The Journal of the Convergence on Culture Technology (JCCT)
Vol. 9, No. 6, pp.911-917, November 30, 2023. pISSN 2384-0358, eISSN 2384-0366

http://dx.doi.org/10.17703/JCCT.2023.9.6.911

JCCT 2023-11-109

시계열 데이터 최적화 기법을 활용한 Key-value store의 엣지 기반 데이터 수집 시스템 평가

Evaluation of Edge-Based Data Collection System for Key-Value Store Utilizing Time-Series Data Optimization Techniques

조우진*, 이형아**, 구재회***

Woojin Cho*, Hyung-ah Lee, Jae-hoi Gu*****

요 약 오늘날 우리는 전쟁과 기후 위기 등에 의해 에너지 위기 요소를 안고 있게 되었다. 이러한 에너지 위기를 대비하기 위해 많은 연구자가 에너지 관리 시스템이라는 에너지 절감 및 관리와 같은 에너지 모니터링 및 에너지 절감에 대한 시스템에 대한 연구를 지속하고, 이에 발맞춰 국가에서도 에너지 다소비 사업장에서 이를 의무화하고 있다.
이러한 공장은 공간과 에너지적 한계가 존재하여 이를 개선 하고자 낮은 성능의 임베디드 디바이스로 데이터 수집 시스템을 구동하는 방안에 대해 연구를 진행한다. 이때 임베디드 디바이스에서 기존의 데이터베이스가 아닌 Key-value store인 RocksDB의 최적화 버전이 시계열 데이터에 우수한 성능을 보임을 평가를 통해 보인다. 또한 이를 평가하기 위한 범용 데이터베이스 평가 도구를 통해 이종 데이터베이스와 평가를 진행한다. 그 결과 낮은 성능의 디바이스에서 타 데이터베이스 대비 11배 짧은 소요 시간을 기록하는 것을 볼 수 있었다.

주요어 : 데이터베이스, 벤치마크, NoSQL, 시계열 데이터, 최적화, 평가

Abstract In today's world, we find ourselves facing energy crises due to factors such as war and climate crises. To prepare for these energy crises, many researchers continue to study systems related to energy monitoring and conservation, such as energy management systems, energy monitoring, and energy conservation. In line with these efforts, nations are making it mandatory for energy-consuming facilities to implement these systems.
However, these facilities, limited by space and energy constraints, are exploring ways to improve. This research explores the operation of a data collection system using low-performance embedded devices. In this context, it proves that an optimized version of RocksDB, a Key-Value store, outperforms traditional databases when it comes to time-series data. Furthermore, a comprehensive database evaluation tool was employed to assess various databases, including optimized RocksDB and regular RocksDB. In addition, heterogeneous databases and evaluations are conducted using a UD Benchmark tool to evaluate them. As a result, we were able to see that on devices with low performance, the time required was up to 11 times shorter than that of other databases.

Key words : Database, Benchmark, NoSQL, Time series Data, Optimization, Evaluation

*정회원, 고등기술연구원 연구원 (제1 저자)
**정회원, 고등기술연구원 연구원 (제2 저자)
***정회원, 고등기술연구원 연구위원 (교신저자)
접수일: 2023년 10월 9일, 수정완료일: 2023년 10월 24일
게재확정일: 2023년 11월 5일

Received: October 9, 2023 / Revised: October 24, 2023
Accepted: November 5, 2023
***Corresponding Author: jaehoi@iae.re.kr
Dept. of Energy Environment IT Convergence Group,
Institute for Advanced Engineering, Korea

