/jkiiect.2023.16.5.354

빅데이터 분석 시스템 구현을 위한 데이터 구조의 복잡성에 따른 MongoDB 환경 구성 연구

이협건*, 김영운, 이진우, 이승현

Study of MongoDB Architecture by Data Complexity for Big Data Analysis System

Hyeopgeon Lee*, Young-Woon Kim, Jin-Woo Lee, Seong Hyun Lee

요 약 빅데이터 분석 시스템들은 다양한 형태의 방대한 데이터를 저장 및 처리, 분석을 위해 MongoDB와 같은 NoSQL 데이터베이스를 적용한다. MongoDB는 환경 구성에 따라 분산 처리 및 데이터 복제를 통해 확장성과 빠른 데이터 처리 속도를 제공한다. 본 논문에서는 구현하는 빅데이터 분석 시스템에 적합한 MongoDB 환경 구성에 대해 연구한다. 성능 평가를 위한 환경은 크게 싱글 노드와 다중 노드 환경으로 구성하였으며, 다중 노드 환경은 데이터 노드의 수를 2대에서 3대까지 확장하여 각 환경별 성능을 측정하였다. 분석 결과, 3차원 이상의 복잡한 데이터 구조의 데이터 처리 속도는 싱글 노드 환경이 2개의 데이터 노드 환경에 비해 약 5.75% 빠르게 처리하지만,3개의 데이터 노드 환경은 싱글 노드 환경에 비해 약 25.15% 이상 빠르게 처리한다. 그러나 데이터 구조가 단순한 1차원 데이터 구조는 다중 노드 환경이 싱글 노드 환경에 비해 약 28.63% 빠르게 처리한다. 향후 본 연구를 기반으로 다양한 데이터 구조 및 방대한 양의 데이터를 통한 실질적인 검증이 필요하다.

Abstract Big data analysis systems apply NoSQL databases like MongoDB to store, process, and analyze diverse forms of large-scale data. MongoDB offers scalability and fast data processing speeds through distributed processing and data replication, depending on its configuration. This paper investigates the suitable MongoDB environment configurations for implementing big data analysis systems. For performance evaluation, we configured both single-node and multi-node environments. In the multi-node setup, we expanded the number of data nodes from two to three and measured the performance in each environment. According to the analysis, the processing speeds for complex data structures with three or more dimensions are approximately 5.75% faster in the single-node environment compared to an environment with two data nodes. However, a setting with three data nodes processes data about 25.15% faster than the single-node environment. On the other hand, for simple one-dimensional data structures, the multi-node environment processes data approximately 28.63% faster than the single-node environment. Further research is needed to practically validate these findings with diverse data structures and large volumes of data.

Key Words : NoSQL, MongoDB, BigData, DB Architecture

This work was supported by the Technology Innovation Program (20023244, Development of metaverse-based marketing service support system for supporting various advertising media and verifying effectiveness) funded By the Ministry of Trade, Industry & Energy(MOTIE, Korea).

*Dept. of Big Data, Seoul Gangseo Campus of Korea Polytechnics College
Lab, Veritas Connect
*Corresponding Author : Dept. of Big Data, Seoul Gangseo Campus of Korea Polytechnics College (hglee67@kopo.ac.kr)
Received October 16, 2023 Revised October 19, 2023 Accepted October 24, 2023

1. 서론

MongoDB[1]는 NO-SQL 기반 데이터베이스로 방대한 양의 데이터를 저장, 처리하는 함에 있어 특화된 데이터베이스이다. MongoDB[2]는 방대한 양의 데이터를 저장, 처리를 위해 다중 노드 환경을 제공한다. 다중 노드 환경은 싱글 노드 환경에 비해 분산 처리를 위한 네트워크, 시스템 상태 등 다양한 여건을 고려해야한다.