Ⅰ. 서 론

전 세계적으로 전쟁과 에너지 비용 증가, 기후 위기와 같은 위기와 문제에 의해 에너지 안보와 에너지 효율화 및 절감에 대해 많은 관심을 지니게 되었다. 에너지 위기에 대비하기 위해 공학자들은 기존에도 많은 노력을 기울였다. 그 중 대표적인 에너지 절감 기술 중 하나가 에너지 관리 시스템(Energy Management System)이라는 기술이다. 에너지 관리 시스템은 기존에 전통적으로 운용하였던 공장, 집, 빌딩 등의 운용 방식을, IoT 기기의 발전에 의한 센서와 같은 장비를 통해 데이터를 수집하여 데이터를 기반으로 운용하여 에너지 사용량 정보를 확인을 넘어서 에너지 절감 기술을 적용할 수 있는 기술이다[1][2]. 에너지 관리 시스템을 적용할 수 있는 수많은 분야 중 가장 큰 규모를 지닌 공장 에너지 관리 시스템(Factory Energy Management System)은 대한민국의 제 3차 에너지 기본 계획을 통해 2025년 부터 10만 TOE(Ton of Oil Equivalent)이상의 에너지 다소비 사업장에서는 공장 에너지 관리 시스템을 의무로 적용해야 하는 정책이 수립되었다. 하지만 이에서 그치지 않고, 10만 TOE 이하의 공장에서도 공장 에너지 관리 시스템에 대한 보급을 진행하고 있다. 소형 공장 대다수는 환경 및 특성상 기존의 서버 시스템을 운용하기에 어려움이 있다. 시스템을 위한 서버실의 환경 구성 자체에 어려움이 있으며 서버 자체의 높은 전력 소모는 큰 걸림돌이다. 서버뿐만이 아닌 서버 공간의 유지관리를 위한 지속적 에너지 소비, 공간 할애와 같은 문제점도 지닌다. 이러한 문제를 해결할 수 있는 것이 저전력의 적은 공간을 사용하는 임베디드 디바이스를 활용한 경량 시스템 구축이다.

하지만 이와 같은 임베디드 시스템은 기존의 서버에 비해 낮은 성능을 지니기 때문에 수많은 기법을 활용하는 모든 에너지 관리 시스템을 한데 구축하기는 어려움이 따른다. 따라서 본 논문에서는 에너지 관리 시스템의 핵심 시스템 중 하나인 데이터를 수집하고 저장하는 데이터 수집 시스템의 임베디드 시스템에서의 연구를 진행한다. 또한 기존의 관계형 데이터베이스는 데이터 쓰기에 단점을 보이고, 시계열 데이터베이스들은 높은 쓰기 성능을 보장하기 위해 높은 자원 요구량을 지닌다. 이러한 단점을 극복하기 위해 Key-value store인 RocksDB를 활용하여 시계열 데이터 삽입에 최적으로 동작할 수 있도록 Parameter 및 압축 알고리즘을 선택하여 타 데이터베이스와의 비교를 통해 낮은 성능을 지닌 디바이스에서 데이터 수집 시스템을 운용하는 것에는 Key-value store가 가장 적합함을 보인다[3]. 마지막으로 완전히 이종인 데이터베이스들인 관계형 데이터베이스, 시계열 데이터베이스, Key-value store를 모두 지원 가능한 범용 평가 도구를 개발하여 한가지 지표로 세 가지 데이터베이스를 비교하여 최적화한 RocksDB의 임베디드 디바이스 적합성에 대해 평가를 진행한다.

따라서 본 논문에서는 RocksDB의 시계열 데이터베이스로써의 활용 여지에 대해 알아보고, 범용 평가 도구를 개발하여 다양한 데이터베이스에서 지원 가능한 벤치마크를 연구한다. 또한 임베디드 디바이스의 데이터 수집 시스템으로써의 활용 및 신뢰도를 보장할 수 있도록 낮은 성능의 스토리지인 MicroSD에서 모든 데이터베이스를 각각 평가하여 실증 공장에 적용이 가능한지에 대한 여부를 확인한다.

Ⅱ. 배 경

1. 관계형 데이터베이스

관계형 데이터베이스는 전통적인 데이터베이스로써 데이터는 관계에 의해 저장되어 각각의 데이터 구조의 관련성을 쉽게 파악하고 이해할 수 있도록 설계된 데이터베이스이다. 이러한 관계형 데이터베이스는 ACID 지원, 편의성, 높은 가시성 등을 지니고 있으나, 확장성 및 복잡한 데이터, 비정형 데이터에는 적합하지 않다는 단점을 지니고 있다.

1) MySQL

MySQL은 현재 전세계에서 가장 많이 사용되고 있는 오픈소스 관계형 데이터베이스이다[4]. 다양한 관련 생태계와 SQL 지원 및 높은 성능을 지니고 있다. B-tree 기반으로 만들어진 InnoDB라는 스토리지 엔진을 기본 스토리지 엔진으로 사용하여 Key를 인덱싱하여 높은 조회 성능을 지니는 데이터베이스이다[5].

2. 시계열 데이터베이스

시계열 데이터베이스는 시계열 데이터 처리에 최적화된 데이터베이스를 의미한다. 이러한 시계열 데이터베이스의 경우 시간과 함께 데이터를 저장하는 것을 주

목적으로 주로 Insert와 Delete 연산 위주의 최적화가 진행되어있다.

1) InfluxDB

InfluxDB는 높은 쓰기 처리와 쿼리 부하 등을 처리하기 위해 Go 언어로 개발된 시계열 데이터베이스 중 하나로써 수많은 오픈소스 시계열 데이터베이스 중 가장 널리 사용되는 데이터베이스이다[6]. 높은 쓰기 성능과 더불어 API 지원 및 InfluxQL과 같은 쿼리들의 지원을 통해 쉬운 사용을 지원한다. 하지만 TICK stack, 기본적인 DBMS의 경량화 문제에 있어서 저사양 디바이스에서 속도 보장이 어려운 문제점이 존재한다[7].

3. Key-value store

Key-value store란 NoSQL 데이터베이스의 일종으로 Key와 Value를 한 쌍으로 저장하는 데이터베이스이다. 이러한 Key-value store는 해시 등 다양한 기법으로 키를 처리할 수 있다. 대다수의 Key-value store는 요구사항이 매우 낮아 자원을 보다 적게 필요로 하는 경우가 많아 보편적으로 여타 데이터베이스에 비해 높은 성능을 얻을 수 있다.

이러한 Key-value store를 사용한 대표적인 데이터베이스는 RocksDB, LevelDB, Big Table 등이 있고, 본 연구에서는 RocksDB를 선택하여 기존의 많은 자원을 요구하는 데이터베이스와의 성능 비교에 대한 연구를 진행한다.

1) LSM-tree (Log Structured Merge Tree)

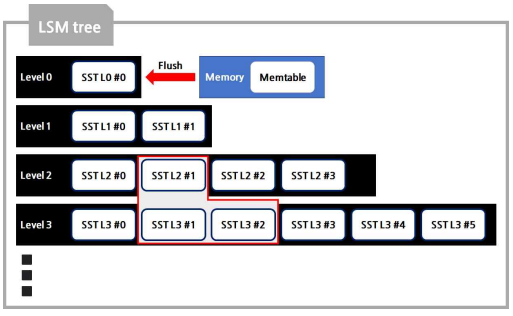

그림 1. LSM-tree 구조
Figure 1. LSM-tree Architecture

LSM-tree는 주로 쓰기 최적화를 위해 Key-value store에서 채택한다[8]. LSM-tree는 Rocksdb, LevelDB, Cassandra, 등의 수많은 데이터베이스에서 채택하고 있다. LSM-tree는 쓰기 최적화를 위해 그림 1과 같은 계층적 구조를 지닌다. 메모리에 쓰기 버퍼라는 memtable을 지니고 memtable이 가득 차면 Immutable memtable이 되어 파일시스템으로 플러시 되고, 이때 플러시 된 파일은 SST(Sorted String Table)라는 정렬된 스트링 파일로 저장 된다. 각 계층은 레벨이라 부르는데, 레벨이 한계에 도달할 때 다음 레벨의 SST 파일의 키 범위가 겹치는 SST 파일과 병합과 정렬을 하는 Compaction이라는 과정을 통해 트리를 유지한다. 이 때 각 레벨의 크기가 증가하는 비율을 Size factor, memory에 존재하는 memtable의 개수를 write buffer number, 크기를 write buffer size라고 한다.