이에 본 논문에서는 빅데이터 분석 시스템 구현을 위한 데이터 구조의 복잡성에 따른 MongoDB 환경 구성 연구한다. 성능 평가를 위한 환경은 크게 싱글 노드와 다중 노드 환경으로 구성하였으며, 다중 노드 환경은 데이터 노드의 수를 2대에서 3대까지 확장하여 각 환경별 성능을 측정하였다.

분석 결과, 3차원 이상의 복잡한 데이터 구조의 데이터 처리 속도는 싱글 노드 환경이 데이터 노드의 수가 2개인 환경에 비해 약 5.75% 빠르게 처리한다. 그러나 데이터 노드의 수가 3개의 다중 노드 환경은 싱글 노드 환경에 비해 최대 25.15% 이상 빠르게 처리한다.

데이터 구조가 단순한 1차원 데이터 구조는 다중 노드 환경이 싱글 노드 환경에 비해 최대 약 28.63% 빠르게 처리한다. 또한 다중 노드 환경은 데이터의 양이 증가할수록 싱글 노드 환경에 비해 빠르게 데이터를 처리한다.

본 논문의 구성은 다음과 같다. 2장에서는 MongoDB에 대한 기본적인 이해와 환경 구성에 대해 살펴본다. 3장에서는 구현하는 빅데이터 분석 시스템에 대해 제시하고, 4장에서는 구현하는 빅데이터 분석 시스템에 적합한 MongoDB 환경을 분석하기 위해 데이터 처리시간을 측정한다. 마지막 5장에서는 결론 및 향후 연구 과제를 제시한다.

2. 관련 연구

본 장에서는 No-SQL 데이터베이스로 가장 보편적인 사용되는 MongoDB 및 MongoDB 환경 구성에 대해 살펴본다.

2.1 MongoDB 이해

MongoDB[3][4]는 빅데이터 저장 및 처리 기술로 개발된 No-SQL 데이터베이스이다. MongoDB 주요 특징은 데이터의 다양성과 확장성이다. 데이터의 다양성은 데이터 저장의 최소 단위를 문서(Document)로 정의하여 형태가 명확한 정형 데이터와 형태가 명확하지 않는 비정형 데이터를 저장한다. 확장성은 방대한 양의 데이터를 보다 빠르게 저장 및 읽기를 수행하기 위해 쉽게 데이터 노드를 추가하여 확장이 가능하다.

MongoDB[5][6]는 환경 구성에 방법에 따라 크게 싱글 노드 환경과 다중 노드 환경으로 구분된다. 싱글 노드 환경은 단일 서버에 데이터를 저장 및 처리하는 환경을 의미한다. 다중 노드 환경은 데이터를 저장하는 서버를 여러 개 구성하고 그 서버들에 데이터 저장을 분산하는 라우터로 구성된 환경을 의미한다.

2.2 MongoDB 싱글 노드 환경

MongoDB 싱글 노드 환경[7]은 데이터 저장 및 처리를 단일 서버로 수행하기에 다중 노드 환경에 비해 네트워크 구조 및 환경 구성이 간단하다. [그림 1]은 MongoDB의 싱글 노드 환경 구성의 예를 나타낸다.

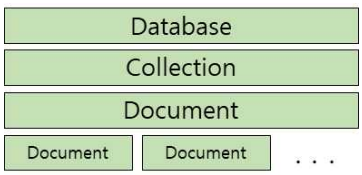

그림 1. MongoDB 싱글 노드 환경 구성의 예
Fig. 1. The example of single node

문서[8]는 MongoDB에 저장되는 데이터를 저장하는 단위로 JSON(Java Script Object Notation) 구조로 데이터를 저장한다. 문서의 구조는 여러 문서들을 하나의 문서에 포함이 가능하다. 컬렉션은 문서들을 그룹화한 것으로 관계형 데이터베이스의 테이블과 유사하다. [그림 2]는 문서와 컬렉션 구조의 예를 나타낸다.

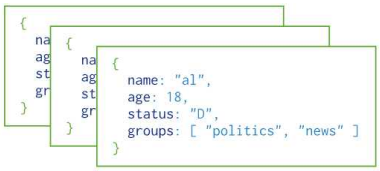

그림 2. 문서와 컬렉션 구조의 예
Fig. 2. The example of documents and collection

MongoDB는 방대한 양의 데이터를 빠르게 저장하기 위한 목적으로 개발된 데이터베이스로 관계형 데이터베이스와 달리 기본 키를 제공하지 않는다. 그러나 싱글 노드 환경은 Null 값이 허용되는 Unique Key를 제공하며, 관계형 데이터베이스의 Unique Key의 기능과 유사하다.

MongoDB는 데이터 수정 및 삭제를 위해 데이터를 구분할 수 있는 고유한 값으로 ObjectId 구조의 _Id 이름의 문서를 기본적으로 제공한다. ObjectId는 저장 시간(Time Stamp)을 활용한 확률적 모델을 기반으로 생성되는 16진수 값이다. 확률적 모델을 활용한 이유는 다중 노드 환경에서의 Sharding 기반 데이터 저장에서 각 노드별 저장되는 데이터들의 ObjectId 중복을 확인함에 있어 많은 리소스가 필요하기 때문이다.

2.3 MongoDB 다중 노드 환경

MongoDB 다중 노드 환경[9]은 발생되는 데이터를 저장하기 위해 여러 노드들을 구성한 환경이다. [그림 3]은 MongoDB의 다중 노드 환경 구성의 예를 나타낸다.

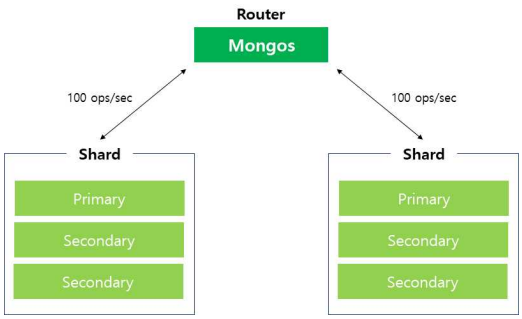

그림 3. MongoDB 다중 노드 환경 구성의 예
Fig. 3. The example of multi nodes

다중 노드 환경 구조는 크게 Mongos와 Shard들로 구성된다. Mongos는 저장되는 데이터를 분산 저장하기 위해 저장될 Shard를 결정하는 라우터 역할을 수행한다. 또한 Mongos는 Shard별 저장하는 문서마다 고유한 값을 제공하기 위해 확률적 모델 기반 ObjectId를 생성한다. 따라서 모든 문서들은 중복되지 않는 ObjectId 값을 가진다.

Shard는 저장되는 데이터를 분산 저장하기 위해 데이터가 저장되는 노드를 의미하며, Shard에 저장되는 과정은 Sharding으로 표현한다. Shard는 저장되는 데이터를 안정성을 높이기 위해 복제를 기능을 제공하여 동일한 데이터를 Primary, Secondary에 저장하는 데이터 다중화를 제공한다.

다중 노드 환경[10]은 데이터를 분산하여 저장하기에 고유 값 및 인덱스를 활용하기 위한 Unique Key를 적용이 불가능하다. 불가능한 이유는 노드별 독립적으로 저장되는 데이터들의 중복 여부를 확인이 불가능하기 때문이다.

2.4 MongoDB 노드 환경별 비교

본 절에서는 앞서 설명한 MongoDB의 싱글 노드 환경과 다중 노드 환경별 특징 비교한다. 〈표 1〉은 MongoDB 노드 환경에 따른 특징 비교를 나타낸다.

표 1. MongoDB 노드 환경에 따른 특징 비교
Table 1. Comparison by MongoDB architecture

Item	Single Node	Multi Nodes
Architecture	Stand Alone	Replica Server
		Config Server
		Shard Server
		Mongos
Distributed Processing	No	Yes
Data Replication	No	Yes
Routing	Not applicable	Yes
Networks Load	Not applicable	Yes
Index	Yes	Yes
Unique Key	No	Yes

앞서 비교한 결과에 따르면, MongoDB는 싱글 노드 환경이 다중 노드 환경에 비해 구성이 단순하여 분산 처리, 데이터 복제를 제공하지 않지만, 저장된 데이터에 대한 중복 여부를 확인이 가능하다.