2) RocksDB

RocksDB는 기존 Google의 LevelDB를 기반으로 Flash drive 최적화 및 여러 기능들을 추가하여 개선한 Key-value store이다. RocksDB는 LSM-tree(Log Structured Merge Tree)를 기반으로 제작되었는데, 이를 통해 모든 쓰기를 순차 쓰기로 변환하여 쓰기 작업에 큰 강점을 지니고, C++ Library 형태로 제공되는 Key-value store이다.

4. 데이터 수집 시스템

데이터 수집 시스템이란 기존의 데이터 수집 시스템이었던 DAQ(Data Acquisition)의 확장된 개념의 시스템이다[9]. 현재는 이의 개념이 확장되어 센서, PLC, HMI만이 아닌 데이터베이스, 데이터 전처리 등 광범위한 개념을 통틀어 이야기 한다. 데이터 수집 시스템은 데이터를 수집하는 수많은 응용에서 사용되는데, 본 논문에서는 공장 에너지 관리 시스템의 데이터 수집 시스템에 가장 적합한 데이터베이스를 선택하기 위한 벤치마크 및 데이터베이스 최적화를 진행한다.

III. RocksDB 시계열 최적화 기법

1. RocksDB

RocksDB의 경우 높은 쓰기 성능과 높은 압축율, 낮은 자원 요구와 같은 이점에 의해 많은 데이터베이스와 응용의 Back-end 스토리지 엔진으로 사용되고 있다. RocksDB는 다양한 응용에서 활용할 수 있지만, 특성에 의해 “순차 데이터”인 시계열 데이터에서도 높은 활용

도를 지닌다. 또한 RocksDB는 MySQL과 같은 관계형 데이터베이스나 InfluxDB와 같은 시계열 데이터베이스에 비해 낮은 자원을 요구하기 때문에 일반적인 디바이스에서 구동하였을 때도 설정에 따라 높은 성능을 보장할 수 있고, 낮은 성능의 디바이스에서는 타 데이터베이스에 비해 나은 성능을 지닐 수 있다. 또한 기존의 관계형 데이터베이스에 비해 높은 자유도를 지녔기 때문에 이외의 여러 방면에서도 추가적인 이득을 얻을 수 있다. 따라서 RocksDB를 본 논문에서는 에너지 관리 시스템에 최적 데이터베이스라 설정하고 시계열 데이터에 적합한 최적화를 진행한다.

1) 압축 알고리즘

공장 에너지 관리 시스템의 데이터 수집을 위한 임베디드 디바이스의 데이터베이스를 운용함에 있어 고려해야 할 부분 중 “데이터 적재”의 측면에서 압축 알고리즘은 매우 중요한 요소이다.

RocksDB는 여러 가지 압축 알고리즘을 지원하는데 모든 압축 알고리즘이 시계열 데이터에 적합한 것은 아니기에 우리가 실제로 사용할 시계열 데이터에 최적화된 압축 알고리즘 평가를 통해 선택한다.

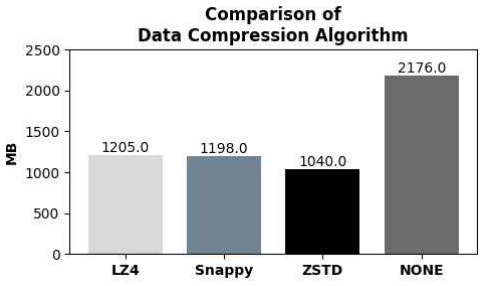

그림 2. 압축 알고리즘 비교
Figure 2. Comparison of Compression Algorithms

표 1. 압축 알고리즘 상세
Table 1. Compression Algorithms Spec.