또한 싱글 노드 환경은 분산 처리를 수행하지 않아 다중 노드 환경에 비해 라우팅, 네트워크 지연이 발생하지 않는다.

3. 구현하는 빅데이터 분석 시스템

구현하는 빅데이터 분석 시스템은 각 지역별 설치된 다수의 측정 장비로부터 실시간으로 측정된 데이터를 전달받아 저장하고, 매일 일괄 데이터 처리하여 분석한다.

[그림 4]는 구현하는 빅데이터 분석 시스템의 구성을 나타낸다.

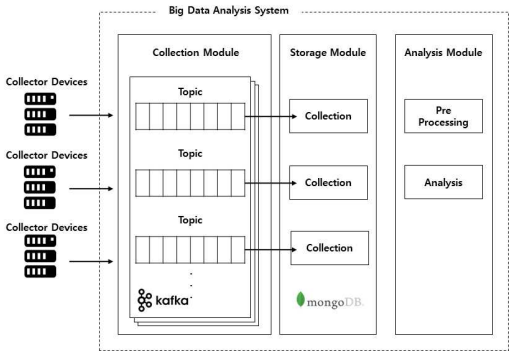

그림 4. 빅데이터 분석 시스템의 구성
Fig. 4. Architecture of big data analysis system

빅데이터 분석 시스템의 주요 구성은 빅데이터 수집 모듈, 빅데이터 저장 모듈과 빅데이터 분석 모듈로 구성된다.

빅데이터 수집 모듈은 측정 장비로부터 데이터를 수집하는 역할을 수행한다. 빅데이터 자장 모듈은 빅데이터 수집 모듈로부터 전달받은 데이터를 MongoDB에 저장하는 역할을 수행한다. 빅데이터 분석 모듈은 빅데이터 저장 모듈로부터 저장된 데이터를 분석하는 역할을 수행한다.

3.1 빅데이터 수집 모듈

[그림 5]는 빅데이터 수집 모듈을 나타낸다.

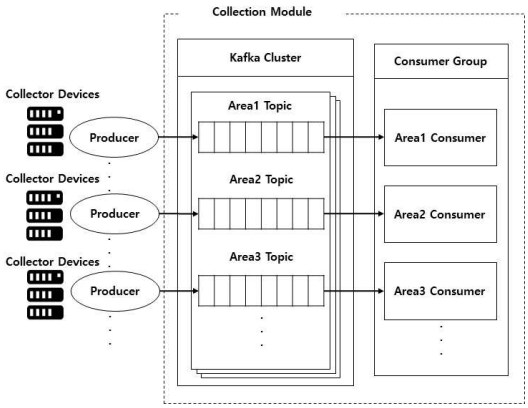

그림 5. 빅데이터 수집 모듈
Fig. 5. The big data collection module

데이터 수집 기술은 메시징 기반 오픈 소스인 카프카를 활용한다. 카프카를 사용한 이유는 다수의 측정 장비로부터 발생되는 데이터들이 주기적인 데이터 전달한다. 이로 인해 전송되는 데이터들은 트래픽 충돌로 패킷 폐기 현상이 발생할 수 있기에 메시징 기반 기술인 카프카를 적용한다. 카프카 토픽은 측정 장비가 설치된 위치에 따라 구분하여 적용한다.

3.2 빅데이터 저장 모듈

[그림 6]은 빅데이터 저장 모듈을 나타낸다.

빅데이터 저장 시스템은 카프카로부터 전달받은 데이터를 저장하며, 사용되는 데이터베이스는 MongoDB를 적용한다. MongoDB를 적용한 이유는 저장되는 데이터 구조가 배열 안에 배열들이 포함된 최대 3차원 JSON 구조이며, JSON 구조 저장에 적합하기 때문이다. 빅데이터 저장 시스템에 적용된 백앤드 프레임워크는 카프카로부터 저장되는 데이터를 저장하기 위해 Spring Boot와 Spring for Apache Kafka, Spring Data MongoDB 프레임워크를 사용한다. 데이터 모델링은 카프카의 토픽별 컬렉션을 생성하여 적용한다.

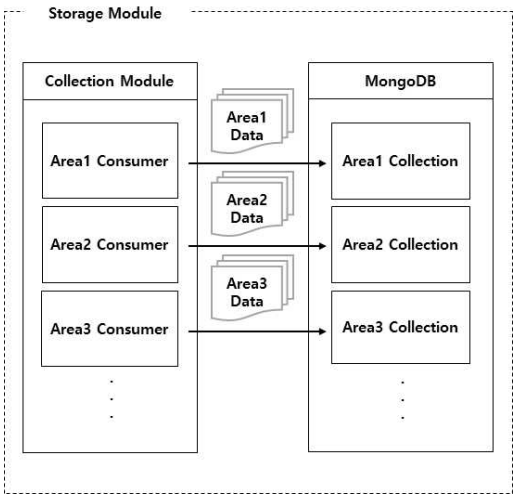

그림 6. 빅데이터 저장 모듈
Fig. 6. The big data storage module

3.3 빅데이터 분석 모듈

빅데이터 분석 시스템은 일괄 배치로 1일 2회 주기적으로 실행된다. 주요 수행 동작 과정은 컬렉션별 빅데이터 처리하는 전처리 단계와 정제된 결과를 각 분석 도메인 정의 및 도메인별 통합 분석하는 단계로 구분된다. 전처리 단계의 주요 기능은 컬렉션별 저장된 데이터들을 정제 및 분석에 용이하기 위해 데이터 집계 및 기초 통계를 수행하며, 그 결과를 도메인별 생성된 신규 컬렉션을 저장한다. 통합 분석 단계는 도메인별 기초 분석 결과를 기반으로 병합 및 최종 분석을 수행한다.

4. 성능 평가

본 장에서는 구현할 빅데이터 시스템에 적합한 MongoDB의 환경 구성을 평가를 목적으로 진행되며, 싱글 노드 및 다중 노드 환경 구성과 1차원, 3차원 데이터의 구조에 따라 데이터 처리 속도를 분석한다. 〈표 2〉는 성능 분석을 위한 실험 환경을 나타낸다.

표 2. 실험 환경
Table 2. Experiment environment

Item	Multi Nodes	
Server Specification	CPU	2Core
	RAM	2G
	SSD	29G
DB Architecture	Stand Alone	
	Distributed DB (Data Node : 2)	
	Distributed DB (Data Node : 3)	
Data Type	JSON	
MongoDB Collection	Total : 5	
	Name : Area1 ... Area5	
Record Count	Total : 200,000	
	Each Collection : 10,000	
Data Complexity	1 Dimension	
	3 Dimension	
Unique Key Index	Stan Alone : Create	
	Etc : Not Create	

MongoDB 환경 구성은 싱글 노드 환경, 데이터 노드 2개와 3개로 구성된 분산 환경으로 성능 평가를 수행한다. 싱글 노드 환경은 저장되는 데이터마다 Unique Key를 생성하여 index를 구성한다. 데이터 처리에 활용되는 데이터 형태는 카프카에서 보내주는 형태와 동일하게 JSON이다. 생성되는 컬렉션은 총 5개로 지역별 구분하며, 각 컬렉션마다 20,000개 레코드를 저장하여 총 100,000개 레코드를 저장한다. 성능 분석을 위한 데이터 복잡성은 1차원 구조와 3차원 구조로 정의한다.