	Compression	Decompression
LZ4	740 MB/s	4600 MB/s
Snappy	550 MB/s	1750 MB/s
ZSTD	530 MB/s	1700 MB/s

표 1을 보면 압축 속도 면에서는 LZ4가 가장 좋은 성능을 보이는 것을 알 수 있다[LZ4]. 압축 알고리즘의 선택에 있어서 속도 또한 중요한 요소이지만, 압축률이 가장 중요한 지표이기 때문에 공장 에너지 관리 시스템의 순차 데이터를 모사하여 2 GB 내외의 데이터를 생성하여 압축률을 확인하였다. 그 결과 그림 2와 같이 실제 임베디드 디바이스에서 압축 알고리즘을 테스트한 결과 ZSTD가 LZ4와 Snappy보다 대략 20%가량 높은 압축을 보이는 것을 확인 할 수 있었다[Snappy, ZSTD]. 임베디드 디바이스에서 가장 흔히 사용되는 스토리지인 MicroSD 같은 경우에는 UHS-3의 기준에서 최소 30MB/s의 속도를 보장하고 높은 퍼포먼스를 보이는 카드가 100MB/s를 지원한다. 또, SATA3로 SSD를 구성한다 가정하여도 최대 500-600 MB/s 가량의 속도를 보인다. 이는 LZ4와 같은 고속 압축 알고리즘의 대역폭을 제대로 활용할 수 없다는 것을 의미하기 때문에 압축률이 가장 높은 ZSTD를 최적화 된 RocksDB의 압축 알고리즘으로 선택한다.

2) LSM-tree parameter

LSM-tree는 파라미터에 의해 효율성과 성능에 큰 영향을 미친다. LSM-tree의 다양한 파라미터 중 삽입 성능에 가장 큰 영향을 줄 것으로 판단한 네 가지의 파라미터를 선택하여 조정하는 것을 목표로 하였다. 이 중 LSM-tree의 레벨 증가는 보편적인 상황에서 레벨 분산 저장 및 Compaction 범위의 최소화로 인해 삽입 성능에서 이점을 갖는다. 하지만, 시계열 데이터와 같이 Key가 겹치지 않는 경우는 Compaction이 발생하지 않기 때문에 일반적인 삽입과 다른 경향을 보인다. 따라서 이 경우 데이터의 범위 중복이 발생하지 않아 Compaction 연산 자체가 발생하지 않게 되고, 그에 따라 Compaction 오버헤드가 없고, 높은 레벨을 지녔을 때의 이점은 멀티 프로세스의 병렬 작업 시에만 해당된다. 하지만 컴퓨팅 파워가 대체로 보다 낮은 임베디드 디바이스에서는 기존의 서버와 달리 병렬성을 확보하기 어렵다. 또한, 레벨 관리에 대한 오버헤드가 오히려 커질 가능성 또한 존재한다. 따라서 레벨을 오히려 낮춰 관리 오버헤드를 줄이고, 적은 메모리에 의해 Write Buffer Size를 128MB로 설정한다. Write Buffer Number의 경우 병렬성을 활용하기 위하여 2개의 Write Buffer로 설정한다. 본 논문에서는 표 2와 같이 RocksDB 파라미터를 설정하여 시계열 데이터에 최적화된 RocksDB를 타 데이터베이스와 비교 평가한다.

표 2. RocksDB 최적화 파라미터
Table 2. RocksDB Optimization Parameters

Parameters	Value
Write Buffer Number	2
Write Buffer Size	128 MB
Level	1
Size Factor	150

Ⅳ. 범용 데이터베이스 성능 평가 도구

MySQL과 InfluxDB, RocksDB는 모두 이종 데이터베이스이므로 각기 다른 방식의 데이터 입력 방식을 채택한다.

특히 MySQL은 관계형 데이터베이스이고, NoSQL인 InfluxDB와 RocksDB는 특성도 사용법도 매우 달라 한 가지 지표로 모두 평가하는 것은 매우 도전적인 연구이다. 임베디드 디바이스의 데이터 수집 시스템 신뢰도 평가 등의 디바이스에서의 평가를 진행하기 위해서는 공평한 비교를 위해 이종 데이터베이스를 비교할 수 있도록 통합 벤치마크를 제작하여 모든 데이터베이스를 비교할 수 있도록 하는 작업이 필요하다. 따라서 본 논문에서는 이종 데이터베이스에서 범용으로 사용이 가능한 성능 평가 도구인 Universal Database Benchmark에 대한 연구를 진행한다..