4.1. 3차원 데이터 처리시간 분석

본 절에서는 각 컬렉션마다 3차원 데이터 구조로 저장된 데이터들을 MongoDB를 통해 처리한 시간을 각 환경 구성별 측정하고, 그 결과를 분석한다. 처리 기술 분석 방법은 MongoDB의 Aggregation을 활용하여 1개부터 5개까지 컬렉션들의 데이터들을 병합 후, 집계 연산을 수행한다. 데이터 노드의 수가 1개인 싱글 노드 환경은 *DN1*, 데이터 노드의 수가 2개인 다중 노드 환경은 *DN2*, 데이터 노드의 수가 2개인 다중 노드 환경은 *DN3*으로 정의하여 분석을 수행한다. 처리 시간의

증감 비율은 싱글 노드 환경을 기준으로 데이터 노드의 수가 2개인 경우와 데이터 노드의 수가 3개인 경우를 비교한다. 데이터 노드의 수가 2개인 경우 증감 비율은 *Rate A*로 정의하고, 데이터 노드의 수가 3개인 경우 증감 비율은 *Rate B*로 정의한다.

[그림 7]과 〈표 3〉은 환경 구성별 데이터 처리시간과 싱글 노드 기준으로 다중 노드 환경의 처리시간 증감에 대한 비율을 나타낸다.

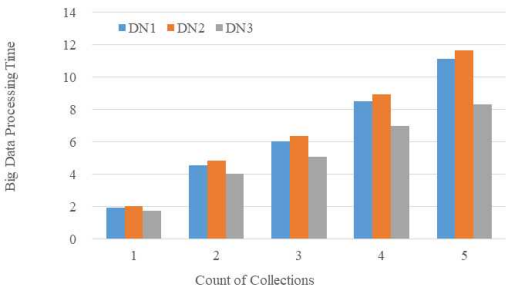

그림 7. 환경 구성 별 데이터 처리 시간

Fig. 7. The data processing time by DB Architecture

표 3. 데이터 처리시간 및 증감률

Table 3. The data processing time and increasing·decreasing rate

	Count of Collections				
	1	2	3	4	5
DN 1	1.912	4.556	6.012	8.512	11.134
DN 2	2.022	4.813	6.343	8.931	11.632
DN 3	1.721	4.012	5.092	6.965	8.334
Rate A	-5.75	-5.64	-5.51	-4.92	-4.47
Rate B	9.99	11.94	15.30	18.17	25.15

데이터 노드의 수가 1개인 싱글 노드 환경과 데이터 노드의 수가 2개인 다중 환경의 데이터 처리시간에 대한 분석 결과, 전반적으로 싱글 노드 환경은 데이터 노드의 수가 2개인 다중 노드 환경보다 약 5.26% 빠르게 데이터를 처리한다. 싱글 노드 환경은 컬렉션이 1개인 경우 1.912초이며, 데이터 노드의 수가 2개인 다중 노드 환경은 2.022초로 싱글 노드 환경이 약 5.75% 빠르게 처리한다. 싱글 노드 환경은 컬렉션이 5개인 경우 11.134초이며, 데이터 노드의 수가 2개인 다중 노드 환경은 11.632초로 싱글 노드 환경이 약 4.47% 빠르게 처리한다.

이러한 결과가 발생한 원인은 싱글 노드 환경은 Unique Key 생성을 통한 고유한 값에 대한 인덱스 생성 가능이 가능하기 때문이다. 다중 노드 환경은 고유한 값에 대한 보장을 할 수 없기에 Unique Key 생성을 통한 인덱스 생성이 불가능하다.

따라서 데이터 노드의 수가 2개인 다중 노드 환경은 데이터 노드별 데이터 처리 시간, 병합 시간, 분산을 통한 네트워크 지연 등 발생한다.

그러나 데이터 노드의 수가 2개인 다중 노드 환경은 컬렉션의 수를 증가 될수록 싱글 노드 환경과의 데이터 속도 차이는 감소한다.

싱글 노드 환경과 데이터 노드의 수가 3개인 다중 환경의 데이터 처리시간에 대한 분석 결과, 전반적으로 데이터 노드의 수가 3개인 다중 노드 환경은 싱글 노드 환경 보다 약 9.99% 빠르게 데이터를 처리한다. 또한 데이터 노드의 수가 3개인 다중 노드 환경은 컬렉션이 5개인 경우 싱글 노드 환경에 비해 약 25.15% 빠르게 처리한다.