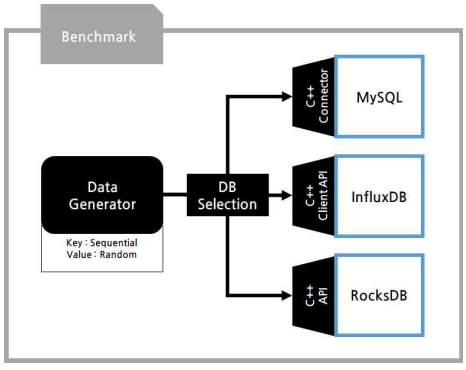

그림 3. UD Benchmark 구조
Figure 3. UD Benchmark Architecture

UD Benchmark의 구조는 그림 3과 같으며 InfluxDB는 influxdb-cpp로 제공되는 API를 통해 데이터를 전송하여 저장하는 방식을 통해 제작하였다. MySQL의 경우 모두 C++ 라이브러리를 지원하여 해당 라이브러리를 활용하여 벤치마크를 개발한다. RocksDB는 C++ API를 지원하기 때문에 해당하는 라이브러리를 활용하여 데이터를 입력받을 수 있다. 하지만, RocksDB는 관계형 데이터베이스인 MySQL과 필드를 가지는 InfluxDB와는 달리 Key와 Value로만 데이터를 저장할 수 있다. 본 논문에서는 단순한 평가 도구로써의 벤치마크가 아닌 데이터 수집 시스템의 실증 적용을 위한 평가 도구 개발을 목표로 하므로 데이터 저장과 조회가 가능하도록 관련된 Convertor를 제작한다.

InfluxDB와 MySQL의 경우 센서 개수에 따라 field N이라는 Column을 할당하였다. 하지만 RocksDB의 경우 Key-value store로써 SST File로 필드 단위나 특정 단위가 아닌 모든 값을 통으로 저장하게 되어 구분자가 필요하게 된다. 따라서 이 문제를 해결하기 위해 본 연구에서는 16진수로 아스키 코드 값이 비어있는 0x80을 선택하여 각 필드 값 사이에 0x80이라는 구분자를 추가하고 각 값에 해당하는 필드명은 첫 번째 키를 활용하여 값에 해당하는 필드명을 저장하였다. 따라서 본 논문에서는 RocksDB의 값을 조회하는 상황을 위해서 0x80로 구분하여 조회 가능하게 개발하여 각 필드에 해당하는 데이터를 쉬이 읽을 수 있도록 구성하였다. 추가적으로 시계열 데이터 첫 번째 Key에 필드의 정보를 저장하여 행 당 모든 데이터를 저장하는 디스크 공간 낭비, 관리 부하 등을 줄였다. 각 필드의 이름은 기존과 같이 fieldN을 사용하여 명명하였다.

평가 도구의 데이터는 키의 경우 1부터 N까지 수행하도록 하였고, 값의 경우는 Sensor 개수에 따라 소수점 5번째 자리까지 표출되도록 Random 데이터를 삽입한다.

위와 같은 방식으로 제작한 UD Benchmark로 이종 데이터베이스인 RDBMS와 TSDB, KV Store의 임베디드 디바이스에서의 성능 평가를 진행한다.

Ⅴ. 평 가

1. Evaluation Setup

1) Embedded device spec.

본 연구에서 사용한 임베디드 디바이스는 가장 흔히 사용되는 Raspberry Pi 4 model B 메모리 4GB 모델을 사용하였다[10]. 운영체제는 Ubuntu 22.04, 스토리지는 Samsung의 MicroSD Evo plus 256GB로 선택하여 평가한다.

2) Evaluation data set

본 연구에서는 UD Benchmark를 통해 총 1000만개의 순차 데이터를 생성하여 평가를 진행하였다. 키는 1부터 1000만까지, 값은 소수점이 있는 수로 측정하는 센서를 모사하여 100개의 센서를 소수점 5자리까지 저장하도록 설정하여 평가 데이터 셋으로 사용한다.

2. Evaluation

본 논문에서는 임베디드 디바이스에서 데이터 수집 시스템을 위한 데이터베이스 평가를 진행한다. MySQL, InfluxDB 두 개의 각 데이터베이스 군의 대표적인 데이터베이스 관리 시스템과 파라미터 최적화를 진행한 RocksDB 세 가지 데이터베이스를 비교 평가한다. 평가 지표는 소요 시간과 데이터베이스 용량, CPU 사용률로 평가를 진행한다.

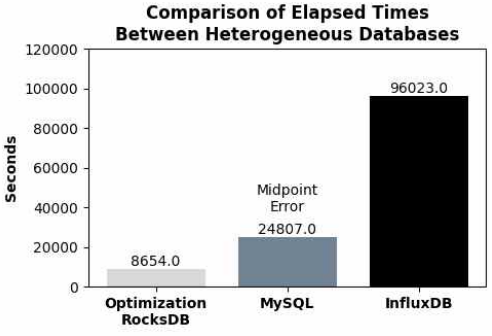

그림 4. 이종 데이터베이스 간의 소요시간 비교
Figure 4. Comparison of Elapsed Times Between Heterogeneous Databases

그림 4는 1000만개의 데이터 입력 시 데이터베이스 간 소요 시간의 비교 결과 그래프이다. 우선 MySQL의 경우 168만 행 가량을 삽입하였을 때 테이블 오류가 발생하여 삽입 오류가 발생하였다. 이유는 B-tree의 inplace update가 지속적 삽입으로 인해 발생하여 지속해서 높은 쓰기 부하가 발생한다. 그에 따라 높은 쓰기 부하로 인한 병목과 트랜잭션 문제로 인해 프로그램 문제를 발생시켜 더 이상의 평가가 불가능한 것을 볼 수 있다. 그를 제외한 InfluxDB와 시계열 데이터에 최적화한 RocksDB의 경우 자원 활용을 가장 효율적으로 한 시계열 데이터에 최적화된 RocksDB가 최대 11배 가량 빠른 처리를 할 수 있는 것으로 나타났다. InfluxDB의 경우 삽입 시 Bulk insert 관련 명령에 대한 오버헤드와 API 관련 오버헤드에 의해 MySQL보다는 빠른 속도를 보이나, 최적화한 RocksDB 대비 11배나 느린 속도를 보이는 것을 알 수 있다.

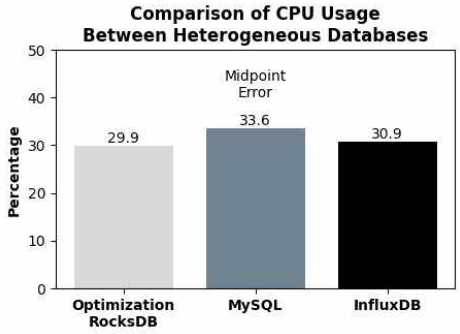

그림 5. 이종 데이터베이스 간의 CPU 사용률 비교
Figure 5. Comparison of CPU Usage Between Heterogeneous Databases

그림 5는 CPU 사용률을 보인다. MySQL의 높은 CPU 사용률을 제외하고는 InfluxDB와 최적화한 RocksDB는 비슷한 CPU 사용률을 보인다.

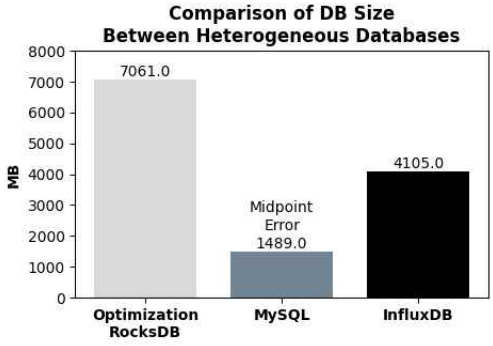

그림 6. 이종 데이터베이스 간의 데이터베이스 용량 비교
Figure 6. Comparison of DB Size Between Heterogeneous Databases

그림 6은 데이터베이스 용량을 비교하였다. MySQL은 중간에 더 이상 수집할 수 없어 가장 적은 용량을 사용하는 것을 볼 수 있었으나, 단순 연산시에도 가장 많은 용량을 사용한다는 것을 미루어 짐작할 수 있었다. 데이터베이스의 용량은 InfluxDB의 압축률이 가장 높은 것을 볼 수 있었다. RocksDB는 조회, Compaction 등의 작업을 설정한 단위로 파일로 나누어 활용하나 InfluxDB의 경우 한 파일을 크게 유지하는 것과 더불어 Gzip 기반 압축 알고리즘을 통해 높은 압축률을 얻는

다.