이러한 결과가 발생한 원인은 MongoDB는 분산 환경에 적합한 데이터베이스로 노드의 수가 증가될수록 데이터 처리시간이 감소하기 때문이다.

4.2. 1차원 데이터 처리시간 분석

본 절에서는 각 컬렉션마다 1차원 데이터 구조로 저장된 데이터들을 MongoDB를 통해 처리한 시간을 각 환경 구성별 측정하고, 그 결과를 분석한다. 처리 기술 분석 방법은 앞서 분석한 3차원 데이터 처리시간 분석 방법과 동일하며, 환경에 대한 정의도 동일하다. [그림 8]과 〈표 4〉는 환경 구성별 데이터 처리시간과 싱글 노드 기준으로 다중 노드 환경의 처리시간 증감에 대한 비율을 나타낸다.

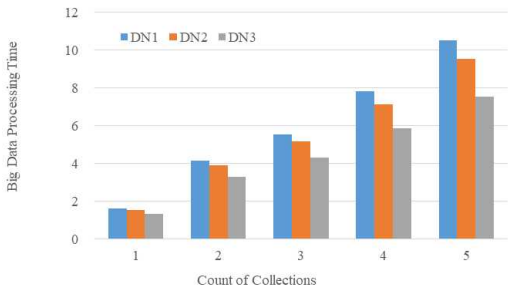

그림 8. 환경 구성 별 데이터 처리 시간
Fig. 8. The data processing time by DB Architecture

표 4. 데이터 처리시간 및 증감률
Table 4. The data processing time and increasing·decreasing rate

	Count of Collections				
	1	2	3	4	5
DN 1	1.612	4.156	5.512	7.812	10.534
DN 2	1.522	3.896	5.143	7.131	9.531
DN 3	1.321	3.289	4.292	5.865	7.518
Rate	18.05	20.86	22.13	24.92	28.63

데이터 노드의 수가 1개인 싱글 노드 환경과 다중 노드 환경의 데이터 처리시간에 대한 분석 결과, 데이터 구조가 단순한 1차원 데이터 구조는 데이터 노드의 수가 증가될수록 빠르게 데이터를 처리한다. 컬렉션의 수가 5개인 경우, 데이터 노드의 수가 3개인 다중 노드 환경은 7.518로 싱글 노드 환경에 비해 약 28.63% 빠르게 데이터를 처리한다. 또한 컬렉션이 1개인 경우보다 5배 많은 컬렉션의 경우, 처리시간은 다중 노드 환경에서 더 빠른 데이터 처리 속도를 보인다.

5. 결론

본 논문에서는 빅데이터 분석 시스템 구현을 위한 데이터 구조의 복잡성에 따른 MongoDB 환경 구성 연구한다. 적합한 MongoDB 환경 구성을 분석 방법은 데이터 노드의 수가 1개인 싱글 노드 환경부터 데이터 노드의 수가 3개인 다중 노드 환경까지 데이터 구조에 따라 데이터 처리시간을 측정한다.

분석 결과, 3차원 이상의 복잡한 데이터 구조의 데이터 처리 속도는 싱글 노드 환경이 2개의 데이터 노드 환경에 비해 약 5.12% 빠르게 처리하지만,3개의 데이터 노드 환경은 싱글 노드 환경에 비해 약 41.91% 이상 빠르게 처리한다. 그러나 데이터 구조가 단순한 1차원 데이터 구조는 다중 노드 환경이 싱글 노드 환경에 비해 약 61.15% 빠르게 처리한다. 향후 본 연구를 기반으로 다양한 데이터 구조 및 방대한 양의 데이터를 통한 실질적인 검증이 필요하다.