평가의 결과로 미루어 보아 MySQL은 임베디드 디바이스에서 데이터 수집 시스템으로써의 활용은 어려울 것이다. RocksDB의 경우 InfluxDB에 비해 월등히 높은 성능을 보이는 것을 볼 수 있다. 데이터베이스 용량의 경우는 InfluxDB가 우수함을 볼 수 있었기에 대단위 데이터를 다루며 좋은 성능의 디바이스에서는 유리할 것이다.

Ⅵ. 결 론

본 연구에서는 UD Benchmark라는 범용 데이터베이스 벤치마크의 개발을 통해 이종 데이터베이스들에서도 한가지 지표로 신뢰도 및 성능을 측정해 볼 수 있었다. 또한 시계열 데이터 수집이 주가 되는 데이터 수집 시스템에서 시계열에 최적화할 수 있는 RocksDB의 파라미터로 최적화를 진행하였을 때 저사양 디바이스에서 타 데이터베이스보다 높은 성능을 보여주었다.

첫째로 100개의 센서를 가정하여 초당 데이터를 수집한다고 가정하였을 때 110일 이상의 데이터를 단시간에 수집하여도 임베디드 디바이스에 문제가 발생하지 않음으로써 신뢰성을 확인하였다. 또한 RocksDB Optimization 기법을 적용한 데이터베이스가 성능이 낮은 기기에서는 시계열 데이터베이스보다도 11배 가량 높은 처리 성능을 보여 가장 적합한 데이터베이스라는 것을 실험을 통해 확인하였다.

향후 연구에서는 RocksDB를 시계열 데이터에 맞게 Auto tuning을 진행하여 각 디바이스의 최적의 파라미터를 찾는 연구를 진행하여 낮은 성능의 디바이스에서 최적화 여지를 탐색하고자 한다. 또한 삽입 연산이 주가 되는 데이터베이스에서의 연구를 추가로 진행하여 시계열 데이터 최적화 Key-value store 연구에 기여하고자 한다.

References

[1] Choi, E, Kang, M, Jung, Y, Paik, J. 2017, "Implementation of IoT-based Automatic Inventory Management System", The International Journal of Advanced Culture Technology , vol.5, no.1 pp.70-75. DOI: https://doi.org/10.17703/IJACT.2017.5.1.70

[2] Yong-Soo Lee, Jun Heo, Yong-Hoon Choi “A Study for Space-based Energy Management System to Minimizing Power Consumption in the Big Data Environments”, The Journal of The Institute of Internet, Broadcasting and Communication, Vol. 13, No. 6, pp. 229-235, Dec 2013. DOI: http://dx.doi.org/10.7236/JIIBC.2013.13.6.229

[3] RocksDB, “RocksDB”, https://rocksdb.org/

[4] MySQL, “MySQL”, https://www.mysql.com/

[5] MySQL, "InnoDB Storage Engine", https://dev.mysql.com/doc/refman/8.0/en/innodb-storage-engine.html

[6] InfluxDB, "InfluxDB", https://www.influxdata.com/

[7] InfluxDB, "The TICK Stack", https://www.influxdata.com/time-series-platform/

[8] O'Neil, P., Cheng, E., Gawlick, D. et al. The log-structured merge-tree (LSM-tree). Acta Informatica 33, 351–385, 1996. DOI: https://doi.org/10.1007/s002360050048

[9] Innopolis, Data Acquisition System Market, 2021.04.

[10] Raspberry PI 4 model B, “Raspberry PI 4 model B”, https://www.raspberrypi.com/products/raspberry-pi-4-model-b/?variant=raspberry-pi-4-model-b-4gb

※ 본 연구는 산업통상자원부(MOTIE)의 재원으로 한국에너지기술평가원(KETEP)의 지원을 받아 수행한 연구입니다. (No.20202020900170)