REFERENCES

[1] H. G. Lee, Y. W. Kim, and K. Y. Kim, "Study of Efficient Algorithm for Deduplication of Complex Structure," Journal of Korea Institute of Information, Electronics, and Communication Technology, vol. 14, no. 1, pp. 29-36, 2021.

[2] 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.

[3] A. Sharma, P. Kaur, "A Multitenant Data Store Using a Column Based NoSQL Database," 2019 Twelfth International Conference on Contemporary Computing (IC3), 2019

[4] H. G. Lee, Y. W. Kim, and K. Y. Kim, "Study of In-Memory based Hybrid Big Data Processing Scheme for Improve the Big Data Processing Rate," Journal of Korea Institute of Information, Electronics, and Communication Technology, vol. 12, no. 2, pp. 127-134, 2019.

[5] H. G. Lee, Y. W. Kim, K. Y. Kim, and J. S. Choi, "Design of GlusterFS Based Big Data Distributed Processing System in Smart Factory," Journal of Korea Institute of Information, Electronics, and Communication Technology, vol. 11, no. 1, pp. 70-75, 2018.

[6] T. Capris, P. Melo, N. M. Garcia, I. M. Pires and E. Zdravevski, "Comparison of SQL and

NoSQL databases with different workloads: MongoDB vs MySQL evaluation," International Conference on Data Analytics for Business and Industry (ICDABI), 2022

[7] D. Yedilkhan, A. Mukasheva, D. Bissengaliyeva and Y. Suynullayev, "Performance Analysis of Scaling NoSQL vs SQL: A Comparative Study of MongoDB, Cassandra, and PostgreSQL," IEEE International Conference on Smart Information Systems and Technologies (SIST), 2023.

[8] A. A. Alhueaimel and M. B. Alotaibi, "An Evaluation of NoSQL Databases Using the Analytical Hierarchy Process (AHP)," 2023 International Conference on Information Management (ICIM), 2023.

[9] N. B. Seghier and O. Kazar, "Performance Benchmarking and Comparison of NoSQL Databases: Redis vs MongoDB vs Cassandra Using YCSB Tool," Journal of Korea Institute of Information, Electronics, and Communication Technology, vol. 11, no. 1, pp. 70-75, 2018.

[10] A. Y. Muhammad and F. N. Azizah, "Conversion of Entity-Relationship Model to NoSQL Document-Oriented Database Logical Model Using Workload Information and Entity Update Frequency," 2022 9th International Conference on Advanced Informatics: Concepts, Theory and Applications (ICAICTA), 2022.

저자약력

이 협 건 (Hyeopgeon Lee) **[종신회원]**

- 2011.03 - 2015.08, 숭실대학교 일반대학원 컴퓨터학과 공학박사
- 2015.12 - 현재, 한국폴리텍대학 서울강서 캠퍼스 빅데이터과 교수

〈관심분야〉 빅데이터처리 프로그래밍, 자연어처리, MSA아키텍처, 클라우드컴퓨팅

김 영 운 (Young-Woon Kim) **[종신회원]**

- 2004.09 - 2008.08, 숭실대학교 일반대학원 컴퓨터학과 공학박사
- 2015.12 - 현재, 한국폴리텍대학 서울강서 캠퍼스 빅데이터과 교수

〈관심분야〉 빅데이터, 인공지능프로그래밍, 컴퓨터비전, 클라우드컴퓨팅

이 진 우 (Jin-Woo Lee) **[정회원]**

- 1987.03 - 1993.08, 숭실대학교 물리학과 학사
- 2000.10 - 2014.08, 코리아닷컴커뮤니케이션 본부장
- 2021.08 - 현재, ㈜베라타스커넥트 연구소장

〈관심분야〉 빅데이터, 신호처리

이 승 현 (Hyun-Seong Lee) **[정회원]**

- 1992.03 - 1998.12, 고려대학교 컴퓨터학과 학사
- 1997.12 - 1999.12, KT하이텔
- 2003.06 - 현재, ㈜베라타스커넥트 부사장

〈관심분야〉 빅데이터, 마케팅